eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Sentimenduen Analisia Ikasketa Automatikoaren laguntzaz

Egilea

Mirai Herrera Piñeiro

2020ko ekaina

Zuzendaria

Basilio Sierra

informatika fakultatea
facultad de informática

2020

Laburpena

Proiektu honetan Sentimenduen Analisia lantzea izan da helburua. Analisi hori, makina bat idatzizko testuak gizakien antzera interpretatzeko gai izatean datza. Datu horiek adierazten dituzten sentimenduak detektatu, hala nola, poztasuna, haserrea, tristura eta halakoak, eta elkarrengandik bereizteko gai izatea da gakoa. Proiektu hau, ordea, sentimendu orokor batzuetara mugatu da, testu bat ea positiboa, neutroa ala negatiboa den jakitera zehazki. Makina bat hori egiteko gai izan dadin Ikasketa Automatikoa aplikatu behar zaio, eta horretarako WEKA softwarea erabili da.

WEKAren bidez datu jakin batzuk entrenatu dira, eta horien ikasketa burutu eta ebaluatu da. Entrenamendua egiteko hainbat metodo desberdin aplikatu dira, eta exekuzio bakoitzarekin emaitza batzuk lortu dira. Emaitza horiek sakonki aztertuz hainbat ondorio atera dira, eta guztiak lanaren memoria honetan ahalik eta ongien azaldu dira.

Hitz gakoak: Sentimenduen Analisia, Ikasketa Automatikoa, WEKA, sailkatzaileak, iruzkinak, sentimenduak.

Abstract

The aim of this project was to work on the Sentiment Analysis. This analysis consists of a machine being able to interpret written texts in a human-like way. The key is to be able to detect the feelings expressed by these data, such as joy, anger, sadness, and so on, and to be able to distinguish them from each other. This project, however, was limited to some general sentiments, such as whether a text is positive, neutral, or negative. A machine, in order to be able to do that, must learn through Machine Learning, which can be applied using the WEKA software.

Using WEKA, certain data have been trained, studied and evaluated. Several different methods of training have been applied, and some results have been achieved with each execution. An in-depth analysis of these results has led to a number of conclusions, all of which have been explained as best as possible in this written memory.

Keywords: Sentiment Analysis, Machine Learning, WEKA, classifiers, comments, feelings.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera eta helburuak

Sentimendu-analisia polaritatea antzematen duen testua analizatzeko metodo bat da. Esaldi, paragrafo zein dokumentu oso batetik datuak hartu, identifikatu, eta haietatik ikastean datza.[1]

Polaritatea aipatzen denean, zerbait positiboa, negatiboa ala neutroa den esan nahi da. Sentimenduen-analisia ez da horretara bakarrik mugatzen ordea. Emozioak eta intentzioak identifikatzen ere saiatzen da, esate baterako, poztasuna, tristura, haserrea, sarkasmoa eta abar.

Proiektu honetan, ordea, polaritatea bakarrik analizatu da, hau da, testu bat ea positiboa, neutroa ala negatiboa den soilik ikertu da.

Sentimenduak identikatzea ez da batere lan erraza. Testuak anbiguoak izan daitezke, eta sarkasmoa adierazten denean hitz positiboak erabiltzen dira intentzio negatiboarekin. Esate baterako, ”Ederra egin duzu!”, ”Primeran! Falta zena!” edota ”Pozez zoratzen nago, bai!” esaldiak. Askotan gizaki batek berak ere ezin du intentzioa desberdindu, kontesturik ematen ez bazaio behintzat.

Gaur egun, edozein negoziotan, bezeroen iritzia jakitea eta haien sentimenduak ulertzea ezinbestekoa da negozioak aurrera egin dezan. Horretarako jendeak ematen dituen *feedback*-iritziak analizatu behar dira, eta zer hobeto lan hori automatikoki egitea baino. Gaur egun sare sozialak izugarri zabaldu dira, eta jendea etengabe aritzen da bere iritzia libreki plazaratzen. Datu-jario handi horri jarraipen bat egitea oso aberasgarria da, eta zeregin hori automakikoki egiteak denbora eta lan ugari aurrezten du.

Analisia egiteko ikasketa automatikoa (*Machine Learning*) baliatu da. Ikasketa automatikoa makina batek esperientziatik bere kabuz ikastean datza, hau da, gizakien portaera garatzea eta imitatzea.[2] Makina batek, zerbait ikasi ahal izateko, hasierako datu batzuk behar ditu. Datu horiek entrenamendu-datuak izango dira, eta

ikasketa-prozesuaren bidez emaitza batzuk lortuko dira.

Ikasketa egiteko bi modu desberdin daude: ikasketa gainbegiratua eta ez-gainbegiratua. Proiektu honetan ikasketa gainbegiratua erabili da, helburua lortzeko modurik praktikoena delako. Izatez, bi horietaz gain ikasketa erdi-gainbegiratua ere badago, baina horretan sartzeak ez du merezi.

Ikasketa gainbegiratuan makinari *input* eta *output* datuak ematen zaizkio, hau da, hasierako datuak eta emaitzak. Emandako datuek adibide bezala funtzionatzen dute, makinak eredu bat ikas dezan. Makina, ikasitako ereduan oinarrituta, emaitzak aurreikusten saiatuko da, datuak sailkatuz. Helburua, makinak aurreikusitako emaitzak eta benetako emaitzak ahalik eta gehien bat etortzea da. Zenbat eta gehiago entrenatu, orduan eta aurreikuspen hobeak egingo ditu.

Bestalde, ikasketa ez-gainbegiratuan makinari *input* datuak besterik ez zaizkio ematen. Hasierako datuak bakarrik, emaitzarik gabe. Baina esan bezala, metodo hau ez da proiektu honetan erabili.

Hortaz, proiektu honen helburua metodo desberdinak erabiliz ahalik eta ikasketarik onena lortzea izan da. Horretarako WEKA softwarea erabili da, eta metodo desberdinak probatu ahala lortutako emaitzen balorazioa egin da.

Erabilitako metodoak ez dira hasieratik guztiz finkatu. Zeintzuk erabili gutxi gorabehera buruan eduki arren, pauso bakoitza aurreko pausoaren araberakoa izan da. Hau da, uneko metodoak eman dituen emaitzek erabaki dute hurrengo zer metodo probatu edota zer egin zehazki.

Txosten hau hainbat atal nagusitan banatuta dago. 1. atala honako hau da, sarrera eta helburuak azaltzen dituena. 2. atala proiektuaren plangintza da, lana nola antolatu den eta denbora nola banatu den azaltzen dituena. 3. atalean proiektua egiteko erabili diren tresnak azaltzen dira, eta 4. atalean, berriz, lana burutzeko erabili diren hasierako datuak. 5. atalean garapenean zehar erabili diren metodo eta kontzeptu askoren azalpenak daude. 6. atala garapena da, proiektuan egindako lan guztia. Lortutako emaitzak ere atal honen barruan daude. Izan ere, pauso bakoitzak bere emaitzak ditu, eta pauso bakoitza aurreko pausoaren emaitza arabera ematen da. Beraz, emaitzak beste atal batean banatzea nahasgarria izan zitekeela ondorioztatu eta garapenaren barruan uztea erabaki da. Azkenik, 7. atalean proiektu amaieran ateratako ondorioak eta haien hausnarketak biltzen dira.

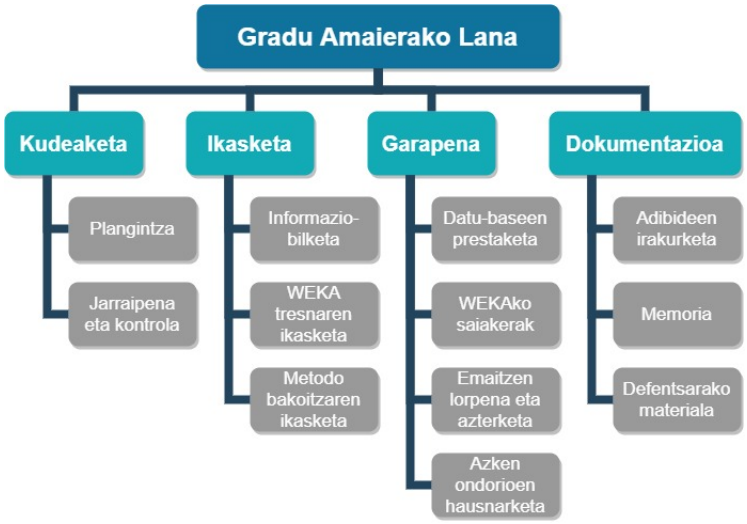
2. KAPITULUA

Plangintza

Atal honetan, proiektua aurrera eraman ahal izateko eraiki den planifikazioaren azalpena ematen da. Modu eraginkorrean lan egiteko, urritik ekaineraino egin den lan guztia nola antolatuko den hain zuzen ere.

2.1 Lan-paketeen identifikazioa

Lana nola banatuko den erakusteko LDE (Lanaren Deskonposaketa Egitura) diagrama bat erabili da. Lan osoa lau ataza nagusitan banatuko da, eta bakoitzak bere azpiatazak edukiko ditu, 2.1 Irudian ikusten den bezala.



2.1 Irudia: Plangintzaren LDE diagrama.

Lau ataza nagusiak lana antolatzea (kudeaketa), egin beharrekoari buruz informatzea (ikasketa), proiektua burutzea (garapena) eta egindako guztia paperean idaztea (dokumentazioa) izango dira. Hauxek dira ataza bakoitzaren deskribapenak:

– **Kudeaketa**: Bi azpiatazatan banatuta dago. Lehena plangintza da, proiektuan egin beharreko guztia planifikatzea hain zuzen. Bigarrena, berriz, jarraipena eta kontrola da. Azken hau, proeiktuan zehar planifikatutakoa betetzen dela bermatzea da, adierazitako puntu bakoitza erabakitako denbora-tartean egiten dela eta proiektua garaiz amaitu ahalko dela ziurtatzea alegia.

– **Ikasketa**: Hiru azpiatazatan banatuta dago. Lehena informazio-bilketa da, gaiari buruz informazioa bilatzea eta ulertzea hain zuzen. Bigarrena, berriz, WEKA tresnaren ikasketa da. WEKA karreran zehar erabilitako programa bat denez, funtzionamendua jadanik ezaguna da. Hala ere, software honen erabilera sakonago ikasi beharko da. Azkenik, hirugarren azpiataza metodo bakoitzaren ikasketa da, WEKAn erabiliko den metodo bakoitza zertan datzan ulertzea alegia.

– **Garapena**: Lau azpiatazatan banatuta dago. Lehena datu-baseak prestatzea da, proiektuan erabiliko diren hasierako datuak lortzea eta egokitzea hain zuzen. Bigarrena, berriz, WEKAko saiakerak egitea da, aurretik ikasitako metodo bakoitza WEKAn probatzea alegia. Hirugarren azpiataza emaitzen lorpena eta azterketa da, hau da, saiakeretatik emaitzak ateratzea eta hauek aztertzea. Azkenik, laugarrena azken ondorioen hausnarketa da, hots, proiektuaren amaieran emaitza guztiak lotuz konklusio batzuk ateratzea.

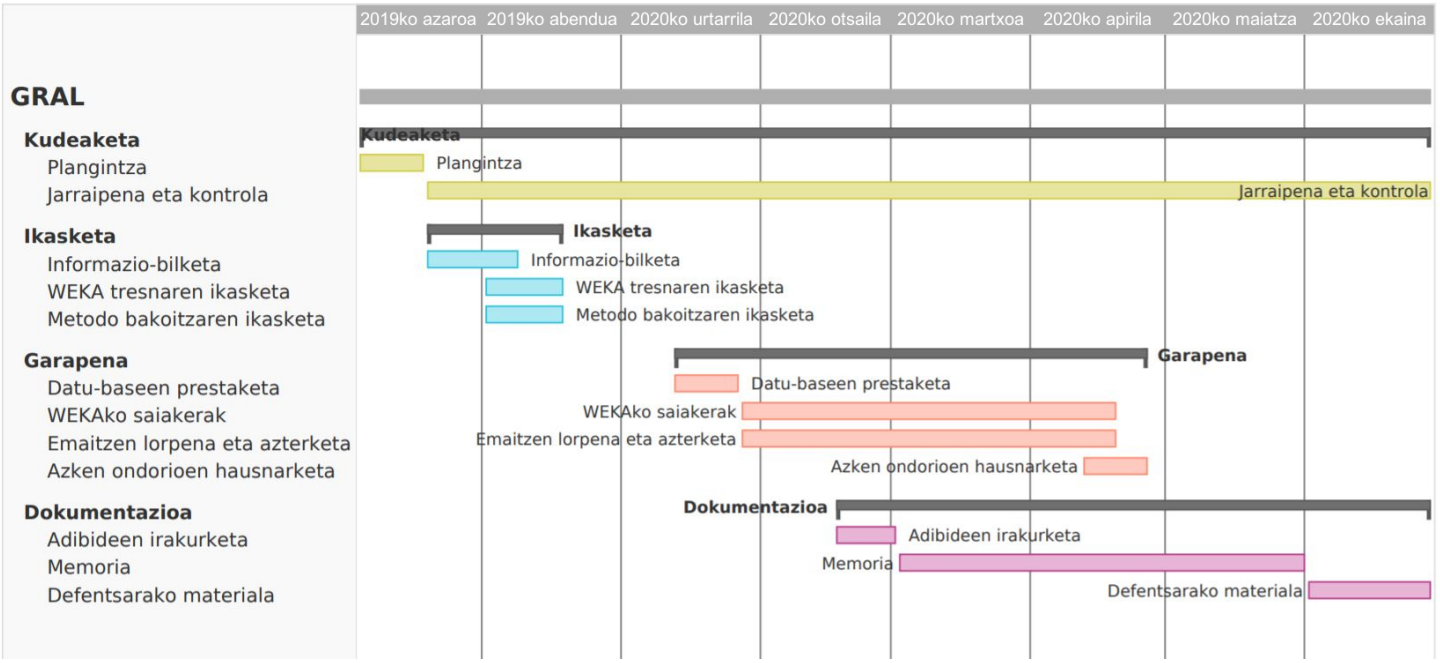
– **Dokumentazioa**: Hiru azpiatazatan banatuta dago. Lehena adibideen irakurketa da, memoria nola idatzi jakiteko adibide batzuk irakurtzea hain zuzen. Bigarrena memoria bera idaztea da. Azkenik, hirugarrena defentsarako materiala prestatzea da, epaimahaiaren aurrean lana aurkezteko diapositiba batzuk eta gidoi bat gertatzea, besteak beste.

2.2 Planifikazio-egutegia

Proiektua 2019ko azaroaren hasieran abiatu da, eta entregatzeko epea 2020ko ekainaren amaieran izango da. Bitarte horretan gradu amaierako lana egiteko denbora soberan dagoen arren, gauza bakoitza noiz egingo den hasieratik ongi planifikatu beharra dago, amaieran ustekaberik egon ez dadin.

Horretarako, lehen gauza plangintza hau egitea da, noski. Horren ondoren, hurrengo gauza ikasketa burutzea izango da, eta Eguberriak iristen direnerako lanerako behar den informazio guztia bilduta egongo da. Gabonetako opor-egunen ostean, geratzen diren bospasei hilabeteak garapena eta dokumentazioa egiteari eskainiko zaizkio.

2.2 Irudian, planifikazio hori guztia Gantt diagrama baten bidez adierazi da.



2.2 Irudia: Plangintzaren Gantt diagrama.

Garapenaren barruan, WEKAko saiakerak eta emaitzen lorpena eta azterketa aldi berean egitea erabaki da. Izan ere, kasu honetan garapena prozesu errepikakor bat izango da: WEKAn metodo bat probatu, emaitzak lortu, ondorioak atera, eta horren arabera beste metodo bat probatu.

Proiektua ongi garatzeko, gutxi gorabehera 300 orduko dedikazioa beharko da. Hori ikusita, lan-pakete bakoitzari denbora bat esleitzea erabaki da, denboraren kontrol egokia egin ahal izateko. 2.1 Taulan, denbora horiek taula baten bidez adierazi dira.

LAN-PAKETEA	ORDU-KOP. ESTIMATUA
- Kudeaketa -	**10**
Plangintza	5
Jarraipena eta kontrola	5
- Ikasketa -	**70**
Informazio-bilketa	25
WEKA tresnaren ikasketa	15
Metodo bakoitzaren ikasketa	30
- Garapena -	**120**
Datu-baseen prestaketa	15
WEKAko saiakerak	50
Emaitzen lorpena eta azterketa	40
Azken ondorioen hausnarketa	15
- Dokumentazioa -	**100**
Adibideen irakurketa	5
Memoria	80
Defentsarako materiala	15
GUZTIRA	**300**

2.1 Taula: Lan-pakete bakoitzari eskaintzea aurreikusi den denbora.

2.3 Lan-metodologiaren plana

Gradu amaierako lana eskolaz kanpoko beste hainbat eginbeharrekin uztartu beharko da. Baina eginbehar horietako asko zehazki zein egunetan burutu beharko diren ezin denez aurretiaz jakin, ezingo da egun bakoitzerako lan-ordutegi finko bat ezarri.

Zortzi hilabetetan zehar, egun oso lanpetuak eta egun oso libreak egongo direla aurreikusten da. Ziurrenik egun gehienak zuriak edo beltzak izango dira: lanerako denbora asko edukiko da edota ez da batere denborarik edukiko. Egun batzuetan lana aurreratu ezingo den arren, egun libreetan lanean buru-belarri murgildu eta hainbat ordu sartzeko asmoa dago.

Kalkulatu bezala, proiektua egiteko 8 hilabete inguru daude, 34 aste zehazki. Hori ikusita, astero gutxienez 10 ordu dedikatzea erabaki da, eta hilabetero gutxienez 40 ordu. Horrela, egunero lan egin ez arren, astero denbora minimo hori betetzen bada, proiektua arazorik gabe egunean eraman ahalko da.

Dena dela, badaezpada proiektuan zehar ezustekoren bat gertatzen den, hobe da aurreratuta joatea atzeratuta baino. Hortaz, ahal izan den heinean, hasieratik denbora gehiago dedikatuko da. Denbora soberan egotekotan, hobe da amaieran egotea.

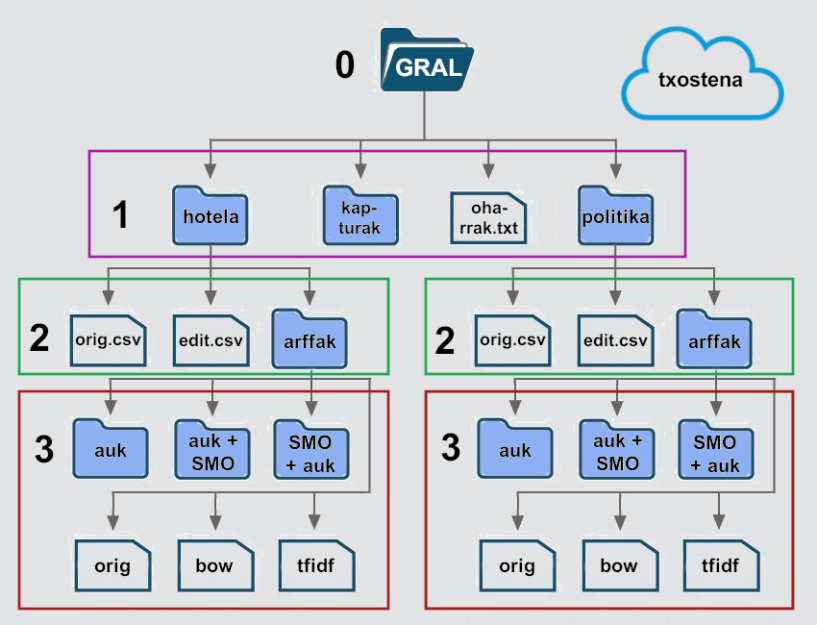
Lanean zehar egiten diren eta lortzen diren gauza guztiak *Google Drive*ko testu-dokumentu batean idatziko dira, zirriborro moduan. Aurrerago, dokumentazioa egiteko unea iristen denean, zirriborroak garbira pasatuko dira eta kontzeptuak gehiago garatuko dira. Dokumentazioa garbian idazteko LaTeX erabiliko da, lana txukun eta profesional gera dadin.

LaTeXen dokumentazioa egin ahala, proiektuaren jarraipena eta kontrola egiteko *Google Drive*ko dokumentua erabiltzen jarraituko da, egun bakoitzean zer egin den eta abar apuntatzeko. Datu horiek, amaieran, hasierako plangintzarekin alderatzeko beharko dira.

2.4 Fitxategien antolaketa-plana

Proiektuan zehar erabili beharko diren fitxategiak, ordenatuta edukitzeko, karpeta batzuetan antolatuko dira. Hori egitea oso onuragarria izango da, batez ere fitxategiak WEKArekin kargatu ahal izateko. Antolamendu honi esker, aldi bakoitzean behar izan den fitxategia berehala lekutu ahal izango da.

2.3 Irudian, fitxategien antolaketa-zuhaitza azaltzen da.



2.3 Irudia: Proiektuko fitxategien antolaketa-zuhaitza.

Karpeta nagusi gisa, ordenagailuko mahaigainean ”GRAL” karpeta sortuko da. Horren barruan lau gauza egongo dira: HOTELA datu-basearen karpeta, POLITIKA datu-basearen karpeta, txostenarako beharko diren irudien karpeta eta oharrak idazteko testu-fitxategi bat.

HOTELA datu-basearen karpetak eta POLITIKA datu-basearen karpetak egitura bera izango dute. Karpeta bakoitzaren barruan jatorrizko datu-basea (orig.csv), proiekturako bereziki editatutako datu-basea (edit.csv) eta ARFF fitxategien karpeta egongo dira. Azkenik, ARFF karpetaren barruan metodo desberdinetarako erabili diren fitxategi desberdinak egongo dira, horietako batzuk azpikarpetetan multzokatuta.

Hori guztiaz gain, txostena eta bestelako informazio guztia hodeian egongo da, internet bidez atzigarri.

2.5 Komunikazio-plana

Proiektuari buruzko zalantzaren dagoen aldiro zuzendariari komunikatuko zaio, gehienbat posta elektroniko bidez. Horretaz gain, bilera fisikoak ere egingo dira.

Proiektuaren lehen hilabeteetan, lanaren norabidea finkatzeko, unibertsitatean bertan ia astero bilera bat egingo da. Behin helburuak finkatuta eta ikasketa eginda, garapen fasean zehar bilerak urriagoak izango dira. WEKAn metodo batekin probak egiten bukatu aldiro, bilera bat antolatuko da, lortutako emaitzak zuzendariarekin komentatzeko eta hurrengo pausora igarotzeko.

Zuzendariari, gutxienez entrega-epea baino hiru aste lehenago, txostena bidaliko zaio bere oniritzia jasotzeko. Hiru asteko denbora-tarte hori azken zuzenketak egiteko utziko da.

Azkenik, dena ongi dagoela ziurtatu denean, itxiera-bilera bat egingo da aurkezpena prestatu eta proiektuari amaiera emateko.

2.6 Arriskuen kudeaketa-plana

Proiektua denbora luzekoa izango denez, oso garrantzitsua da gerta daitezkeen arazoak aurreikustea. Jarraian, arrisku horiek zeintzuk izan daitezkeen eta zer konponbide posible izan dezaketeen azalduko da:

– **Ezjakintasuna**: Interneten orokorrean informazio asko egoten den arren, batzuetan zaila izaten da kontzeptu batzuk ulertzea. Informazioa bera aurkitzeko ere arazoak egon daitezke, nola bilatu ez jakiteagatik edota horri buruzko informazioa oso urria izateagatik. Proiektuan zehar zerbait ez ulertu eta trabatuta gelditzeko arriskua dago. Zerbait ulertzeko zailtasunak daudenean, tutorearengana joko da aholku eske.

– **Lanaren galera**: Proiektua egiteko erabiliko den material guztia ordenagailuan besterik gabe edukitzea arriskutsua izan daiteke. Ordenagailuari edozer gauza gerta dakioke eta egun batetik bestera datu guztiak galdu. Hori ekiditeko, *Google Drive* erabiliko da noizbehinka fitxategien segurtasun-kopiak gordetzeko. Horretaz gain, txostena zuzenean online dauden testu-editoreetan idatziko da. Horrela ez da uneoro ”gorde” botoia sakatzen ibili beharko, eta aurrerapen guztiak denbora errealean gordeko dira.

– **Denbora-falta**: Printzipioz, denbora-arazorik ez litzateke egon beharko. Izan ere, edozein ustekabe gertatu arren, zortzi hilabete daude lana egiteko. Plangintza ongi jarraituz gero, denbora ez da kezkatzeko arrazoi bat izango.

– **Ordenagailuaren muga**: WEKA programa karreran erabiltzetik ezaguna da. Datu-meatzaritza ikasgaian erabili zen, eta laborategiko ordenagailuekin arazorik ez zegoen arren, 24 orduko gelako ordenagailuek askotan ez zeukaten exekuzioak egiteko indar nahikorik. Proiektua egiteko etxeko ordenagailua erabiliko da, baina hau ere ez da superkonputagailu bat. Beraz, oso posiblea da exekuzio batzuetan WEKA memoriarik gabe geratzea. Hori gertatzearen probabilitatea gutxitzeko, tamaina egoki bateko datu-baseak bilatuko dira.

– **Komunikazio txarra**: Zuzendariarekin edozein unetan komunikazioa eten edota gaizki-ulertuak egoteko arriskua dago. Hori ekiditeko, noizbehinka zuzendariari proiektuaren egoeraren berri emango zaio, eta arazoren bat sortzen den bakoitzean momentuan bertan jakinaraziko zaio. Horretaz gain, bilerak aldez aurretik hitzartuko dira, datekin arazorik egon ez dadin.

2.7 Plangintza vs. errealitatea

Atal honetan, hasieran planifikatutakoa errealitatean gertatu denarekin alderatzen da. Behin proiektua amaituta dagoelarik, atzera begiratuta, jarraipena nolakoa izan den deskribatzen da.

Planifikazio-egutegian, garapena Eguberrietako oporren ondoren hasiko zela, eta une hori iritsi arte informazioa bildu eta gauzak nola egin ikasiko zela, erabaki zen. Errealitatean ordea, plangintza eta ikasketa-prozesua uste baino lehenago burutu dira, eta garapenari abenduan eman zaio hasiera. Dokumentazioa idazten ere lehenago hasi da, otsaila hasieran hain zuzen ere.

Lan egiteko metodologian, astero gutxienez 10 ordu eta hilabetero gutxienez 40 ordu dedikatuko zirela erabaki zen. Errealitatean ordea, egun lanpetu asko egon dira bata bestearen atzetik, eta baita egun libre asko jarraian ere. Ondorioz, aste batzuetan ezin izan zaizkio lanari 10 ordu horiek eskaini, baina aldiz, beste aste batzuetan hori eta bikoitza ere dedikatzeko astia egon da. Beraz, aste batzuetan galdutako denbora beste aste batzuetan errekuperatu da, eta azkenean hilabetero 40 ordu sartzearen promesa ongi bete ahal izan da.

Fitxategien antolaketan, ARFF karpetetan fitxategi mordoa pilatu dira, azkenean uste baino fitxategi askoz gehiago sortu behar izan direlako. Hala ere, fitxategi guztiak txukun ordenatuta eduki dira, azpikarpeta gehiagotan multzokatuta. Beraz, planifikatutako antolaketa oso praktikoa izan da.

Komunikazioaren aldetik, proiektuaren erdialdetik aurrera aldaketa handiak gertatu dira. Izan ere, aurreikusi gabeko arazo bat agertu da, mundu mailan eragin handia eduki duen arazoa. Arazo horren izena COVID-19 da, eta pandemia horren eraginez komunikazio fisikoa erabat moztu da. Hala ere, posta elektroniko bidezko komunikazioa mantendu egin da, eta derrigorrez etxean sartuta egon behar izan denez, bideokonferentziak egiteko plataforma bat erabili da bilerak egin ahal izateko. Pandemiak sortu dituen arazo guztiak gorabehera, epeak ongi mantendu dira, eta proiektuan ez du eragin handiegirik izan.

Azkenik, 2.2 Taulan ataza eta azpiataza bakoitzerako behar izan diren denborak adierazten dira, plangintzan estimatu ziren denborekin alderatuta.

LAN-PAKETEA	ORDU-KOP. ESTIMATUA	BENETAKO ORDU-KOP.
- Kudeaketa -	**10**	**10**
Plangintza	5	5
Jarraipena eta kontrola	5	5
- Ikasketa -	**70**	**45**
Informazio-bilketa	25	20
WEKA tresnaren ikasketa	15	5
Metodo bakoitzaren ikasketa	30	20
- Garapena -	**120**	**140**
Datu-baseen prestaketa	15	20
WEKAko saiakerak	50	80
Emaitzen lorpena eta azterketa	40	30
Azken ondorioen hausnarketa	15	10
- Dokumentazioa -	**100**	**110+**
Adibideen irakurketa	5	10
Memoria	80	100
Defentsarako materiala	15	-
GUZTIRA	**300**	**305+**

2.2 Taula: Lan-pakete bakoitzari eskaintzea aurreikusi den denbora eta benetan eskaini zaion denbora.

Ikus daitekeen bezala, ez da desbiderapen handirik egon. Desbiderapen nabarienak bi atazatan egon dira: WEKAko saiakerak egiterakoan eta memoria idazterakoan.

Kudeaketa planifikatutako denboran burutu da. Ikasketa egiteko, berriz, pentsatu baino denbora gutxiago behar izan da, batez ere WEKA tresna aurretik nahiko ongi ezagutzen zelako. Garapenean, ordea, WEKAko saiakerak egiteko denbora askoz gehiago behar izan da. Honen arrazoia, espero baino askoz exekuzio gehiago egin behar izan direla izan da, eta exekuzio batzuk oso luzeak izan direla. Dokumentazioan, bestalde, adibideak irakurtzen uste baino apur bat denbora gehiago igaro da, eta memoria idazteko are denbora gehiago hartu da.

Dena dela, memoria idazteari ordu gehiago eskaini izana, berez ez da planifikazio-akatsa izan. Entrega-epea arte oraindik denbora zegoenez, berridazketa asko egiteko denbora hartu da, kontzeptuak ahalik eta ulergarrien azalduta uzteko asmoarekin.

Azkenik aipatu behar, hau idatzi den momentuan oraindik ez dela defentsarako materiala prestatu. Ataza hori, defentsa-eguna arte falta diren datozen hiru asteetan burutuko da.

3. KAPITULUA

Erabilitako tresnak

Proiektua egiteko hainbat tresna erabili behar izan dira. Atal honetan, tresna horiek zeintzuk izan diren eta zertarako erabili diren dago azalduta.

3.1 WEKA

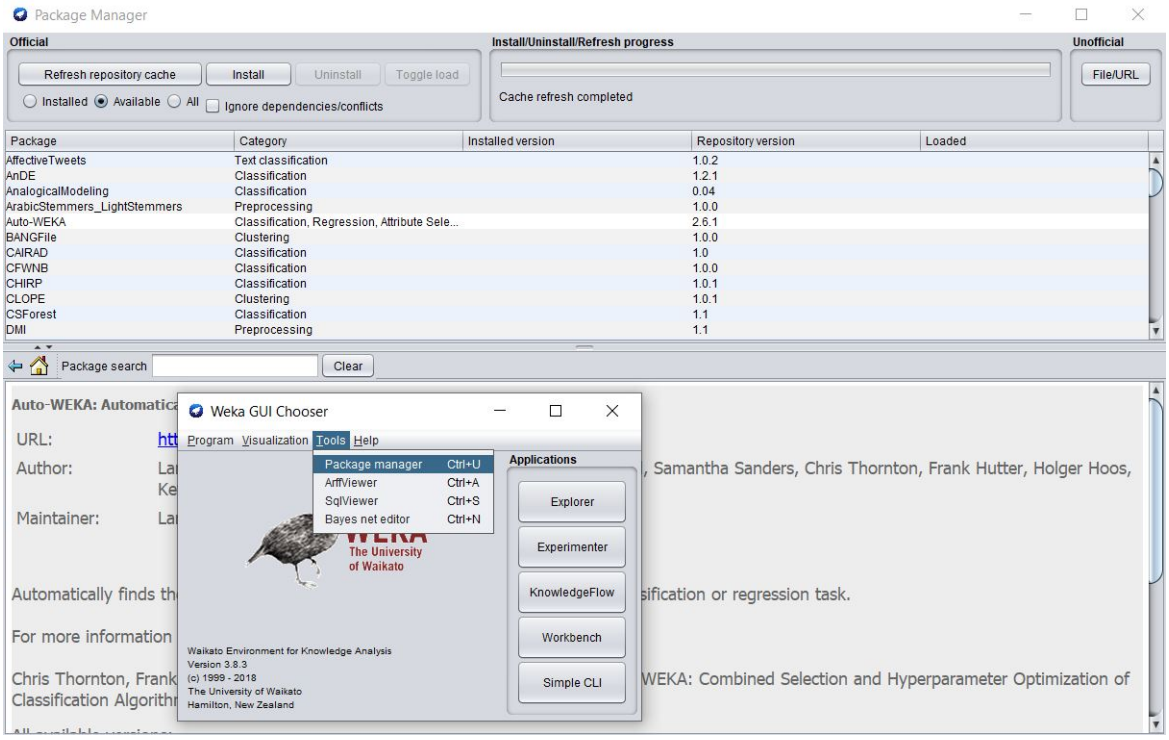
Hau izan da proiektu honetan erabili den tresna nagusia. 3.1 Irudian bere interfazearen hasierako pantaila ikus daiteke.

3.1 Irudia: WEKAren interfaze nagusia.

WEKA (*Waikato Environment for Knowledge Analysis*, hau da, ezagutzaren analisirako Waikato ingurunea) ikasketa automatikoan eta datu-meatzaritzan erabiltzen den software-ingurune bat da.[3] Zeelanda Berriko Waikato izeneko unibertsitatean garatu zen mundu guztiko ikertzaileen laguntzaz, eta JAVA programazio-lengoaian inplementatuta dago. GNU-GPL lizentziapean banatzen den software librea da. Bitxikeria gisa, WEKAren ikur moduan erabiltzen den txoria ”weka” bat da, Zeelanda Berrian bakarrik bizi den hegazti-espezie bat hain zuzen ere.

WEKA softwareak, datuen analisia egiteko eta sailkapen-eredu iragarleak sortzeko, hainbat bistaratze-tresna eta algoritmo eskaintzen ditu. Bere interfaze grafikoari esker funtzio hauek guztiak erabiltzea oso erraza gertatzen da.

Gainera, WEKA barruan inplementatuta dagoen ”Package manager” (pakete-kudeatzailea) izeneko tresnari esker, Waikato Unibertsitateko garatzaileen taldetik kanpoko beste garatzaileei ere, haien software-garapenak erabiltzaile-komunitatearen eskura jartzeko aukera ematen zaie. Garapen horiek pakete moduan eskaintzen dira, eta nahi dituenak pakete-kudeatzailetik karga ditzake. Aukera horrek, WEKAk eskaintzen duen algoritmo- eta baliabide-sorta asko zabaltzen du. Kudeatzaile horren interfazea 3.2 Irudian ikus daiteke.



3.2 Irudia: WEKAren ”Package manager” interfazea.

Proiektuan zehar erabili diren metodo batzuk, hala nola aurrerago azalduko diren *ReliefFAttributeEval* eta *SymmetricalUncertAttributeEval* atributu-ebaluatzaileak, pakete-kudeatzailearen bidez instalatu dira.

Lan honetan, datuak WEKAn sartu eta sailkatzaile desberdinak erabili dira hainbat aurreikuspen egiteko. Helburua iragarpen horiek benetako emaitzetatik ahalik eta hurbilen egotea da, eta hori lortzeko, sailkatzaile arruntak ez ezik, atributu-aukeraketa eta sailkatzaile-konbinazio desberdinak ere erabili dira. Hori guztia egiteko WEKA zehazki nola erabili den txostenaren garapenean dago xehetasun handiz azalduta.

3.2 Txostena idazteko tresnak

– *Google Sheets*: Googleren kalkulu-orriak, errenkadaz eta zutabez osatutakoak. Garapenean zehar lortutako emaitzak tauletan sartu ahal izateko erabili dira.

– *Notepad++*: Hainbat programazio-lengoia desberdinekin bateragarria den testu-editorea.[4] Itxuraz testu-editore arrunta dirudien arren, eta oso erabilerraza den arren, funtzio aurreratu eta erabilgarri asko barneratzen ditu. CSV eta ARFF motako fitxategiak editatzeko erabili da, besteak beste.

– *Snipping Tool*: Ebaketa-tresna. Pantailako edozein zati ebaki eta irudi moduan gordetzeko tresna. Windowsen tresna natibo bat da. Txostenan zehar dauden ia irudi guztiak honen bidez lortu dira. Esate baterako, WEKAren interfaze desberdinak eta emaitzen taulak.

– *Gimp*: Irudiak editatu eta manipulatzeko programa.[5] Ebaketa-tresnaren bidez moztutako irudi-zatiak irudi bakarrean elkartzeko eta Wikipediatik ateratako irudi batzuk euskaratzeko erabili da.

– *Overleaf*: Onlineko LaTeX-editorea. LaTeX kalitate handiko tipografia duten dokumentuak sortzeko diseinatuta dagoen software libre bat da.[6] Bere ezaugarriengatik eta eskaintzen dituen aukerengatik, artikulu eta liburu zientifikoak sortzeko asko erabiltzen da, besteak beste, adierazpen matematikoak dituztenetan. Overleaf, berriz, LaTeX softwarea sarean erabiltzea ahalbidetzen duen webgune bat da, instalaziorik gabe eta erosotasun guztiekin.

3.3 Komunikazio-tresnak

– *Web Posta EHU*: Euskal Herriko Unibertsitateak eskaintzen duen posta elektronikozko zerbitzua.[7] Proiektuaren zuzendariarekin harremanetan mantentzeko erabili da.

– *Jitsi Meet*: Internet bidezko bideokonferentziak egiteko aplikazioa.[8] Zuzendariarekin bilerak egiteko erabili da. COVID-19 gaixotasun pandemikoa dela eta, proiektuaren erditik aurrera ezin izan da bilera fisiko gehiagorik egin, eta tresna hau lanerako ezinbestekoa izan da.

4. KAPITULUA

Erabilitako datuak

Proiektua burutzeko, datu-multzo (*dataset*) gisa bi datu-base desberdin erabili dira, bien arteko emaitzak alderatuz ondorio gehiago atera ahal izateko. Txostenean zehar, lehenengo datu-baseari HOTELA deitu zaio, eta bigarrenari, berriz, POLITIKA. Garapenean emaitzak eta abar azaltzerakoan, izen hauek erabili dira.

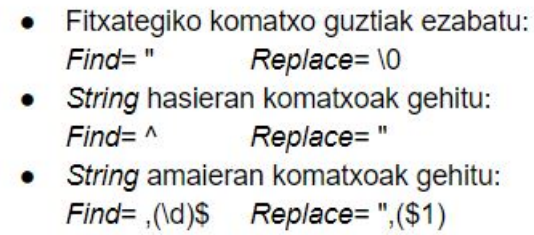
HOTELA datu-basea hoteletako iruzkinak dira, Indiako Chennai hiriko hainbat hoteletako bezeroen iruzkinak hain zuzen. Datuak *www.kaggle.com/datasets* webgunetik atera dira. Kaggle enpresa filial bat da, gaur egun Googleren eskuetan dagoena.[9] Erabiltzaileek igotzen dituzten datuak eta kodeak biltzen dituen doako gordailu handi bat, hodeian kokatua dagoena. Bertan gai askotariko datu-baseak aurki daitezke. Gainera, komunitate zabal bat dauka, eta txapelketak ere antolatzen dira.

Datu-baseari dagokionez, webgunetik deskargatutako fitxategiak behar ez zen informazio asko zeukan. Beraz, soberan zeuden zutabeak kendu behar izan dira, hala nola, hotelaren izena, iruzkinaren izenburua eta balorazio-portzentaia. Utzi diren zutabeak bi bakarrik izan dira: iruzkina eta sentimendua. Iruzkina *string* motakoa da, baina sentimendua *integer* motakoa. Sentimendua positiboa baldin bada ”3”, neutroa baldin bada ”2”, eta negatiboa baldin bada ”1”. Erabili den datu-multzoak, beraz, 4.1 Irudiko itxura dauka. Iruzkinak, ordea, datu-basean ingelesez idatzita daude.

IRUZKINA	SENTIMENDUA
“Oso pozik egon naiz.”	3
“Komuna buxatuta zegoen”	1
“Jende gutxi zegoen”	2

4.1 Irudia: Datu-multzoen itxuraren adibidea.

Horretaz gain, fitxategiko lerro batzuk bikoiztuta edo gaizki definituta zeuden. Bikoiztutako lerroak kontsolan *sort -u myfile.csv -o myfile.csv* komandoa idatziz ezabatu dira, eta gaizki definituta zeuden apurrak eskuz zuzendu edota ezabatu behar izan dira. Horretarako, Notepad++ programako *find&replace* funtzioa oso erabilgarria izan da. Batez ere lerro batzuek kakotxak falta zituztelako, eta *string* motako aldagaiak izanik lerro guztiek kakotxen artean joan behar dutelako. Hau automatikoki konpontzeko adierazpen erregularra erabili da, 4.2 Irudian azaltzen den bezala.



4.2 Irudia: Datu-multzoak konpontzeko egin diren aldaketak.

Konponketa guztiak egin ondoren, fitxategia 3212 iruzkinez osatuta geratu da: 2192 positibo, 644 neutro eta 376 negatibo.

POLITIKA datu-basea, ordea, proiektu honen zuzendari den Basilio Sierra tutoreak jarri du mahai gainean. Duela hiru urte Oscar Miguel Cumbicus Pineda izeneko ikasle batek master amaierako lana egiteko erabili zuen datu-multzoa da. Lan hori ere Basilio Sierra tutoreak gidatu zuen.

Datu-base honek 2017an Ecuadorren egin ziren hauteskundeei buruzko Twitterreko hainbat iruzkin biltzen ditu. HOTELA datu-baseak baino datu-kopuru askoz txikiagoa dauka, 202 iruzkin hain zuzen ere: 69 positibo, 108 neutro eta 25 negatibo. Honetan ere, beharrezkoak ez ziren zutabeak kendu eta konponketa txiki batzuk egin behar izan dira. Aipatu beharra dago ere, datu-base honetako iruzkinak gazteleraz daudela. Gainera, sentimenduen zutabea, aurreko datu-basean ez bezala, oraingo honetan *string* bat da, ”positiva”, ”neutral” eta ”negativa” hitzak erabiltzen dituelako ”3”, ”2” eta ”1” zenbakien ordez.

5. KAPITULUA

Kontzeptuak

Garapenean zehar hainbat kontzeptu eta algoritmo aipatzen dira. Hori guztiari buruzko azalpena atal honetan ematen da. Garapenaren atalean, kontzeptuetako bat agertzen den bakoitzean, azalpena jakiteko atal honetara jotzeko iradokitzen duen ohar bat dago.

5.1 CSV eta ARFF fitxategiak

CSV (*Comma Separated Values*) formatua taula itxurako fitxategi mota bat da. Errenkada bakoitza fitxategiko lerro bat da, eta zutabeak adierazteko gelaxka bakoitza koma baten bidez separatzen da. Gelaxka baten barruan komak erabili nahi badira, gelaxka osoa kakotxen artean sartu behar da.

```
Izena,HTML kodea,RGB margoak
Txuria,#FFFFFF,"255,255,255"
Beltza,#000000,"0,0,0"
Arrosa,#FF00FF,"255,0,255"
```

5.1 Irudia: CSV motako fitxategi baten adibidea.

Egitura hori 5.1 Irudiko adibidean ikus daiteke. Lehenengo lerroan atributu bakoitzaren izena idazten da, eta gainerako lerroetan instantziak idazten dira.

ARFF (*Attribute Relation File Format*) formatua, berriz, WEKAk soilik erabiltzen duten testuzko fitxategi mota bat da. Fitxategi horietan atributuen arteko erlazioa definitzen da. ARFF fitxategi batek hiru atal izaten ditu: erlazioa (@relation), atributuak (@attribute), eta datuak (@data).

```
@relation lore
@attribute izen string
@attribute kolore {arros,urdin,hori,more,gorri,berde,txuri,beltz}
@attribute kopuru numeric
@data
“lilium candidum”,txuri,5
“rosa rubiginosa”,arros,2
“tulipa gesneriana”,gorri,3
```

5.2 Irudia: ARFF motako fitxategi baten adibidea.

Egitura hori 5.2 Irudiko adibidean ikus daiteke. Erlazioa lehen lerroan idazten da, eta hurrengo lerroetan atributuak idazten dira, ordena kontuan hartuta. Atributuen ondoren datuak idazten dira, taula moduko bat irudikatuz. Hortik aurrerako lerro bakoitza instantzia bat izango da. Instantzia bakoitzaren barruko balioak koma bidez banatzen dira, zutabeak irudikatuz, eta bakoitzak atributu bati egiten dio erreferentzia, ordena kontuan hartuta. Atributuren bat *string* motakoa bada, atributu horri erreferentzia egiten dioten balio guztiak kakotxen artean idatzi behar dira.

5.2 Hitz-zakuak

Hitz-zakua (*bag-of-words*) testu bidezko datuak adierazteko teknika bat da. Hizkuntzaren prozesamenduan (*NLP: Natural Language Processing*) oso erabilia da, dokumentu-sailkapena (*Document Classification*) egiteko batez ere.[10]

Hizkuntzaren prozesamendua informatika (*Computer Science*), adimen artfiziala (*AI: Artificial Intelligence*) eta hizkuntzalaritza (*Linguistics*) batzen dituen alorra da.[11] Hizkuntzaren bidez pertsona eta makinen arteko komunikazioa errazteko tresna konputazionalak ikertzeaz arduratzen da.

Dokumentu-sailkapena, berriz, dokumentu bateko informazioa klase edota kategorietan sailkatzea da. Hau eskuz edo automatikoki egin daiteke, eta automatikoki egiteari ”ikasketa automatikoa” (*Machine Learning*) deritzo, txostenaren sarreran azaldutakoa hain zuzen.

Hitz-zakuaren teknikak hitz-agerpenak zenbatzen ditu, hitzen arteko ordena eta harremana kontuan hartu gabe. Batzuetan hitzen ordena garrantzitsua izaten da, eta hori mantentzeko ”n-grama ereduak” (*n-gram*) erabili daitezke. Dena dela, proiektu honetan bide horretatik joatearen ideia baztertu egin da. Beraz, hemen hitz-agerpenak bakarrik zenbatu dira, besterik ez.

5.3 Irudian hitz-zaku baten adibide bat erakusten da. Esaldiak bektore bihurtzen dira, posizio bakoitzean esaldiko hitz bakoitzaren agerpen-kopurua adieraziz, eta bektore horiekin hitz-zakua osatzen da. Bektoreen luzera zakuko hitz desberdin kopurua izango da, eta bakoitzak falta dituen hitzen lekuan zeroak jarriko dira.

```
"Mirai neska da. Isabel ere neska da."
Hitz-zaku-1 = {"Mirai":1,"neska":2,"da":2,"Isabel":1,"ere":1}
"Xabier ez da neska, mutila da."
Hitz-zaku-2 = {"Xabier":1,"ez":1,"da":2,"neska":1,"mutila":1}
"Mirai neska da. Isabel ere neska da. Xabier ez da neska, mutila da."
Hitz-zaku-3 = {"Mirai":1,"neska":3,"da":4,"Isabel":1,"ere":1,"Xabier":1,"ez":1,"mutila":1}
(1) [1,2,2,1,1,0,0,0]
(2) [0,1,2,0,0,1,1,1]
(3) [1,3,4,1,1,1,1,1]
```

5.3 Irudia: Hitz-zaku baten sorreraren adibidea.

WEKAn hitz-zakuak sortzerakoan, ARFF fitxategia goitik behera aldatzen da. Hasieran @data atalean esaldi bakoitza eta bere sentimendua agertzen zen. Hitz-zakuaren filtroa aplikatu ondoren, ordea, @data ataleko esaldiak zenbakizko bektoreekin ordezkatzen dira, aurreko irudian ikusten den bezala. Gainera, hitz desberdin bakoitzeko @attribute lerro bat sortzen da.

Sortutako bektoreen luzera atributu-kopuru +1 izango da. +1 hori sentimendu-klasea ere gehitzen delako da. Eta logikoa den bezala, bektoreak gehienbat zeroz beteta egongo dira, atributu gehienak ez direlako esaldi guztietan agertzen. Adibide bat jartzearren, 7 instantziako eta 15 atributuko hitz-zaku batek 5.4 Irudian ikusten den itxura edukiko luke.

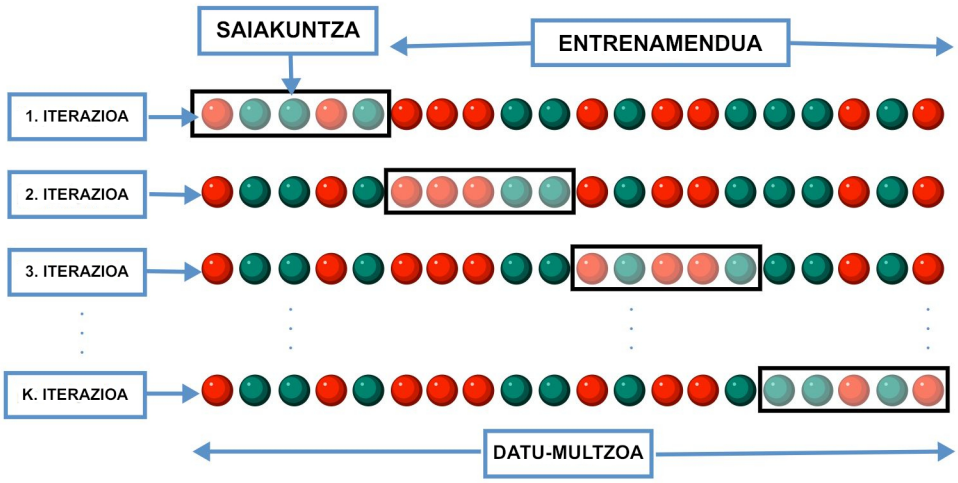
```
          Hitz1,Hitz3,Hitz4,Hitz5...Hitz15,Klasea
Iruzkin1: [1,2,2,1,1,0,0,0,0,0,0,0,0,0,0]+positibo
Iruzkin2: [0,1,2,0,0,1,1,1,0,0,0,0,0,0,0]+negatibo
Iruzkin3: [1,3,4,1,1,1,1,1,0,0,0,0,0,0,0]+neutro
...
Iruzkin7: [0,0,1,0,1,0,0,0,0,0,1,0,0,2,1]+neutro
```

5.4 Irudia: Proiektuan sortutako hitz-zakuen itxura.

Izatez, ARFF fitxategia zabaltzen bada, bektoreak beste modu batean adierazita daudela ikusiko da. Zeroak ez dira agertzen, bektoreen luzera esaldi bakoitzak jatorriz zeukan hitz-kopurua gehi sentimendu-klasea da, eta bektoreko gelaxka bakoitzean zenbatgarren atributua dagoen agertzen da. Baina WEKAn sartu eta fitxategia kargatzen bada, "Edit" botoia sakatuz, 5.4 Irudian agertzen den bezala ikusiko da, beraz hobe da modu horretan ulertzea.

5.3 Balioztatze gurutzatua

Balioztatze gurutzatuaren (*cross-validation*) bidez, datu-multzoa kopuru (*fold*) jakin batean zatitzen da, eta kopuru bereko iterazioak egiten dira. Iterazio bakoitzean zati desberdin bat erabiliko da saiakuntza (*testing*) moduan, eta gainerako zatiak entrenamendu (*training*) moduan erabiliko dira. Prozesu hori 5.5 Irudia ikusiz uler daiteke. Iterazio guztiak egin ondoren, emaitza erabakitzeko, guztiaren arteko batezbestekoa egiten da.c



5.5 Irudia: Balioztatze gurutzatuaren funtzionamendua, *k* iteraziorekin (*k-fold*).

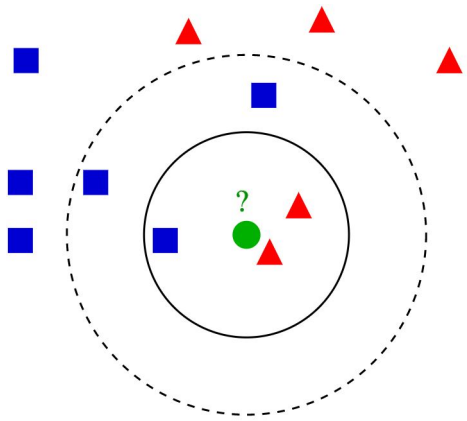
Proiektu honetan 10 iterazioko balioztatze gurutzatua erabili da, hau da, *10 fold cross-validation* deritzona.

5.4 Sailkatzaileak

Sailkatzaileak instantziak sailkatzen dituzten algoritmoak dira, instantzia bakoitzari klase bat iragarriz. Proiektu honetan, aurretik azaldu den sailkapen gainbegiratua erabili da. Horretarako metodo asko dauden arren, txosten honetan proiektuan erabilitakoak bakarrik daude azalduta.

5.4.1 k-NN

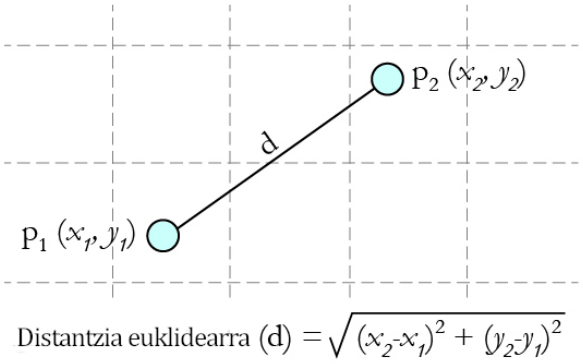
k-NN (*Nearest Neighbours*, hau da, auzokide hurbilenak) metodoa oso algoritmo erraz eta intuitiboa da. Kasu bati zein klase esleitu erabakitzeko, kasu horretatik hurbilen dauden *k* auzokideak kontuan hartzen dira. *k*=3 bada, 3 auzokide hurbilenak hartuko dira. *k*=5 bada, 5 auzokide hurbilenak. Kasu berriari ezarriko zaion klasea, auzokideek duten klaseen artean gehien errepikatzen dena izango da.[13]



5.6 Irudia: Kasu berri bati *3-NN* eta *5-NN* erabiliz klasea esleitzen.

5.6 Irudian ikusten den bezala, barruko zirkulua *3-NN* da eta kanpoko zirkulua *5-NN* da. *3-NN*ren kasuan bi gorri eta urdin bat daude, beraz kasu berriari klase gorria esleituko zaio. *5-NN*ren kasuan, berriz, bi gorri eta hiru urdin daude, beraz kasu berriari klase urdina esleituko zaio.

Auzokide hurbilenak zeintzuk diren jakiteko haien arteko distantzia kalkulatu beharra dago, eta horretarako, normalean, distantzia euklidearra erabiltzen da. 5.7 Irudian, bi punturen arteko distantzia kalkulatzeko formula matematikoa adierazten da.



5.7 Irudia: Distantzia euklidearraren formula.

Metodo honek ikasketa alferra (*lazy learning*) egiten du, ikasketa-fasean ez duelako eredurik induzitzen. Kasu berrien klasea zuzenean datu-basea erabiliz iragartzen du. Ikasketa mota hau datu-basea etengabe eguneratzen ari den kasuetan oso praktikoa da, horrela ikasketa-faseko datuak ez direlako zaharkituta geratuko.

Lan honetan *3-NN* eta *7-NN* erabili dira. Sailkatzaile hau WEKAko "lazy" karpetan aurki daiteke. *k* kopurua sailkatzailearen aukeretan zehaztu daiteke.

5.4.2 NaiveBayes

Sailkatzaile hau eredu probabilistiko bat da. Hau da, probabilitatean oinarritzen da. Gertakizun bat gauzatu dela jakinda, beste gertakizun bat gauzatzeko probabilitea zein den adierazten du.[14] Horretarako, Bayesen teorema eta klasearen balioa emanda aldagai iragarleak burujabeak direla dioen hipotesia hartzen ditu oinarri. 5.8 Irudiak Bayesen teoremaren formula erakusten du.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

A gertatzeko probabilitatea = $P(A)$
B gertatzeko probabilitatea = $P(B)$
A gertatu izanik, B gertatzeko probabilitatea = $P(B|A)$
B gertatu izanik, A gertatzeko probabilitatea = $P(A|B)$

5.8 Irudia: Bayesen teorema.

Esan bezala, metodo honetan ezaugarri guztiak elkarren independenteak direla suposatzen da, klasearen balioa emanda. Esate baterako, fruta batek sagar bat izateko eduki behar dituen ezaugarriak borobila izatea, gorria izatea eta 10 zentimetroko diametroa edukitzea direla esan daiteke. *Naive Bayes*en arabera, ezaugarri horietako bakoitzak modu independentean eragiten du edozer gauza sagar bat izateko probabilitatean. Horrela izanik, koloreari, formari eta diametroari buruzko informazioa ematen duten aldagaien artean korrelaziorik ez dagoela suposatzen da.

Aldagaiak elkarren artean independenteak direla suposatzeak sinplifikazio bat eragiten du, eta hori abantaila bat da, sailkapenerako behar diren parametroak estimatzeko entrenamendu-datu kopuru txikia behar duelako. Hortik datorkio *NaiveBayes* izena, "Bayes inuzentea" ingelesez. Sailkatzaile hau WEKAko "bayes" karpetan aurki daiteke.

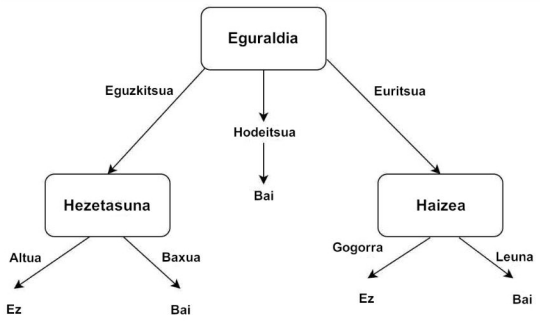
5.4.3 J48

J48 sailkatzailea *C4.5* algoritmoaren WEKArako Java-inplementazio bat da. Algoritmo hau Ross Quinlan informatikariak asmatu zuen, eta berak aurretik egindako *ID3* (*Iterative Dichotomiser 3*, hau da, dikotomia-gile iteratiboa 3) algoritmoaren hobekuntza bat da.[15] Algoritmo honek, sailkapena egiteko, erabaki-zuhaitz bat sortzen ditu.

Zuhaitz hau sortzeko, lehen gauza datu-multzoko atributu bakoitzaren irabazi-ratioa kalkulatzea da. Horretarako, aurretik atributu bakoitzaren informazio-entropia kalkulatu behar da, hau da, atributu bakoitzaren ziurgabetasuna. Hori eginda, irabazi-ratio handiena duen atributua zuhaitzaren erro-nodo gisa jarriko da, eta atributu horren erabaki posibleen arabera, nodo horri ume-nodoak gehituko zaizkio. Ume-nodoei gauza bera egingo zaie, eta horrela jarraituko da modu errekurtsiboan zuhaitz osoa osatu arte.

Eraikitze-prozesuan inausketa ere egin daiteke, hau da, informazio gehigarri askorik ematen ez duten adarrak moztea, zuhaitza sinplifikatzeko. Adar bat inausi ala ez erabakitzeko, errore-ratioa kalkulatzen da. Labur esanda, adarra hedatuta mantentzen bada edukiko den errorea eta adarra inausten bada edukiko den errorea alderatzen dira, eta inaustearen errorea txikiagoa bada adarra inausiko da.

	Eguraldia	Tenperatura	Haizea	Hezetasuna	Kalera irten
kasu1	Eguzkitsua	Beroa	Leuna	Altua	Ez
kasu2	Eguzkitsua	Beroa	Gogorra	Altua	Ez
kasu3	Hodeitsua	Beroa	Leuna	Altua	Bai
kasu4	Euritsua	Epela	Leuna	Altua	Bai
kasu5	Euritsua	Hotza	Leuna	Baxua	Bai
kasu6	Euritsua	Hotza	Gogorra	Baxua	Ez
kasu7	Hodeitsua	Hotza	Gogorra	Baxua	Bai
kasu8	Eguzkitsua	Epela	Leuna	Altua	Ez
kasu9	Eguzkitsua	Hotza	Leuna	Baxua	Bai
kasu10	Euritsua	Epela	Leuna	Baxua	Bai
kasu11	Eguzkitsua	Epela	Gogorra	Baxua	Bai
kasu12	Hodeitsua	Epela	Gogorra	Altua	Bai
kasu13	Hodeitsua	Beroa	Leuna	Baxua	Bai
kasu14	Euritsua	Epela	Gogorra	Altua	Ez



5.9 Irudia: Erabaki-zuhaitz baten osaketa, erabaki-taula batetik abiatuta.

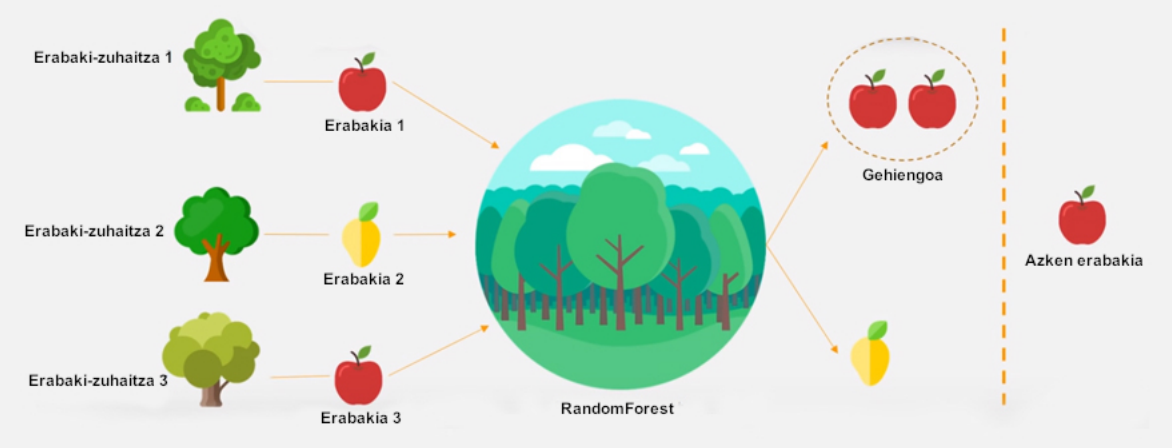
Adibidez, 5.9 Irudiko adibidean, eguraldia hodeitsua denean iragarpena ”Bai” izango da beti. Baina eguzkitsua edo euritsua bada, gainerako atributuen arabera, iragarpena desberdina izan daiteke. Adibide honetan, eguraldi euritsua haizeak bakarrik baldintzatzen du. Eguraldi eguzkitsua, berriz, hezetasunak bakarrik. Tenperaturak, berriz, ez du ezer baldintzatzen, eta beraz ez da zuhaitzean agertzen.

Adibidea jarraituz, ”kasu15=(Eguzkitsua,Hotza,Gogorra,Baxua)” kasu berriari iragarpena egiterakoan, emaitza ”Bai” izango litzateke.

Sailkatzaile hau WEKAko ”trees” karpetan aurki daiteke.

5.4.4 RandomForest

Sailkatzaile hau *J48*ren oso antzekoa da, baina *J48* ”zuhaitz” bat den bitartean, *RandomForest* ”baso” bat da. *RandomForest*ek, datu-multzo osoarekin erabaki-zuhaitz bat sortu beharrean, ausazko hainbat atributu-multzo hartzen ditu eta multzo bakoitzarekin erabaki-zuhaitz bat osatzen du. Ondoren, erabaki-zuhaitz guztiak konbinatzen ditu, ”baso” bat sortuz.[16] Hortik dator *RandomForest* izena, ”ausazko basoa” ingelesez.



5.10 Irudia: *RandomForest* basoko legea: gehiengoak agintzen du.

5.10 Irudian ikusten den bezala, baso barruko zuhaitz bakoitzak erabaki bat hartzen du, eta amaieran gehiengoak dauka hitza. Horixe da, hitz gutxitan, sailkatzaile honek egiten duena.

RandomForest, izatez, ”Bagging” motako sailkatzaile bat da, eta beraz, multisailkatzaile moduko bat dela ere esan daiteke. Izan ere, hainbat zuhaitz txiki konbinatzen ditu sailkatzaile indartsu bihurtzeko. ”Bagging” eta multisailkatzaileei buruz gehiago jakiteko, jo 5.5.1 kapitulura.

Sailkatzaile hau ere, *J48* bezala, WEKAko ”trees” karpetan aurki daiteke.

5.4.5 SMO

SMO (*Sequential Minimal Optimization*, hau da, optimizazio minimo sekuentziala) optimizazio problemak ebazteko algoritmo iteratibo bat da. John Plattek asmatu zuen, Microsofterako lan egiten duen informatikari batek.[17]

Sailkatzaile honek heuristikoak erabiltzen ditu, hau da, problemen ebazpenak, ikasketak eta aurkikuntzak egiteko teknikak. Heuristikoen bidez problema nagusia problema txikiagoetan zatitzen du, eta zati bakoitza analitikoki ebazten du.

SMO algoritmoa euskarri bektoredun makinak (*SVM*, hau da, *Support Vector Machine*) entrenatzeko erabiltzen da. Aldi berean, makina hauek sailkapenerako erabiltzen diren algoritmoak dira.

Sailkatzaile hau WEKAko ”functions” karpetan aurki daiteke.

5.4.6 DecisionTable

DecisionTable, izenak dioen bezala, erabaki-taula bat da. Erabaki-zuhaitzen antzekoak dira, baina zuhaitz bat irudikatu beharrean taula erraz bat bakarrik irudikatzen da.[18]

	ARAUAK								
BALDINTZAK	1	2	3	4	5	6	7	8	9
Bero egiten du	X	X	-	-	X	-	-	-	X
Euria egiten du	-	-	X	X	-	-	-	X	-
Uda da	X	-	X	X	-	X	X	-	X
Lagunak libre daude	-	-	X	-	-	X	X	-	X
Azterketak laster dira	-	X	-	-	X	-	X	X	X
ONDORIOAK	1	2	3	4	5	6	7	8	9
Etxean geratu			X	X				X	
Parkera joan						X	X		
Liburutegira joan		X			X				
Hondartzara joan	X								X

5.11 Irudia: Erabaki-taula baten adibidea.

5.11 irudian ikus daitekeen bezala, taulak hainbat baldintza, ondorio eta arau dauzka. Baldintzen arabera ondorioak desberdinak dira, eta baldintza-konbinazio bakoitza arau bat da. Kasu berri bati iragarpena egiteko, kasu horrek zein arau betetzen duen begiratu beharko da, eta arau horren ondorioa iragarri.

Sailkatzaile hau WEKAko ”rules” karpetan aurki daiteke.

5.4.7 RepTree

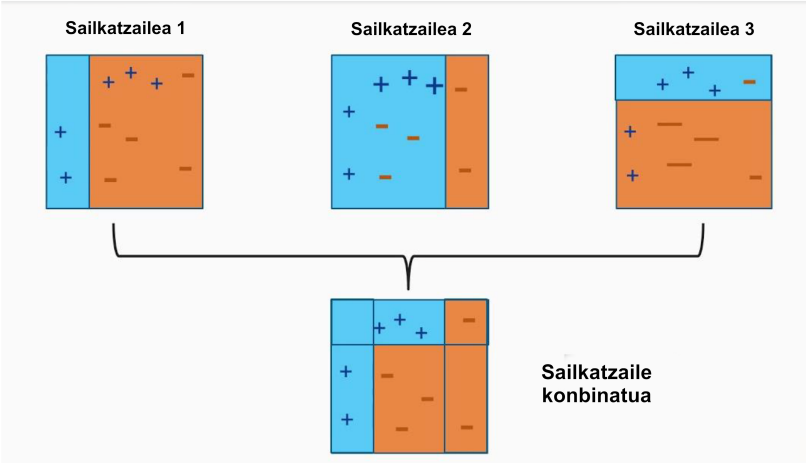
REPTree (*Reduced Error Pruning Tree*, hau da, errore gutxituko inausketadun zuhaitza) erabaki-zuhaitzen ikasketan erabiltzen den algoritmo bizkor bat da, *C4.5* (*J48*) algoritmoan oinarrituta. Honek ere erabaki-zuhaitz bat eraikitzen du, informazio-entropia eta ”errore gutxituko inausketa” deituriko inausketa-metodo erraz eta bizkor bat erabiliz.[19]

Inausketa mota honetan, hostoetatik hasita, nodo bakoitzaren azpizuhaitza inausi eta nodoa hosto bihurtzen da, nodoari klase ohikoena esleituz. Baina inausketa hauetako bakoitza egin aurretik, inausketa egiteak merezi duen ala ez begiratzen da. Hau da, inausketa bat egin ondoren geratuko den zuhaitza inausketaren aurretik dagoena bezalakoa edo hobea bada soilik egingo da inausketa hori.

Sailkatzaile hau WEKAko ”trees” karpetan aurki daiteke, *J48* eta *RandomForest*ekin batera. Sailkatzaile hau, proiektu honetan, multisailkatzaileak aplikatzerakoan bakarrik erabili da.

5.5 Multisailkatzaileak

Multisailkatzaileak sailkatzaileen konbinaketak dira. Multisailkatzailea osatzen duten sailkatzaile bakoitzak bere iragarpena egiten du, eta ondoren iragarpen horiek konbinatzen dira azken iragarpena emateko.[20]



5.12 Irudia: Sailkatzaile-konbinaketaren adibide bat.

5.12 Irudian ikusten den bezala, sailkatzaile bakoitzak ez du oso emaitza ona ematen, baina hirurak konbinatzerakoan emaitza perfektua lortzen da. Horixe da multisailkatzaileen helburua, hitz gutxitan esanda.

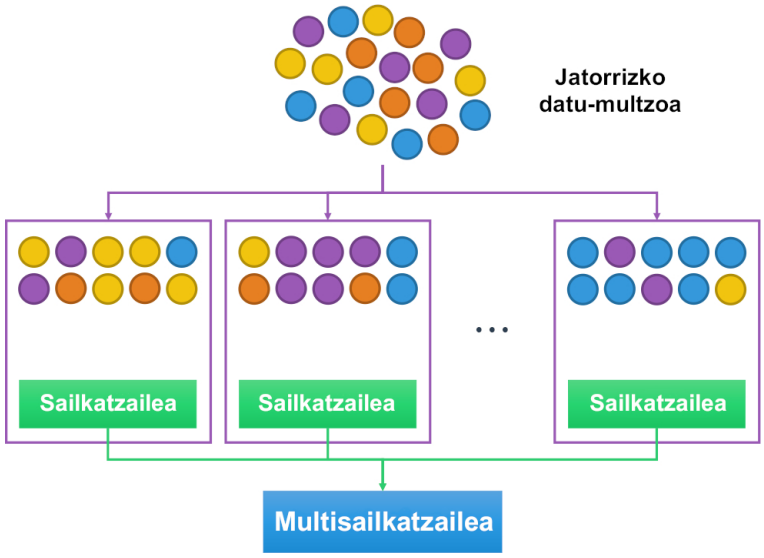
Proiektu honetan *Bagging* eta *Boosting* metodoak erabili dira. Bi metodoak oso antzekoak dira, eta biek sailkatzaile ahulak konbinatzen dituzte. Sailkatzaileak indibidualki ahulak izan arren, elkartzen badira, sailkatzaile indartsu bat osatzen dute. Sailkatzaile ahulak, ahulak izan arren, ausazko erabakiak baino hobeak dira. Horregatik, sailkatzaile indartsu bat osatzeko aproposak dira.

Sailkatzaile hauek WEKAko ”meta” karpetan aurki daitezke. Multisailkatzailearen aukeren barruan, sailkatzaile-algoritmo bat aukeratu behar da oinarri gisa. Orokorrean, erabaki-zuhaitzak aukeratzen dira.

5.5.1 Bagging

Bagging metodoa (”zakuraketa” ingelesez) multisailkaketa egiteko algoritmo bat da. Metodo honek dispertsioa, hau da, balioak elkarrengandik oso urrun egotea, gutxitzen du. Gainera, gehiegizko entrenamendua gertatzea ere saihesten laguntzen du. Gehiegizko entrenamendua uneko datuentzako eredu perfektuegi bat lortzea da. Datu berriak sartzerakoan eredu hori ez litzateke oso eraginkorra izango, horregatik hobe da eredu orokor bat lortzea eta ez perfektu bat.

5.13 Irudian, *Bagging* metodoak nola funtzionatzen duen azaltzen da, eskema sinple baten bidez.



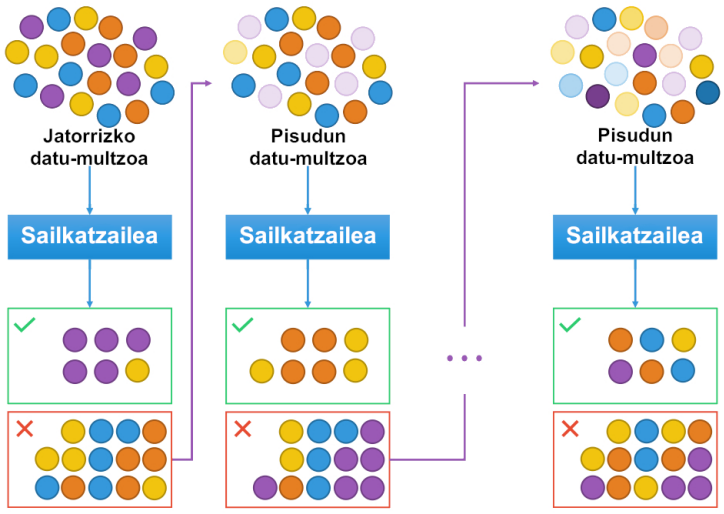
5.13 Irudia: *Bagging* metodoaren eskema.

Metodo honek, jatorrizko datu-multzotik abiatuta, ausazko hainbat entrenamendu-multzo sortzen ditu, laginketa bidez eta ordezkapenak erabiliz. Multzo horiek paraleloki sailkatzen dira, eta bakoitzak iragarpen bat ematen du. Gehien iragarri den klasea izango da azken iragarpen gisa hautatuko dena.

5.5.2 AdaBoostM1

AdaBoost (Adaptive Boosting, hau da, bultzada moldagarria) *Boosting* metodoaren inplementazio bat da, eta izenak dioen bezala, adaptatu egiten da. Erroreak kontuan hartzen ditu, eta hobetzen saiatzen da.

5.14 Irudian, *Boosting* metodoak nola funtzionatzen duen azaltzen da, eskema sinple baten bidez.



5.14 Irudia: *Boosting* metodoaren eskema.

Metodo honek ere, jatorrizko datu-multzotik abiatuta, hainbat entrenamendu-multzo sortzen ditu, laginketa bidez eta ordezkapenak erabiliz. Baina *Bagging*ek ez bezala, *Boosting*ek sekuentzialki sailkatzen ditu multzoak, eta multzo bat sortzeko aurrekoan lortutako emaitzak hartzen dira kontuan.

Multzo bat sailkatzerakoan instantziei pisuak esleitzen zaizkie, ongi ala gaizki sailkatu diren arabera, eta sailkapenarekin lortu den eredua aurrekoa baino hobea den ala ez ebaluatzen da, eredu horri ere beste pisu bat ezarriz. Gaizki iragarri diren instantziei pisu gehiago ematen zaie, eta beraz, hurrengo multzoan agertzeko eta berriro sailkatuak izateko aukera handiagoa daukate, hurrengo sailkatzaileek kasu horiek landu ditzaten. Sailkapenean lortutako ereduek ordea, zenbat eta hobeak izan orduan eta pisu handiagoa izango dute. Azken iragarpena egiteko, pisu gehien duten ereduek ematen duten iragarpena hartuko da gehienbat kontuan.

5.6 Azpimultzoak sortzeko atributu-aukeraketak

Datu-multzo osoa sailkatu beharrean, batzuetan praktikoagoa izaten da datu batzuk baztertzea eta multzo txikiago batekin jardutea. Horretarako badira hainbat metodo, datu-multzotik datu jakin batzuk bakarrik aukeratzeko. Metodo hauek ebaluatzaile eta bilatzaile batez osatuta daude. WEKAn, metodo hauek datu-basearen gainean aplikatzen diren ”filtroak” dira.[21]

5.6.1 Ebaluatzaileak

Ebaluatzaileek datu-multzoko atributuak ebaluatzen diztuzte, ondoren bilatzaileek horietako batzuk aukeratzeko. Ebaluatzaile bakoitzak ebaluaketa egiteko bere irizpideak dauzka. Proiektu honetan erabili diren ebaluatzaileak honako hauek izan dira:

– *CfsSubsetEval*: Atributu bakoitzaren banakako iragarpen-gaitasuna eta atributuen arteko erredundantziamaila kontuan hartuta, azpimultzo baten balioa ebaluatzen du. Klasearekiko korrelazio altua duten baina elkarrekiko korrelazio baxua duten atributuzko azpimultzoak nahiago izaten dira.

– *LatentSemanticAnalysis*: Datuen eraldaketa eta ezkutuko analisi semantikoa eginez, atributu bakoitzaren balioa ebaluatzen du. Semantikoki oso antzekoak diren hitzak antzeko esanahia duten testuetan agertzen direla suposatzen du.

– *ReliefFAttributeEval*: Instantzien laginketa behin eta berriro errepikatuz, eta klase bereko eta desberdineko instantzia hurbilenetako atributuen balioak kontuan hartuz, atributu bakoitzaren balioa ebaluatzen du.

– *SymmetricalUncertAttributeEval*: Klasearekiko ziurgabetasun simetrikoa kontuan hartuta, atributu bakoitzaren balioa ebaluatzen du.

5.6.2 Bilatzaileak

Bilatzaileak bilaketa-algoritmoak dira. Atributu-aukeraketan, datu-multzoaren barruan baldintza jakin bat betetzen duten atributuak bilatzeko erabiltzen dira. Metodo bakoitzak bere baldintzak izaten ditu. Proiektu honetan erabili diren bilatzaileak honako hauek izan dira:

– *BestFirst* (”onena hasieran” ingelesez): Metodo hau ”SubsetEval” motako ebaluatzaileekin bakarrik erabili daiteke, hau da, azpimultzoak ebaluatzen dituzten metodoekin bakarrik. Datu-multzoko atributu onenak bilatzen ditu, eta gainerakoak baztertzen ditu. Horretarako *Greedy* (”gutiziatsua” edo ”irenskorra”) algoritmo bat eta *Backtracking* (”atzera jotzea”) estrategia erabiltzen ditu.

– *Ranker* (”mailakatzailea” ingelesez): Metodo hau ”AttributeEval” motako ebaluatzaileekin bakarrik erabili daiteke, hau da, atributuak ebaluatzen dituzten metodoekin bakarrik. Datu-multzoko atributuak mailakatzen ditu. Hau da, atributu bakoitzari, ebaluatzailearen partetik jaso duten ebaluaketaren arabera, nota bat jarriko dio. WEKAn algoritmo honi X kopuru bat adierazten zaio, azpimultzoa sortzeko X onenak bakarrik aukera ditzan.

5.7 SMOTE metodoa

SMOTE, hau da, *Synthetic Minority Oversampling TEchnique* (gutxiengo sintetikoen gehiegizko laginketaren teknikna) datuen analisian erabiltzen den teknika bat da.[22] Teknika honen helburua datu-multzo bateko klaseen banaketa orekatzea da, laginketa gehiago eginez, eta horretarako klase minoritarioko instantzia gehiago sortu behar dira.

Instantzia berri horiek sortzeko, klase minoritarioko entrenamendurako erabiltzen diren instantziak interpolatzen dira. Interpolazioa datu berriak sortzean datza, jadanik existitzen diren datuen joera errepikatuz.

WEKAn, SMOTE filtroa aplikatzen den bakoitzean, klase minoritarioko instantziak bikoiztu egiten dira. 5.15 Irudian horren adibide bat ikus daiteke.

	Positibo	Neutro	Negatibo
Jatorrizkoa:	2192	644	376
SMOTE behin:	2192	644	752
SMOTE birritan:	2192	1288	752
SMOTE hirutan:	2192	1288	1504

5.15 Irudia: Klase bakoitzaren instantzia-kopurua SMOTE erabilita.

Adibide hau garapenean erabilitako HOTELA datu-basearekin bat dator.

Gogoan eduki beharra dago atributu-kopurua eta instantzia-kopurua ez direla gauza bera. Atributuak, kasu honetan behintzat, hitz desberdin bakoitza dira. Instantziak, berriz, iruzkin-testu bakoitza dira. SMOTE aplikatzerakoan atributu-kopurua ez da aldatuko, hau da, ez da hitz berririk sortuko. SMOTEk egiten duena atributuak erabiliz esaldi berriak sortzea da, eta beraz, instantzia-kopurua handitzea.

5.8 TF-IDF metodoa

TF-IDF, hau da, *Term Frequency - Inverse Document Frequency* (terminoen maiztasuna - alderantzizko dokumentu-maiztasuna) zenbakizko estatistika bat da. Dokumentu bateko hitz bakoitzaren garrantzia kalkulatzeko erabiltzen da.[23]

Dokumentu guztietan hitz batzuk beste batzuk baino gehiago agertzen dira. Hizkuntza guztiek dauzkate ohikoagoak diren hitzak. Euskaraz, adibidez, ”eta” hitza askotan agertzen da edozein testutan. ”Errinozero” hitza, ordea, animalia jakin horri buruzko dokumentuetatik kanpo ez da batere ohikoa. Metodo honen helburua maiztasun horiek kontuan hartzea da, ondoren hitz-zaku egokiago bat sortzeko.

TF-IDF balioa, hitz bat dokumentuan agertzen den kopuruarekiko proportzionalki hazten da, eta hitz horrek dokumentuan daukan maiztasunekin orekatzen da, zeinak hitz batzuk, oro har sarriago agertzen direla erakusten duen.

Teknika honek bi estatistiko erabiltzen ditu. Alde batetik terminoen maiztasuna (*TF: Term Frequency*) dago, hitz jakin bat dokumentu batean zenbat aldiz agertzen den adierazten duena. Beste aldetik, alderantzizko dokumentu-maiztasuna (*IDF: Inverse Document Frequency*) dago. Azken honek, dokumentu batean maiz azaltzen diren hitzei pisua murrizten die, hitz ezohikoei pisua areagotu bitartean. TF-IDF metodoak bi estatistiko hauen arteko biderketa bat egiten du hitz bakoitzari balio bat emateko.

Baliabide estatistiko hau oso ohikoa da informazio-atzitze prozeduretarako eta testu-corpusetatik datuak biltzeko. Modu horretan, dokumentu esanguratsuak bilatu daitezke, eta hauek garrantzi handirik ez daukatenengandik bereizi. Google bezalako bilaketa-motorrek ere, besteak beste, metodo honetan oinarritutako algoritmo bat erabiltzen dute.

6. KAPITULUA

Garapena eta emaitzak

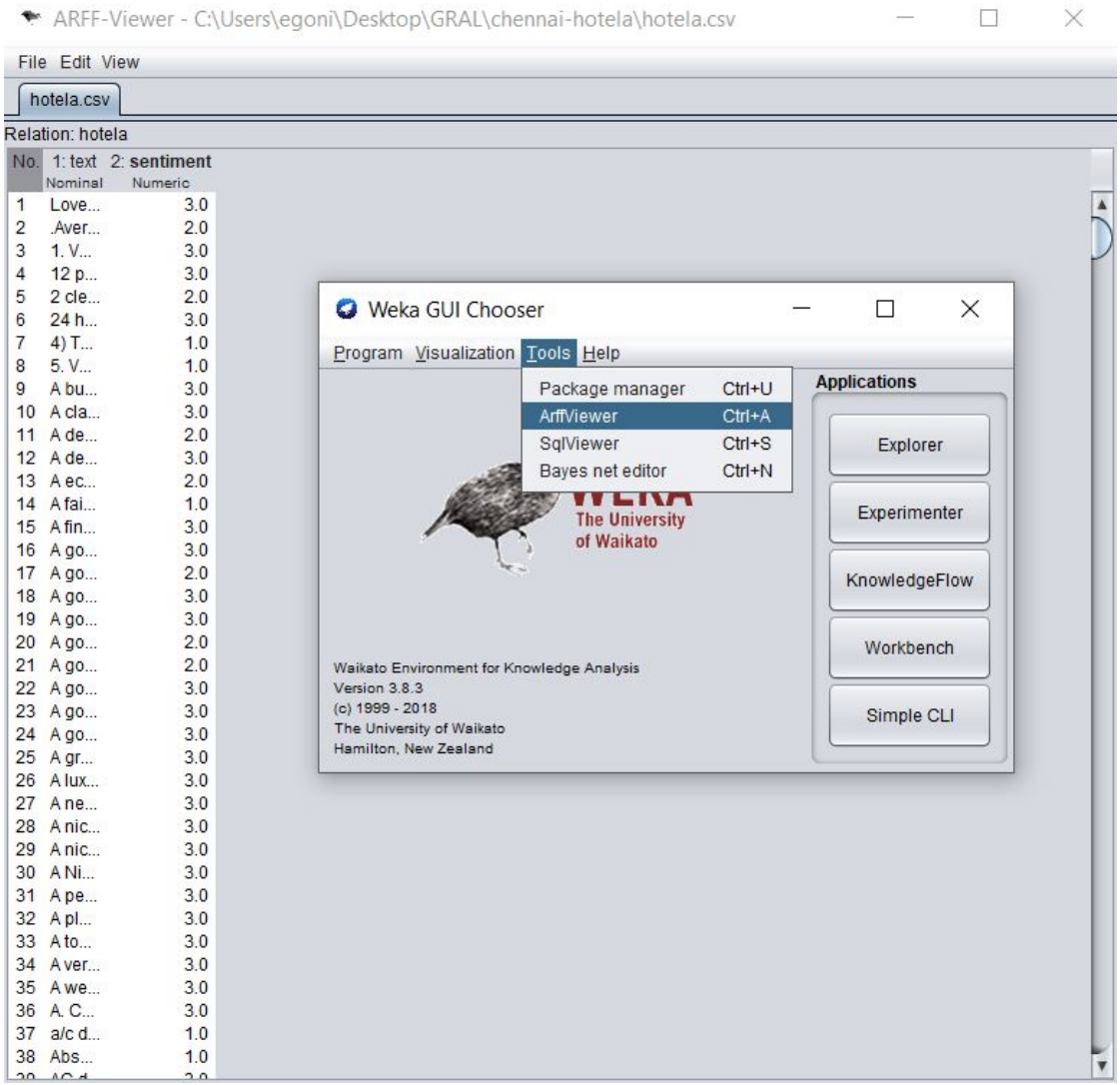
Proiektu hau aurrera eramateko eman diren pausoak urratsez urrats azalduko dira atal honetan, pauso bakoitzean lortutako emaitzekin batera. Lehenago ere aipatu den bezala, proiektu honetan garapena eta emaitzak atal desberdinetan banatzea nahasgarria izango litzatekeenez, elkarrekin jartzea erabaki da.

6.1 Sarrera-datuen bihurketa

Lehenengo pausoa, erabili beharreko datu-baseak WEKArekin bateragarriak egitea izan da. WEKAk ARFF fitxategiekin lan egiten du, baina datu-baseak CSV formatuan daude. Hortaz, CSVak ARFF bihurtu behar izan dira, WEKAk interpretatu ditzan.

CSV eta ARFF formatuen azalpena jakiteko jo berriro kontzeptuetara, zehazki 5.1 kapitulura.

Horretarako, WEKAren funtzio bat erabili da, interfaze nagusiko menu-barran dagoen ”Tools” ataleko ”ArffViewer” funtzioa hain zuzen. 6.1 Irudian funtzio honen pantaila ikus daiteke. Datu-basean akatsen bat baldin bazegoen, bihurketak huts egiten zuen. Baina hori gertatzerakoan, programak arazoa zein lerrotan zegoen adierazten zuenez, arazo puntual horiek Notepad++ekin berehala konpondu ahal izan dira.



6.1 Irudia: WEKAren "ArffViewer" interfazea.

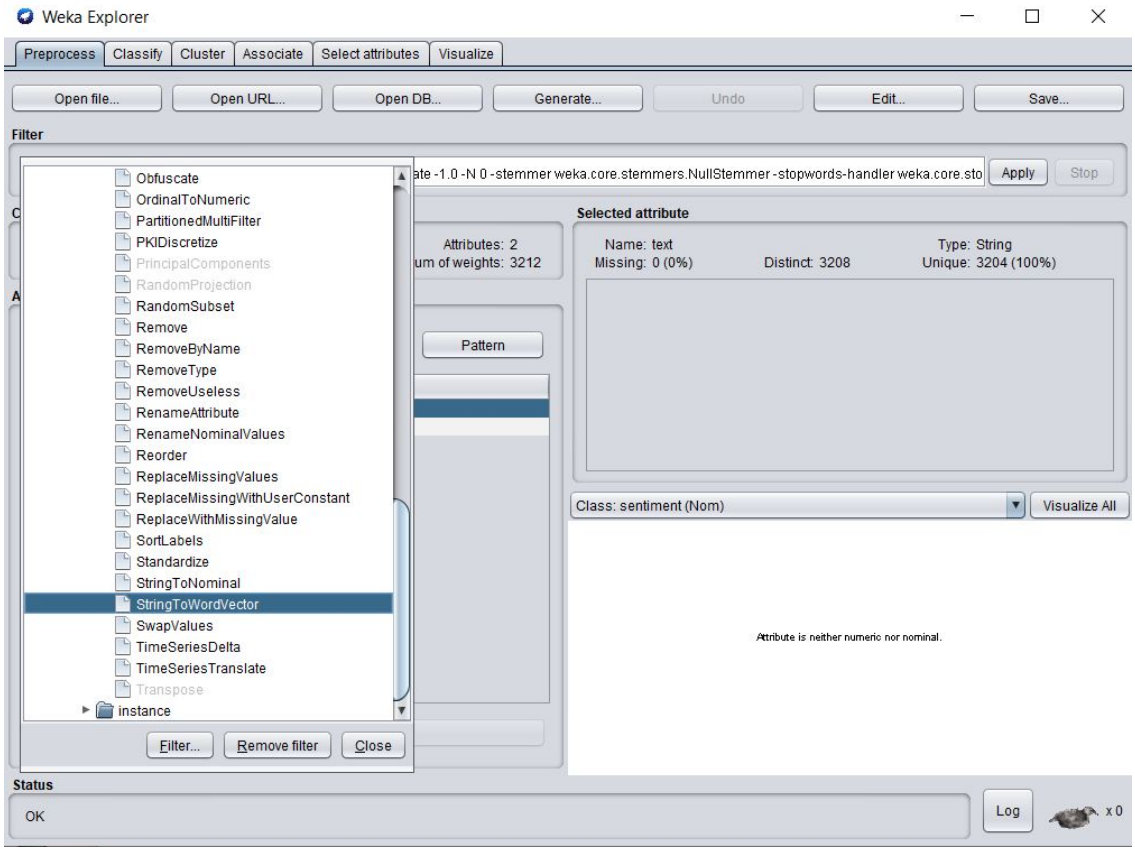
ARFFa sortu ondoren, atributuak defektuz gaizki zetozenez, eskuz zuzendu behar izan dira. Bai HOTELA eta bai POLITIKA datu-basean, lehen atributua *string* moduan adierazi behar izan da, eta bigarrena hiru sentimendu motak barne hartzen dituen bektore gisa.

Kontuan hartzekoa da datu-base bakoitzean sentimendu-klaseek ordena desberdina daukatela. Batean ordena "negatibo-neutro-positibo" da, eta bestean, berriz, "neutro-positibo-negatibo" da. Izatez txikikeria honek ez du ezer aldatzen, baina emaitzak apuntatzerakoan nahasketarik ez gertatzeko adi ibili beharra dago. Hau editatu egin zitekeen biak ordena berean edukitzeko, baina horretaz konturatzerako atzera egiteak ez zuen merezi.

6.2 Hitzak zaku batean sartzen

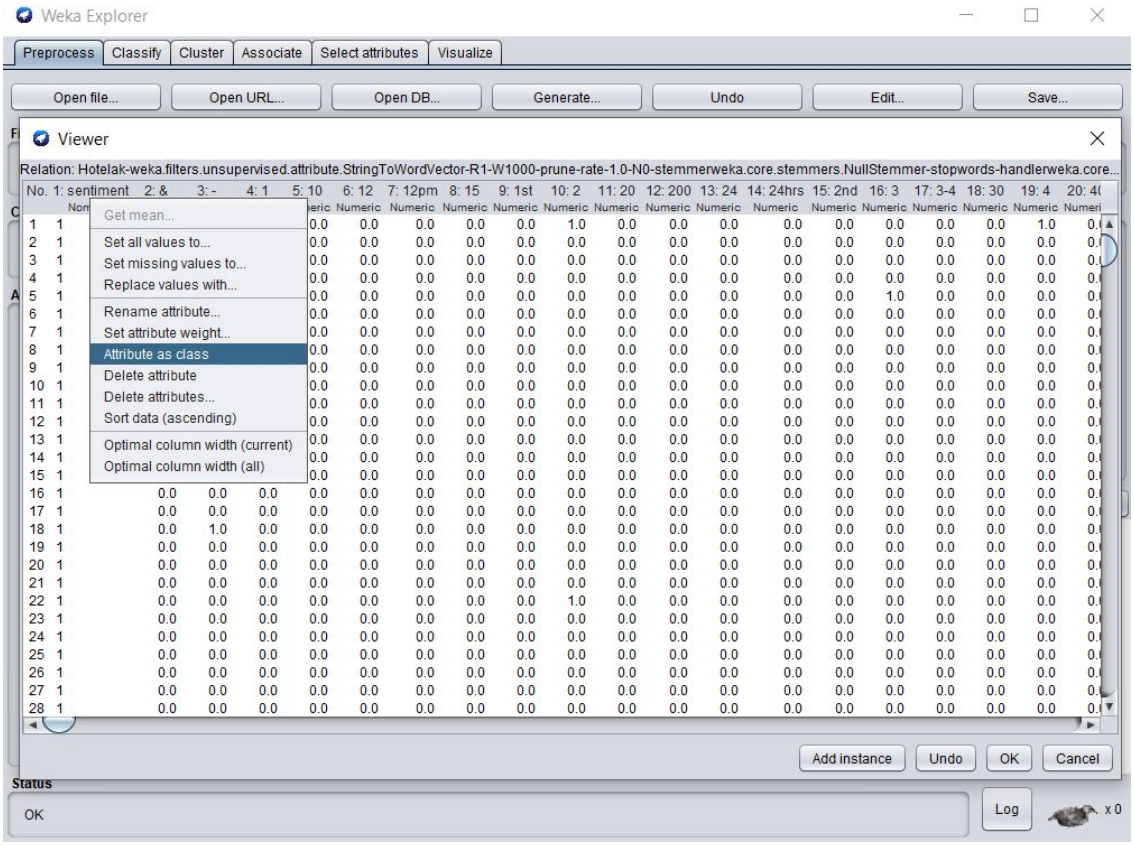
Behin ARFFa ongi edukita, hurrengo pausua hitz-zakua (*bag-of-words*) sortzea izan da. Honi buruzko azapena jakiteko jo berriro kontzeptuetara, zehazki 5.2 kapitulura.

Hasieran ARFF fitxategia WEKAn kargatu behar izan da. Horretarako, WEKAren ”Explorer” interfazera jo da, eta bertako ”Preprocess” atalean ”Open File...” egin da. Behin fitxategia kargatuta, hitz-zakua sortzeko filtroa aplikatu behar izan zaio. Filtro hori *StringToWordVector* da, eta ”unsupervised” karpeta barruko ”attribute” karpetan aurki daiteke, 6.2 Irudian adierazten den bezala.



6.2 Irudia: WEKAren ”Explorer” interfazean *StringToWordVector* filtroa bilatzen.

WEKAn hitz-zakua sortu ondoren, ordea, klasea defektuz atributu bezala agertzen da. Hori konpontzeko, WEKAren aukeren artean dagoen ”Edit” botoiari eman behar zaio, eta bertan, klasearen zutabearen gainean, ”Attribute as class” aukera sakatu, 6.3 Irudian adierazten den bezala.



6.3 Irudia: WEKAren ”Edit” funtzioarekin hitz-zakua zuzentzen.

Prozesu hau guztia bi datu-baseekin egin da, eta horrela bi hitz-zakuak lortu dira. Zaku horiek, WEKAren aukeren artean dagoen ”Save” botoiari emanez, fitxategi berri gisa gorde dira.

6.3 Sailkatzaileekin jolasean

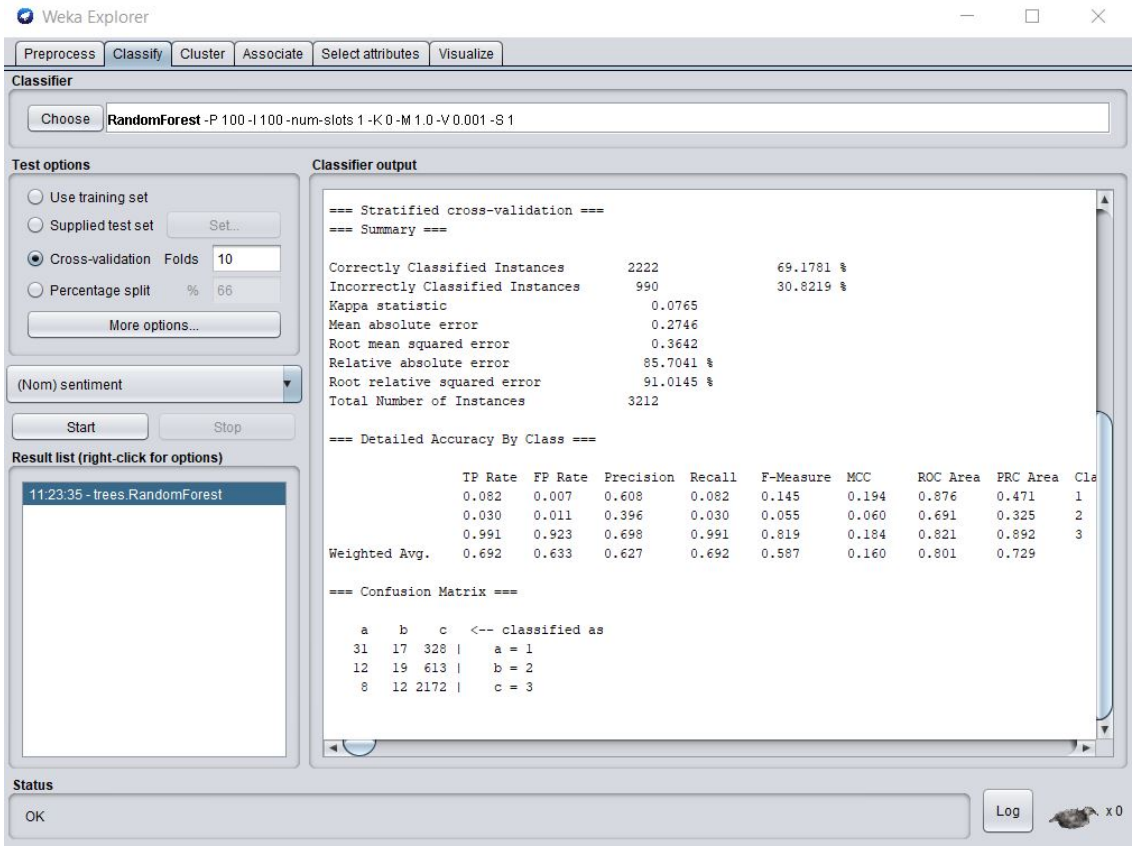
Hurrengo pausoa, hitz-zaku bakoitzari sailkatzaile desberdinak aplikatzea eta bakoitzaren asmatze-tasak ateratzea izan da. Horretarako, WEKAren ”Classify” atalean ”Choose” sakatu eta sailkatzaile-zerrendatik hainbat sailkatzaile desberdin aukeratuz probak egin dira. Proba guztiak 10 iteraziodun balioztatze gurutzatua (*10 fold cross-validation*) erabiliz egin dira.

Balioztatze gurutzatuari buruzko azalpena jakiteko jo berriro kontzeptuetara, zehazki 5.3 kapitulura.

Balioztatze gurutzatua defektuz 10 iteraziotan jarrita datorrenez, horretan ez da ezer aldatu behar izan. Sailkatzailea aukeratu ostean ”Start” sakatu eta apur bat itxaron ondoren sailkatzaile horrek emandako emaitzak

lortu dira. Prozesu hau sailkatzaile bakoitzarekin errepikatu da, eta sailkatzailearen eta datu-basearen arabera, itxaron beharreko denbora luzeagoa edo laburragoa izan da. Sailkatzaileak zenbat eta algoritmo konplexuagoa eduki, orduan eta denbora gehiago behar du, eta datu-basea zenbat eta handiagoa izan, orduan eta denbora gehiago behar da, baita.

6.4 Irudian, sailkatzaileak probatu eta emaitzak ematen dituen pantaila ikus daiteke.



6.4 Irudia: WEKArekin *RandomForest* sailkatzailea aplikatzen.

WEKAk datu asko ematen ditu emaitza gisa, baina proiektu honetarako asmatze-tasak bakarrik dira interesgarriak. Hau da, sailkatzaileak instantzia guztien ehuneko zenbat ongi sailkatu dituen klase bakoitzeko. Klaseak, lehen azaldu den bezala, positibo, neutro eta negatibo sentimenduak dira. Instantziak, berriz, datu-baseko lerro bakoitza, hau da, sentimendu bat daukan iritzi bakoitza.

Erabili diren sailkatzaileak 7 izan dira: *3-NN*, *7-NN*, *NaiveBayes*, *J48*, *RandomForest*, *SMO* eta *DecisionTable*. Sailkatzaile bakoitzaren azalpena jakiteko jo berriro kontzeptuetara, zehazki 5.4 kapitulura.

Sailkatzaile horiek aplikatu ondoren, lortu diren emaitzak 6.1 Taulan adierazi dira.

HOTELA datu-basea (aukeraketak)							
NORMAL 1651	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.912	0.981	0.722	0.826	0.991	0.818	0.932
Neutral	0.087	0.045	0.469	0.281	0.03	0.303	0.137
Negative	0.146	0.04	0.617	0.316	0.082	0.481	0.226
Avg.	0.657	0.683	0.659	0.657	0.692	0.676	0.69
POLITIKA datu-basea (aukeraketak)							
NORMAL 1266	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.84	0.913	0.928	0.58	0.754	0.783	0.638
Neutral	0.731	0.731	0.806	0.88	0.935	0.907	0.907
Negative	0.04	0	0.56	0.04	0.04	0.16	0.08
Avg.	0.683	0.703	0.817	0.673	0.762	0.772	0.713

6.1 Taula: HOTELA eta POLITIKA datu-baseetan salkatzaile bakoitza aplikatuz lortutako emaitzak.

Emaitzetan, berde argiarekin sailkatzaile bakoitzak gehien asmatu duen klasea adierazten da. Gehienetan klase positiboa izaten da nagusi, diferentzia nabarmenarekin. Klase bakoitzaren instantzia-kopuruak eragin handia dauka horretan. Izan ere, HOTELA datu-basean adibidez, 2192 positibo, 644 neutro eta 376 negatibo daude. Berde ilunarekin, berriz, sailkatzaile guztien artean lortutako emaitzarik onena azpimarratu nahi izan da.

Emaitzak ikusita, HOTELA datu-basearen kasuan onena *RandomForest* sailkatzailea izan dela ikus daiteke. Aipagarria da ere, *NaiveBayes* sailkatzailea klaseen arteko balantza orekatuena lortzen duena dela. Hau da, neutro eta negatibo klaseen tasa altuenak *NaiveBayes* erabilita lortu direla. POLITIKA datu-basean, berriz, *NaiveBayes* izan da nagusi zentzu guztietan.

POLITIKA datu-basean HOTELA datu-basean baino emaitza hobeak lortu dira, orokorrean %10-%15ean hobeto. Ondorioak ateratzeko goizegi den arren, horren arrazoia baliteke datu-baseen kalitatea izatea. HOTELA datu-basea ingelesez idatzita dauden Indiako hotel bateko iruzkinak dira, eta Indiako ingelesa gehienetan ez da oso ona izaten. Horrek eragina izan dezake beharbada.

Emaitzen taulako lehen zutabean ageri den ”normal” hitzak emaitza horiek datu-multzo osoarekin jardunez lortu direla esan nahi du. Aurrerago azpimultzoak erabiliko dira, eta ”normal” ordez azpimultzo bakoitza sortzeko erabilitako teknikaren izena idatziko da. Izen horren ondoan dagoen zenbakia, berriz, datu-multzo horrek daukan atributu-kopurua da. Atributuak hitz-zakuan dauden hitz desberdin bakoitza dira, eta zenbakiak kopuru hori adierazten du.

6.4 Azpimultzoak sortzen

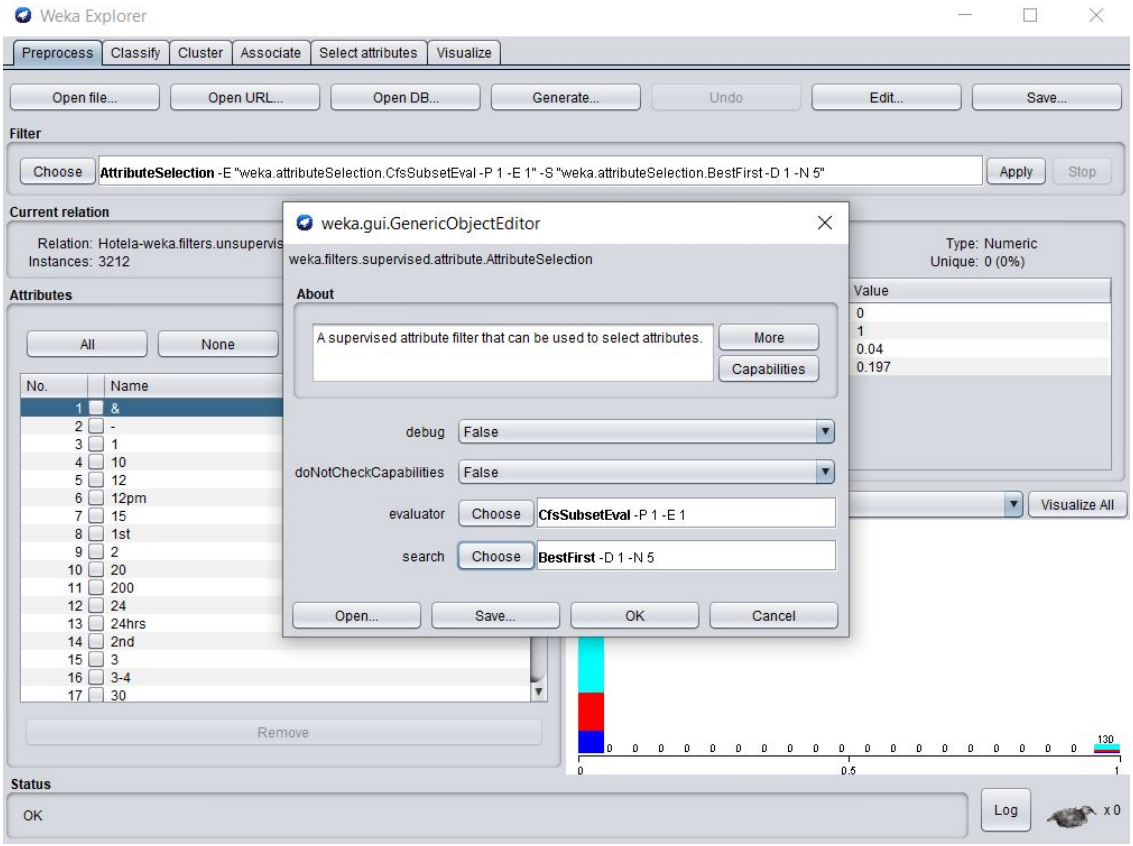
Emaitzak hobetu nahian, azpimultzo (*subset*) batzuk sortu eta horiekin probak egitea erabaki da. Azpimultzo horiek atributu-aukeraketaren (*attribute-selection*) bidez egin dira. Atributu-aukeraketa datu-basearen lagin txikiago bat hartzean datza, jatorrizko datu-basearen atributu guztiak erabili beharrean, kopuru jakin bat baka-

rrik erabiltzeko. Aukeraketa hori egiteko hainbat metodo desberdin erabili dira, beraz, metodo horien azalpena jakiteko, jo berriro kontzeptuetara, zehazki 5.6 kapitulura.

Egia esan beste atributu-aukeraketa batzuekin ere proba txikiren bat egin da, esate baterako *Clustering Variation* metodoarekin, baina oso emaitza txarrak lortzen zirela ikusita, metodo horiekin ez jarraitzea erabaki da.

Bestalde, POLITIKA datu-basean, atributu-aukeraketa batzuek ”cannot handle missing class values” errorea ematen zuten. Horren zergatia lerro batean klase-balioa falta zela da. Metodo batzuek halako errore txikiak arbuiatzen dituzten arren, beste batzuk oso zorrotzak izaten dira eta ez dute akatsik onartzen. Aurrera jarraitu ezinik, datu-basetik lerro hori ezabatu eta orain arteko pausu guztiak errepikatu behar izan dira.

Aukeraketa-metodoak aplikatzeko, ”Preprocess” ataleko filtroen artean ”AttributeSelection” aukeratzen da, ”supervised” karpetako ”attribute” karpetan dagoena hain zuzen. ”AttributeSelection” gainean sakatuta, ebaluatzaile (*evaluator*) eta bilatzaile (*search*) desberdinak aukera daitezke. Aukeraketa hori egiteko pantaila 6.5 Irudian ageri den interfazea da.



6.5 Irudia: WEKAn azpimultzoak aukeratzen.

Proiektu honetan erabilitako ebaluatzaile-bilatzaile konbinazioak 6.6 Irudian adierazi dira.

- *BestFirst*:
  evaluator=CfsSubsetEval + search=BestFirst
- *LatentSemantic*:
  evaluator=LatentSemanticAnalysis + search=Ranker (numToSelect=150)
- *Relief*:
  evaluator=ReliefFAttributeEval + search=Ranker (numToSelect=50)
- *SymmetricalUncert*:
  evaluator=SymmetricalUncertAttributeEval + search=Ranker (numToSelect=50)

6.6 Irudia: WEKAn aukeratutako azpimultzoak.

Datu-base bakoitzeko atributu-kopurua desberdina denez, kopuru handiena duen HOTELA datu-basean Relief eta SymmetricalUncert 75 eta 50 kopuruekin (*numToSelect*) egitea erabaki da, eta kopuru txikiena duen POLITIKA datu-basean, berriz, 50 eta 30 kopuruekin.

Atributu-aukeraketaren filtroa jatorrizko hitz-zakuaren gainean aplikatu da beti. Hau da, filtro bat aplikatu ostean, filtrodun fitxategia ”Save” botoia erabiliz fitxategi berri gisa gorde da, eta ondoren, ”Open file...” bidez jatorrizko hitz-zakua berriz kargatu da beste filtro bat aplikatzeko. Hau guztia eginez, azpimultzo bakoitzeko ARFF fitxategi bat lortu da. Hori guztia datu-base bakoitzeko.

Hurrengo pausoa azpimultzo bakoitza banan-banan WEKAn kargatu eta hasierako sailkatzaile berdinak aplikatzea izan da. Hori egin ondoren, lortu diren emaitzak 6.2 Taulan eta 6.3 Taulan adierazi dira.

HOTELA datu-basea (aukeraketak)							
NORMAL 1651	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.912	0.981	0.722	0.826	0.991	0.818	0.932
Neutral	0.087	0.045	0.469	0.281	0.03	0.303	0.137
Negative	0.146	0.04	0.617	0.316	0.082	0.481	0.226
Avg.	0.657	0.683	0.659	0.657	0.692	0.676	0.69
BESTFIRST 44	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.93	0.958	0.872	0.943	0.914	0.949	0.951
Neutral	0.141	0.118	0.238	0.143	0.193	0.158	0.116
Negative	0.266	0.21	0.529	0.274	0.332	0.319	0.263
Avg.	0.694	0.702	0.705	0.704	0.701	0.717	0.703
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.766	0.877	0.647	0.741	0.998	0.945	0.951
Neutral	0.172	0.146	0.267	0.283	0.005	0.172	0.096
Negative	0.351	0.245	0.548	0.205	0.005	0.38	0.106
Avg.	0.599	0.657	0.559	0.586	0.682	0.724	0.681
RELIEF 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.972	0.992	0.942	0.994	0.936	0.99	0.994
Neutral	0.039	0.025	0.076	0.006	0.061	0.011	0
Negative	0.035	0.003	0.048	0.003	0.077	0.021	0.005
Avg.	0.675	0.682	0.663	0.68	0.66	0.68	0.679
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.973	0.992	0.953	0.997	0.944	0.996	0.996
Neutral	0.042	0.023	0.054	0.002	0.051	0.009	0
Negative	0.027	0	0.027	0	0.056	0.008	0.005
Avg.	0.675	0.682	0.664	0.681	0.661	0.682	0.681
SYMMETRIC 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.913	0.959	0.942	0.917	0.922	0.917	0.949
Neutral	0.163	0.096	0.286	0.165	0.2	0.165	0.135
Negative	0.218	0.136	0.842	0.301	0.351	0.301	0.21
Avg.	0.682	0.69	0.698	0.694	0.711	0.694	0.699
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.925	0.96	0.86	0.928	0.906	0.952	0.946
Neutral	0.0112	0.093	0.281	0.172	0.194	0.14	0.146
Negative	0.314	0.218	0.535	0.301	0.362	0.343	0.237
Avg.	0.69	0.7	0.706	0.703	0.699	0.718	0.703

6.2 Taula: HOTELA datu-basean azpimultzo bakoitzarekin lortutako emaitzak.

POLITIKA datu-basea (aukeraketak)							
NORMAL 1266	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.84	0.913	0.928	0.58	0.754	0.783	0.638
Neutral	0.731	0.731	0.806	0.88	0.935	0.907	0.907
Negative	0.04	0	0.56	0.04	0.04	0.16	0.08
Avg.	0.683	0.703	0.817	0.673	0.762	0.772	0.713
BESTFIRST 24	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.623	0.551	0.913	0.638	0.797	0.797	0.638
Neutral	0.926	0.926	0.898	0.917	0.917	0.944	0.926
Negative	0.04	0	0.44	0.16	0.2	0.32	0.08
Avg.	0.713	0.683	0.847	0.728	0.787	0.817	0.723
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.913	0.855	0.652	0.696	0.522	0.725	0.855
Neutral	0.537	0.546	0.741	0.741	0.954	0.806	0.759
Negative	0.04	0	0.24	0.08	0	0.12	0
Avg.	0.604	0.584	0.649	0.644	0.688	0.693	0.698
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.29	0.232	0.391	0.333	0.391	0.29	0.072
Neutral	0.824	0.843	0.852	0.87	0.704	0.898	0.981
Negative	0	0	0.04	0.12	0	0	0
Avg.	0.54	0.53	0.594	0.594	0.51	0.579	0.55
RELIEF 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0	0	0.058	0	0.072	0	0
Neutral	0.991	1	0.935	1	0.944	0.963	1
Negative	0	0	0	0	0	0	0
Avg.	0.53	0.535	0.52	0.535	0.53	0.515	0.535
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.623	0.507	0.942	0.609	0.783	0.768	0.609
Neutral	0.926	0.944	0.898	0.917	0.917	0.926	0.917
Negative	0.04	0	0.56	0.2	0.24	0.28	0
Avg.	0.713	0.678	0.871	0.723	0.787	0.792	0.698
SYMMETRIC 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.652	0.565	0.812	0.594	0.768	0.725	0.638
Neutral	0.926	0.926	0.917	0.917	0.917	0.898	0.926
Negative	0.04	0	0.44	0.2	0.24	0.4	0
Avg.	0.723	0.688	0.822	0.718	0.782	0.777	0.713

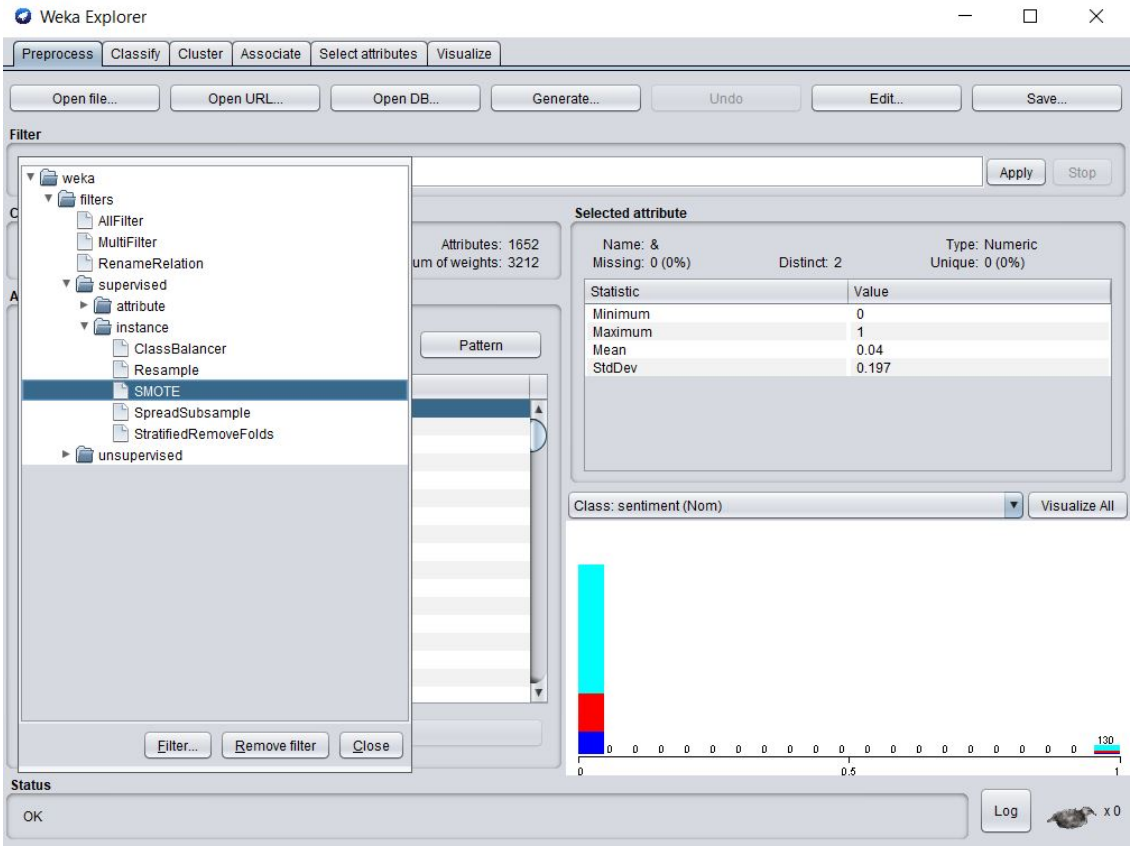
6.3 Taula: POLITIKA datu-basean azpimultzo bakoitzarekin lortutako emaitzak.

Ikus daitekeenez, metodo batzuekin emaitzak apur bat hobetzea lortu da. Bai HOTELA datu-basean eta bai POLITIKA datu-basean, *Relief* metodoarekin salbu, besteekin emaitza hobetzea lortu da. Hala ere, klaseen artean diferentzia handia egoten jarraitzen du, batez ere HOTELA datu-basean, non klase positiboak erabateko garaipena izan duen. Lehenago ere aipatu den bezala, honen arrazoi nagusia instantzia-kopurua da, klase bakoitzaren artean alde izugarria baitago. Baina badago arazo honi buelta emateko modu bat, hurrengo pausoan garatuko dena hain zuzen.

6.5 Datuak orekatzen

Datu-multzoak apur bat orekatu beharra zegoela erabaki da, eta horretarako SMOTE (*Synthetic Minority Oversampling TEchnique*) teknika ezin aproposagoa da. Teknika honek, instantzia gutxien dauzkan klasearen instantzia-kopurua bikoizten du. Metodo hau nola dabilen jakiteko jo berriro kontzeptuetara, zehazki 5.7 kapitulura.

SMOTE aplikatzeko, WEKAren ”Preprocess” ataleko filtroen artean ”SMOTE” aukeratzen da, ”supervised” karpeta barruko ”instance” karpetaren barruan hain zuzen, 6.7 Irudian ikusten den bezala.



6.7 Irudia: WEKAn filtroen artean SMOTE bilatzen.

Hori eginez, azpimultzo guztiei SMOTE aplikatu zaie, eta bakoitza fitxategi aparte moduan gorde da. Ondoren, fitxategi bakoitza banan-banan kargatu eta bakoitzari hasierako sailkatzaile berdinak aplikatu zaizkio.

SMOTE azpimultzo bakoitzari behin bakarrik aplikatu zaio. Izan ere, instantzia-kopurua handitzeak exekuzio-denbora ere nabarmen handitzen du. SMOTE bi aldiz aplikatzearen saiakera ere egin da, baina sailkatzaile batzuek ordu luzeak behar zituzten, eta kasurik okerrenean ordenagailuko RAM memoria agortu egiten zen,

”WEKA out of memory” errorea emanez. Hori dela eta, SMOTE fitxategi bakoitzeko behin bakarrik aplikatzea erabaki da.

Sailkatzaile guztiekin exekutatu ondoren, lortu diren emaitzak 6.4 Taulan eta 6.5 Taulan adierazi dira.

HOTELA datu-basea (aukeraketak + SMOTE)							
NORMAL 1651	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.693	0.73	0.731	0.824	0.992	0.808	0.948
Neutral	0.053	0.017	0.493	0.259	0.008	0.318	0.051
Negative	0.806	0.622	0.496	0.697	0.602	0.894	0.407
Avg.	0.602	0.579	0.637	0.696	0.734	0.738	0.673
BESTFIRST 44	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.907	0.93	0.903	0.902	0.898	0.92	0.947
Neutral	0.113	0.081	0.189	0.124	0.165	0.067	0.076
Negative	0.569	0.527	0.503	0.58	0.61	0.61	0.372
Avg.	0.694	0.693	0.691	0.695	0.706	0.702	0.67
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.552	0.565	0.622	0.733	0.978	0.894	0.879
Neutral	0.073	0.043	0.292	0.231	0	0.137	0.07
Negative	0.824	0.799	0.707	0.495	0.495	0.703	0.479
Avg.	0.523	0.52	0.581	0.593	0.701	0.718	0.65
RELIEF 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.937	0.966	0.812	0.952	0.914	0.974	0.99
Neutral	0.036	0.03	0.197	0.025	0.045	0.009	0.002
Negative	0.17	0.093	0.166	0.128	0.299	0.078	0.061
Avg.	0.615	0.615	0.566	0.613	0.629	0.613	0.618
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.943	0.971	0.646	0.966	0.926	0.984	0.994
Neutral	0.04	0.025	0.18	0.03	0.053	0.003	0.002
Negative	0.129	0.073	0.483	0.068	0.229	0.048	0.048
Avg.	0.61	0.613	0.528	0.61	0.623	0.612	0.617
SYMMETRIC 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.884	0.925	0.881	0.898	0.924	0.912	0.927
Neutral	0.124	0.067	0.252	0.16	0.16	0.12	0.053
Negative	0.642	0.576	0.524	0.59	0.688	0.668	0.399
Avg.	0.697	0.698	0.693	0.701	0.737	0.719	0.66
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.892	0.933	0.887	0.896	0.902	0.918	0.951
Neutral	0.102	0.07	0.186	0.134	0.191	0.084	0.059
Negative	0.594	0.533	0.537	0.596	0.632	0.649	0.366
Avg.	0.688	0.694	0.688	0.696	0.718	0.712	0.668

6.4 Taula: HOTELA datu-basean lortutako emaitzak.

POLITIKA datu-basea (aukeraketak + SMOTE)							
NORMAL 1266	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.652	0.71	0.71	0.667	0.812	0.783	0.58
Neutral	0.667	0.546	0.657	0.88	0.917	0.926	0.907
Negative	0.9	0.86	0.9	0.68	0.68	0.98	0.34
Avg.	0.714	0.665	0.727	0.771	0.833	0.894	0.683
BESTFIRST 24	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.609	0.565	0.884	0.565	0.71	0.667	0.493
Neutral	0.907	0.917	0.907	0.926	0.917	0.935	0.907
Negative	0.48	0.3	0.62	0.58	0.58	0.62	0.46
Avg.	0.722	0.674	0.837	0.74	0.78	0.784	0.683
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.913	0.884	0.667	0.551	0.594	0.696	0.754
Neutral	0.611	0.593	0.787	0.787	0.954	0.815	0.778
Negative	0.64	0.62	0.7	0.52	0.54	0.82	0.38
Avg.	0.709	0.687	0.731	0.656	0.753	0.78	0.683
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.275	0.203	0.261	0.319	0.391	0.232	0.058
Neutral	0.824	0.833	0.778	0.88	0.75	0.861	0.981
Negative	0.16	0.08	0.22	0.16	0.34	0.18	0
Avg.	0.511	0.476	0.498	0.551	0.551	0.52	0.485
RELIEF 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0	0	0.072	0	0.058	0	0
Neutral	0.87	0.889	0.935	1	0.843	0.954	1
Negative	0.1	0.1	0	0	0.12	0.02	0
Avg.	0.436	0.445	0.467	0.476	0.445	0.458	0.476
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.565	0.536	0.812	0.609	0.667	0.667	0.58
Neutral	0.917	0.926	0.907	0.926	0.917	0.926	0.917
Negative	0.58	0.26	0.66	0.64	0.7	0.76	0.38
Avg.	0.736	0.661	0.824	0.767	0.793	0.811	0.696
SYMMETRIC 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.565	0.551	0.797	0.58	0.681	0.623	0.551
Neutral	0.907	0.926	0.907	0.907	0.917	0.917	0.898
Negative	0.38	0.18	0.6	0.64	0.52	0.54	0.28
Avg.	0.687	0.648	0.806	0.749	0.758	0.744	0.656

6.5 Taula: POLITIKA datu-basean lortutako emaitzak.

Emaitzetan ikusten denez, jatorrizko hitz-zakuen asmatze-tasak nabarmen igo dira, orain arteko emaitzarik onenak lortuz. Atributu-aukeraketako azpimultzoen asmatze-tasak, berriz, kasu batzuetan apur bat igo dira, eta beste batzuetan apur bat jaitsi. Oraingo proba honekin SMOTE teknika jatorrizko datu-multzoarekin oso eraginkorra dela ondoriozta daiteke. Baina azpimultzoei buruz ezin da gauza bera esan, tasa batzuk apur bat igo diren arren, guztiak datu-multzo normalaren azpitik geratu baitita.

Aipagarria da HOTELA datu-basean negatiboaren asmatze-tasa nahikotxo igo dela, beste klaseen tasak apur bat jaitsiz. Hori SMOTEri esker izan da, izan ere klase negatiboa zen instantzia gutxien zeukana, eta hortaz instantzia-kopuruaren bikoizketa jasan duena. POLITIKA datu-basean gauza bera gertatu da, hor ere negatiboa baitzen klase minoritarioa.

Gogoratu beharra dago azpimultzo hauetan SMOTE atributu-aukeraketen ondoren aplikatu dela, eta ez lehenago. SMOTE atributu-aukeraketaren aurretik egin balitz, ziurrenik beste emaitza batzuk lortuko lirateke. Kontuan izanda SMOTEk datu-multzo normalaren gainean eragin handia izan duela, pauso berri bat ematea erabaki da: atributu-aukeraketak SMOTE filtroa pasa ondoren egitea.

Azken batean, aukeraketa egin aurretik klaseen instantzia-kopurua orekatzeak logika handia dauka, eta hori egiteak emaitzei positiboki eragingo diela aurreikusi da.

Hortaz, SMOTE aplikatuta zeukan hitz-zaku normalari atributu-aukeraketa guztien filtroak pasa zaizkio, bakoitza fitxategi separatu gisa gordez. Ondoren, fitxategi bakoitza banan-banan kargatu eta bakoitzari sailkatzaile guztiak berriro aplikatu zaizkio. Hori guztia egin ostean, lortu diren emaitza berriak 6.6 Taulan eta 6.7 Taulan adierazi dira.

HOTELA datu-basea (SMOTE x2 + aukeraketak)							
NORMAL 1651	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.398	0.405	0.822	0.802	0.988	0.767	0.974
Neutral	0.736	0.604	0.346	0.583	0.545	0.716	0.333
Negative	0.729	0.544	0.695	0.588	0.443	0.871	0.229
Avg.	0.56	0.49	0.655	0.697	0.756	0.77	0.647
BESTFIRST 76	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.495	0.504	0.729	0.848	0.949	0.807	0.978
Neutral	0.675	0.666	0.259	0.505	0.534	0.486	0.351
Negative	0.672	0.58	0.852	0.455	0.459	0.512	0.218
Avg.	0.581	0.567	0.608	0.674	0.736	0.657	0.652
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.396	0.385	0.676	0.656	0.942	0.777	0.796
Neutral	0.667	0.627	0.183	0.43	0.385	0.529	0.438
Negative	0.783	0.715	0.649	0.382	0.265	0.574	0.113
Avg.	0.547	0.517	0.521	0.539	0.625	0.666	0.566
RELIEF 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.554	0.577	0.696	0.835	0.962	0.82	0.983
Neutral	0.647	0.641	0.224	0.491	0.529	0.435	0.355
Negative	0.653	0.573	0.831	0.457	0.412	0.501	0.176
Avg.	0.6	0.596	0.576	0.664	0.733	0.646	0.648
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.617	0.656	0.713	0.849	0.953	0.83	0.981
Neutral	0.622	0.598	0.122	0.474	0.522	0.393	0.353
Negative	0.63	0.54	0.834	0.412	0.415	0.368	0.182
Avg.	0.621	0.618	0.554	0.657	0.726	0.615	0.648
SYMMETRIC 75	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.421	0.422	0.68	0.813	0.953	0.781	0.984
Neutral	0.672	0.666	0.245	0.483	0.514	0.436	0.337
Negative	0.697	0.63	0.834	0.471	0.367	0.492	0.164
Avg.	0.547	0.533	0.575	0.652	0.715	0.625	0.641
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.529	0.551	0.65	0.826	0.948	0.774	0.98
Neutral	0.632	0.61	0.186	0.444	0.51	0.446	0.342
Negative	0.625	0.533	0.848	0.392	0.378	0.402	0.166
Avg.	0.577	0.566	0.544	0.633	0.713	0.608	0.641

6.6 Taula: HOTELA datu-basean lortutako emaitza berriak.

POLITIKA datu-basea (SMOTE + aukeraketak)							
NORMAL 1266	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.652	0.71	0.71	0.667	0.812	0.783	0.58
Neutral	0.667	0.546	0.657	0.88	0.917	0.926	0.907
Negative	0.9	0.86	0.9	0.68	0.68	0.98	0.34
Avg.	0.714	0.665	0.727	0.771	0.833	0.894	0.683
BESTFIRST 23	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.594	0.594	0.841	0.638	0.754	0.754	0.565
Neutral	0.935	0.935	0.889	0.898	0.917	0.917	0.907
Negative	0.32	0.2	0.74	0.66	0.7	0.62	0.42
Avg.	0.696	0.67	0.841	0.767	0.819	0.802	0.696
LATENT 150	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.768	0.768	0.507	0.638	0.464	0.739	0.725
Neutral	0.593	0.417	0.741	0.759	0.917	0.824	0.806
Negative	0.86	0.86	0.8	0.62	0.62	0.98	0.42
Avg.	0.705	0.621	0.683	0.692	0.714	0.833	0.696
RELIEF 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.696	0.652	0.841	0.638	0.87	0.783	0.58
Neutral	0.861	0.889	0.852	0.935	0.861	0.944	0.926
Negative	0.44	0.34	0.68	0.64	0.6	0.68	0.4
Avg.	0.718	0.696	0.811	0.78	0.806	0.837	0.705
RELIEF 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.725	0.71	0.783	0.58	0.812	0.754	0.594
Neutral	0.824	0.889	0.87	0.889	0.833	0.889	0.926
Negative	0.38	0.32	0.38	0.5	0.56	0.58	0.22
Avg.	0.696	0.709	0.736	0.709	0.767	0.78	0.67
SYMMETRIC 50	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.797	0.783	0.681	0.667	0.812	0.739	0.565
Neutral	0.88	0.88	0.917	0.88	0.861	0.88	0.907
Negative	0.5	0.2	0.86	0.66	0.74	0.72	0.5
Avg.	0.771	0.7	0.833	0.767	0.819	0.802	0.714
SYMMETRIC 30	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.725	0.71	0.754	0.696	0.812	0.754	0.565
Neutral	0.87	0.88	0.907	0.88	0.833	0.88	0.935
Negative	0.38	0.18	0.82	0.62	0.62	0.64	0.44
Avg.	0.718	0.674	0.841	0.767	0.78	0.789	0.714

6.7 Taula: POLITIKA datu-basean lortutako emaitza berriak.

Emaitza berriak ikusita, HOTELA datu-basean oso emaitza antzekoak lortu direla esan daiteke. Kasu batzueta hobera egin du eta beste batzuetan okerrera. POLITIKA datu-basean, ordea, emaitza guztiak hobetu dira. Horietako batzuk nabarmen gainera. Honekin, SMOTE aplikatzeko ordenak gehienetan garrantzia handia daukala ondoriozta daiteke.

HOTELA datu-basearen emaitzetan klase negatiboen asmatze-tasek nabarmen gora egin dutela ikus daiteke. Eta ez da harritzekoa, kontuan izanda prozesuaren puntu batean hanka sartu dela. Akats hori HOTELA datu-basean SMOTE nahi gabe bi aldiz aplikatzea izan da, eta akats hau aurrerantz eraman denez, aurreragoko emaitza batzuetan ere agertuko da, ”SMOTE x2” edo ”SMO2” gisa adierazita.

Akats hau ez da proiektuaren amaiera arte detektatu, beraz ezin izan zaio momentuan bertan konponbidea eman. Arazoa probak egiteko erabili den ordenagailuarekin egon da. WEKArako RAM memoria handitu nahian proba batzuk egin dira, SMOTE hainbat aldiz aplikatuz, eta antza denez, momenturen batean fitxategiren bat okerreko fitxategiarekin gainidatzi da. Horrek ustezko "hotela-SMOTE.arff" fitxategia izatez "hotela-SMOTEx2.arff" izatea eragin du. Akatsaz konturatzerako, kalkulu guztiak berregiteak ez zuen merezi. Gainera akats txiki bat da, eta ez dauka eragin handirik aurrerago egin diren kalkuluetan.

Puntu honetaraino emaitza asko lortu dira, eta hainbeste taularekin ez dago ongi desberdintzen gauza bakoitza. Hori dela eta, emaitza nagusi guztiak taula bakar batean adieraztea erabaki da. 6.8 Taulan, datu-multzo bakoitzarekin metodo bakoitzean eta datu-base bakoitzean lortu diren asmatze-tasa altuenak bildu dira.

	HOTELA			POLITIKA		
	auk	auk + SMO	SMO2 + auk	auk	auk + SMO	SMO + auk
NORMAL	0.692	0.738	0.77	0.817	0.894	0.894
BESTFIRST	0.717	0.706	0.736	0.847	0.837	0.841
LATENT	0.724	0.718	0.666	0.698	0.78	0.833
RELIEF	0.682	0.629	0.733	0.594	0.551	0.837
RELIEF	0.682	0.623	0.726	0.535	0.476	0.78
SYMMETRIC	0.711	0.737	0.715	0.871	0.824	0.833
SYMMETRIC	0.718	0.718	0.713	0.822	0.806	0.841

6.8 Taula: Orain arteko emaitza onenak.

Taula honetan, emaitzek izan duten eboluzioa garbiago ikus daiteke. Begibistan dagoen bezala, emaitzarik onena datu-multzo normalean SMOTE aplikatuta lortu da, bi datu-baseetan. Beraz, aukeraketa-metodoek ezin izan dituzte emaitzak hobetu. Hala ere, SMOTErekin emaitzak hobetu direnez, pauso handi hau arrakastatsua izan dela esan daiteke.

6.6 Multisailkatzaileak probatzen

Orain arte lortutako emaitzak nahikoa ez, eta are gehiago hobetzeko asmoarekin beste pauso bat ematea erabaki da. Hurrengo pausoa multisailkatzaileak probatzea izan da.

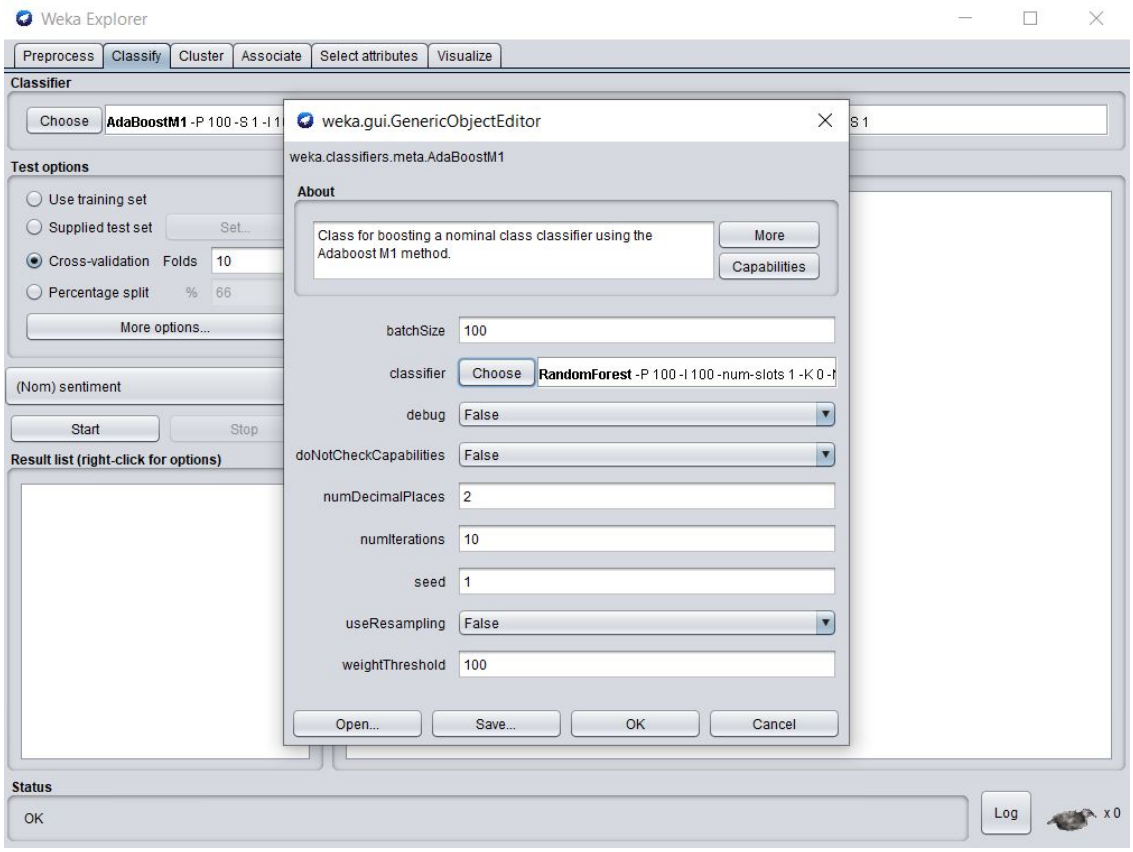
Multisailkatzaileak sailkatzaile-konbinaketak dira. Algoritmo metaheuristikoak erabiltzen dituzte, problemak ebatzi eta soluzioa aurkitzeko teknika desberdinen konbinazioak alegia. Hemen bi multisailkatzaile soilik erabili dira: *AdaBoostM1* eta *Bagging*. Bi teknika hauen azalpena jakiteko jo berriro kontzeptuetara, zehazki 5.5 kapitulura.

Teknika hauek aplikatzeko, WEKAko sailkatzaileen artean "meta" karpeta jo behar da, bertan baitaude

algoritmo mota hauek. Behin multisailkatzaile bat aukeratuta, algoritmo horrek bere baitan erabiliko duen sailkatzailea aukeratu beharra dago. Horretarako, izenaren gainean sakatu eta ”classifier” atalean sailkatzaile bat hautatu behar da. Pauso honetan, sailkatzaile desberdinekin hainbat proba egin dira.

HOTELA datu-baserako, multisailkatzaile bakoitzarekin *REPTree*, *RandomForest* eta *SMO* erabiltzea erabaki da. POLITIKA datu-baserako, berriz, *REPTree*, *RandomForest* eta *NaiveBayes*. Pauso honetan, sailkatzaile guztiekin probak egitearen ideia zeharo baztertu da, denbora galtzea izango litzatekeelako. Horregatik, aurreko pausoetan ongien aritu diren sailkatzaileak aukeratu dira lan honetarako. *REPTree* ordea, *Bagging* multisailkatzaileak defektuz eskaintzen duelako eta teorian konbinazio eraginkorra izaten delako gehitu da. *AdaBoostM1*ek defektuz *DecisionStump* sailkatzailea eskaintzen du, baina proba txiki bat egin ondoren oso emaitza txarrak ematen zituela ikusi eta baztertu egin da.

Hori erabakita, aldi bakoitzean sailkatzaile bat aukeratu eta ”Start” botoia sakatuz exekuzioak banan-banan egin dira. Hemen ere, orain arte bezala, kasu guztietan 10 iteraziodun balioztatze gurutzatua (*10 fold cross-validation*) erabili da. 6.8 Irudian *AdaBoost* multisailkatzailearen interfazea ikus daiteke. *Bagging* multisailkatzailearen interfazea ere oso antzekoa da.



6.8 Irudia: WEKAn *AdaBoost* multisailkatzailea eta bere barruan *RandomForest* sailkatzailea aukeratzen.

Bestalde, multisailkatzaileak ez dira datu-multzo guztietan aplikatu, ez litzatekeelako oso bideragarria izango. Datu-base bakoitzerako bost multzo soilik aukeratu dira: jatorrizko hitz-zakua, jatorrizko hitz-zakua SMOTE aplikatuta, aukeraketa-multzo onena, aukeraketa-multzo onena SMOTE ondoren aplikatuta eta aukeraketa-multzo onena SMOTE aurretik aplikatuta.

Multzo horien gainean exekuzio guztiak egin ondoren, lortu diren emaitzak 6.9 Taulan eta 6.10 Taulan adierazi dira.

HOTELA datu-basea						
NORMAL 1651	AdaBoostM1			Bagging		
(auk)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.905	0.992	0.798	0.945	0.995	0.857
Neutral	0.188	0.028	0.258	0.115	0.011	0.293
Negative	0.279	0.072	0.487	0.258	0.032	0.436
Avg.	0.688	0.691	0.667	0.698	0.685	0.695
NORMAL 1651	AdaBoostM1			Bagging		
(SMOTE)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.877	0.99	0.766	0.929	0.991	0.773
Neutral	0.557	0.543	0.727	0.478	0.529	0.735
Negative	0.479	0.436	0.843	0.372	0.387	0.797
Avg.	0.709	0.755	0.767	0.693	0.743	0.766
LATENT 150	AdaBoostM1			Bagging		
(auk)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.896	0.998	0.945	0.945	0.999	0.939
Neutral	0.18	0.009	0.172	0.115	0.006	0.196
Negative	0.133	0.008	0.38	0.09	0	0.364
Avg.	0.663	0.684	0.724	0.678	0.683	0.723
SYMMETRIC 75	AdaBoostM1			Bagging		
(auk + SMO)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.922	0.924	0.912	0.93	0.934	0.918
Neutral	0.123	0.155	0.12	0.089	0.137	0.115
Negative	0.629	0.661	0.668	0.621	0.674	0.669
Avg.	0.717	0.731	0.719	0.714	0.737	0.722
BESTFIRST 76	AdaBoostM1			Bagging		
(SMO2 + auk)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.866	0.951	0.807	0.922	0.963	0.803
Neutral	0.534	0.536	0.486	0.467	0.524	0.5
Negative	0.444	0.451	0.512	0.382	0.412	0.54
Avg.	0.69	0.736	0.657	0.688	0.731	0.664

6.9 Taula: Multisailkatzaileen emaitzak HOTELA datu-basean.

POLITIKA datu-basea						
NORMAL 1266	AdaBoostM1			Bagging		
(auk)	REPTree	RandomForest	NaiveBayes	REPTree	RandomForest	NaiveBayes
Positive	0.725	0.754	0.681	0.725	0.754	0.928
Neutral	0.907	0.926	0.843	0.889	0.917	0.852
Negative	0.08	0.04	0.52	0	0.04	0.12
Avg.	0.743	0.757	0.748	0.723	0.752	0.787
NORMAL 1266	AdaBoostM1			Bagging		
(SMOTE)	REPTree	RandomForest	NaiveBayes	REPTree	RandomForest	NaiveBayes
Positive	0.752	0.812	0.696	0.623	0.754	0.754
Neutral	0.853	0.926	0.769	0.889	0.917	0.75
Negative	0.776	0.62	0.9	0.48	0.62	0.92
Avg.	0.805	0.824	0.775	0.718	0.802	0.789
SYMMETRIC 50	AdaBoostM1			Bagging		
(auk)	REPTree	RandomForest	NaiveBayes	REPTree	RandomForest	NaiveBayes
Positive	0.754	0.797	0.841	0.71	0.754	0.928
Neutral	0.926	0.917	0.889	0.926	0.917	0.898
Negative	0.04	0.2	0.52	0.04	0.2	0.52
Avg.	0.757	0.787	0.827	0.743	0.772	0.861
BESTFIRST 24	AdaBoostM1			Bagging		
(auk + SMO)	REPTree	RandomForest	NaiveBayes	REPTree	RandomForest	NaiveBayes
Positive	0.667	0.739	0.884	0.623	0.739	0.884
Neutral	0.907	0.917	0.907	0.917	0.917	0.907
Negative	0.5	0.56	0.62	0.58	0.58	0.62
Avg.	0.744	0.784	0.837	0.753	0.789	0.837
SYMMETRIC 30	AdaBoostM1			Bagging		
(SMO + auk)	REPTree	RandomForest	NaiveBayes	REPTree	RandomForest	NaiveBayes
Positive	0.696	0.725	0.754	0.725	0.768	0.739
Neutral	0.87	0.843	0.907	0.898	0.88	0.907
Negative	0.54	0.64	0.82	0.56	0.62	0.84
Avg.	0.744	0.762	0.841	0.771	0.789	0.841

6.10 Taula: Multisailkatzaileen emaitzak POLITIKA datu-basean.

Oraingoan, sailkatzaileen artean emaitzak oso berdinduak atera dira. Dena dela, emaitzak garbiago ikusteko, multisailkatzaileak aplikatu aurretik eta ondoren lortutako emaitzak 6.11 Taulan bildu dira.

		HOTELA			POLITIKA	
		Before	After		Before	After
(auk)	NORMAL 1651	0.692	0.698	NORMAL 1266	0.817	0.787
(SMOTE)	NORMAL 1651	0.77	0.767	NORMAL 1266	0.894	0.824
(auk)	LATENT 150	0.724	0.724	SYMMETRIC 50	0.871	0.861
(auk + SMO)	SYMMETRIC 75	0.737	0.737	BESTFIRST 24	0.837	0.837
(SMO + auk)	BESTFIRST 76	0.736	0.736	SYMMETRIC 30	0.841	0.841

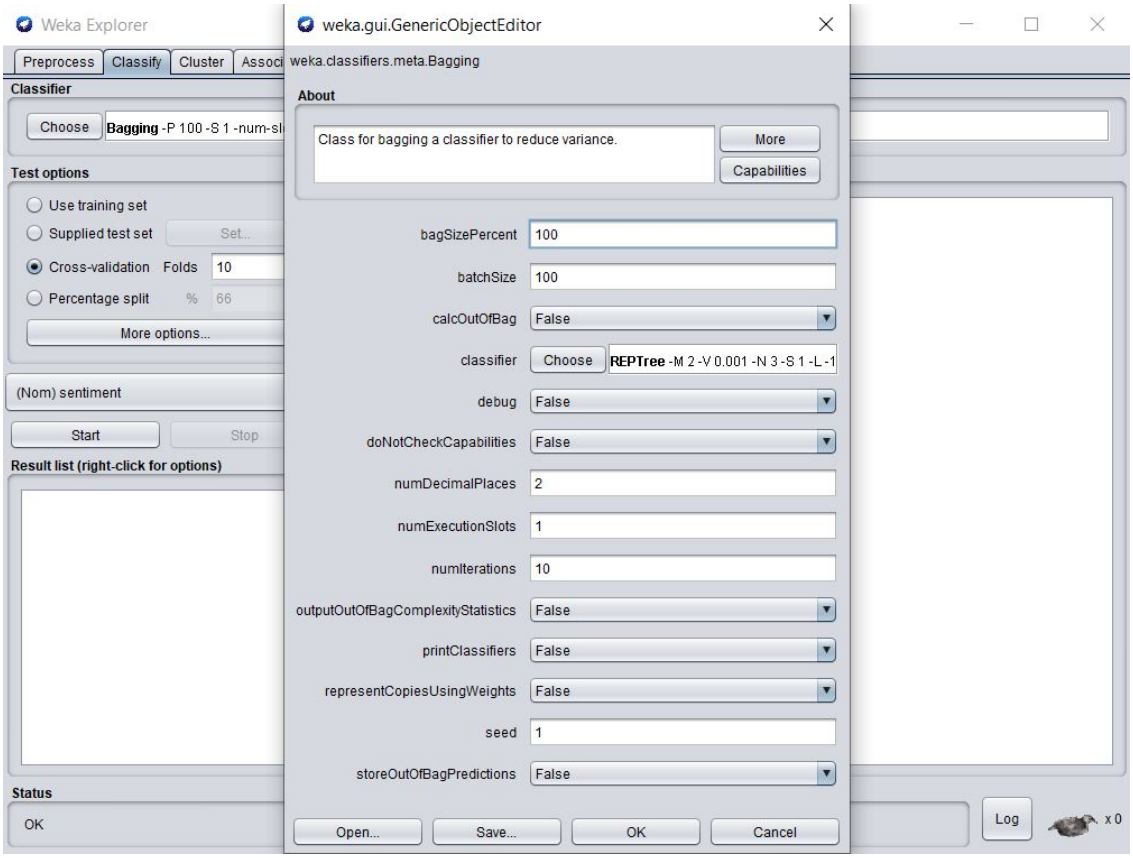
6.11 Taula: Emaitzen alderaketa HOTELA eta POLITIKA datu-baseetan.

Orain bai argi eta garbi esan daiteke, emaitzak oso antzeko mantendu diren arren, multisailkatzaile bakar batek ere ez duela aurretik zegoen emaitza gainditu. Emaitza berri guztiak berdinak edo okerragoak izan dira. Kontuan izanda multisailkatzaileek exekuzio-denbora askoz handiagoa behar izan dutela, eta batzuetan ordenagailuko RAM memoriarekin arazoak eman dituztela, pauso hau porrot hutsa izan da. Zaharrak berri.

6.7 Gauza bera beste modu batean

Aurreko pausoko porrotarekin pozik ez, eta multisailkatzaileekin beste proba bat egitea erabaki da.

Multisailkatzaileak, orokorrean, sailkatzaile ahulekin hobeto jarduten dute, sailkatzaile ahulak konbinatuz sailkatzaile indartsuago bat eraikitzen duelako. Hortaz, pauso berri bat eman eta *Bagging* barruan *k-NN* sailkatzailea aplikatuz proba batzuk egitea erabaki da, bide batez zakuaren tamaina (”bagSize”) eta iterazio-kopurua (”numIterations”) ere aldatuz. Aldaketa horiek guztiak 6.9 Irudian ikus daitekeen pantailan egin dira, *Bagging* multisailkatzailearen interfazean hain zuzen.



6.9 Irudia: WEKAn *Bagging* multisailkatzailearen aukerak aldatzen.

Defektuz, zakuaren tamaina %100ean dago. Horrek, entrenamendu-multzo gisa zaku osoa hartzen dela esan nahi du. Beraz, zakuaren tamaina txikitzen bada, algoritmoak ez du zaku osoarekin entrenatuko, eta emaitza desberdinak lortuko dira. Emaitza horiek, hobeak izateko probabilitate askotxo dago.

Bestalde, iterazio-kopurua defektuz 10ean dago. Baina kopurua igotzen bada saiakera gehiago egingo dira, eta orduan emaitza hobeak lortzeko aukera dago. Iterazio mota hau ez da balioztatze gurutzatuko iterazioekin nahasi behar. Bi kontzeptu desberdin dira. Iterazio hauek multisailkatzailearen barruan sailkatzaileekin burutuko direnak dira.

Pauso honetan probak HOTELA datu-basea bakarrik egin dira, exekuzioek luze jotzen dutelako eta pauso honetan denbora gehiegi galtzeak ez duelako merezi. Probak egiteko *3-NN* eta *7-NN* sailkatzaileak aplikatu dira, zakuaren tamaina %85ean jarrita, eta 15, 25 eta 50 iteraziorekin probatu da. Hori guztia egin ondoren, bildu diren emaitzak 6.12 Taulan adierazi dira.

HOTELA datu-basea						
NORMAL 1651	Bagging 3-NN + bagSize85			Bagging 7-NN + bagSize85		
(auk)	iterations15	iterations25	iterations50	iterations15	iterations25	iterations50
Positive	0.964	0.968	0.97	0.995	0.995	0.995
Neutral	0.047	0.05	0.051	0.016	0.012	0.017
Negative	0.066	0.064	0.072	0.011	0.011	0.011
Avg.	0.675	0.678	0.681	0.683	0.683	0.684
NORMAL 1651	Bagging 3-NN + bagSize85			Bagging 7-NN + bagSize85		
(SMOTE)	iterations15	iterations25	iterations50	iterations15	iterations25	iterations50
Positive	0.423	0.426	0.413	0.41	0.41	0.413
Neutral	0.755	0.759	0.641	0.639	0.639	0.641
Negative	0.642	0.668	0.488	0.48	0.48	0.488
Avg.	0.563	0.57	0.496	0.492	0.492	0.496
LATENT 150	Bagging 3-NN + bagSize85			Bagging 7-NN + bagSize85		
(auk)	iterations15	iterations25	iterations50	iterations15	iterations25	iterations50
Positive	0.861	0.863	0.866	0.973	0.934	0.937
Neutral	0.186	0.172	0.161	0.093	0.095	0.092
Negative	0.242	0.234	0.226	0.162	0.16	0.17
Avg.	0.653	0.651	0.65	0.673	0.675	0.678
SYMMETRIC 75	Bagging 3-NN + bagSize85			Bagging 7-NN + bagSize85		
(auk + SMO)	iterations15	iterations25	iterations50	iterations15	iterations25	iterations50
Positive	0.905	0.906	0.911	0.936	0.938	0.938
Neutral	0.104	0.099	0.095	0.065	0.062	0.054
Negative	0.625	0.63	0.64	0.57	0.566	0.573
Avg.	0.702	0.704	0.708	0.703	0.703	0.703
BESTFIRST 76	Bagging 3-NN + bagSize85			Bagging 7-NN + bagSize85		
(SMO2 + auk)	iterations15	iterations25	iterations50	iterations15	iterations25	iterations50
Positive	0.518	0.517	0.518	0.521	0.451	0.515
Neutral	0.726	0.722	0.726	0.684	0.688	0.684
Negative	0.609	0.616	0.621	0.529	0.523	0.533
Avg.	0.597	0.597	0.6	0.572	0.572	0.573

6.12 Taula: Multisailkatzaileekin emaitza berriak HOTELA datu-basean.

Zoritxarrez, bistan dagoen bezala, emaitzak oso txarrak izan dira. Aurreko pausoko emaitzak baino okerragoak izan dira, eta emaitza horiek ere ez ziren oso pozik egotekoak. Beraz, bigarrenez ondorioztatu da multisailkatzaileak proiektu honetan porrot hutsa izan direla.

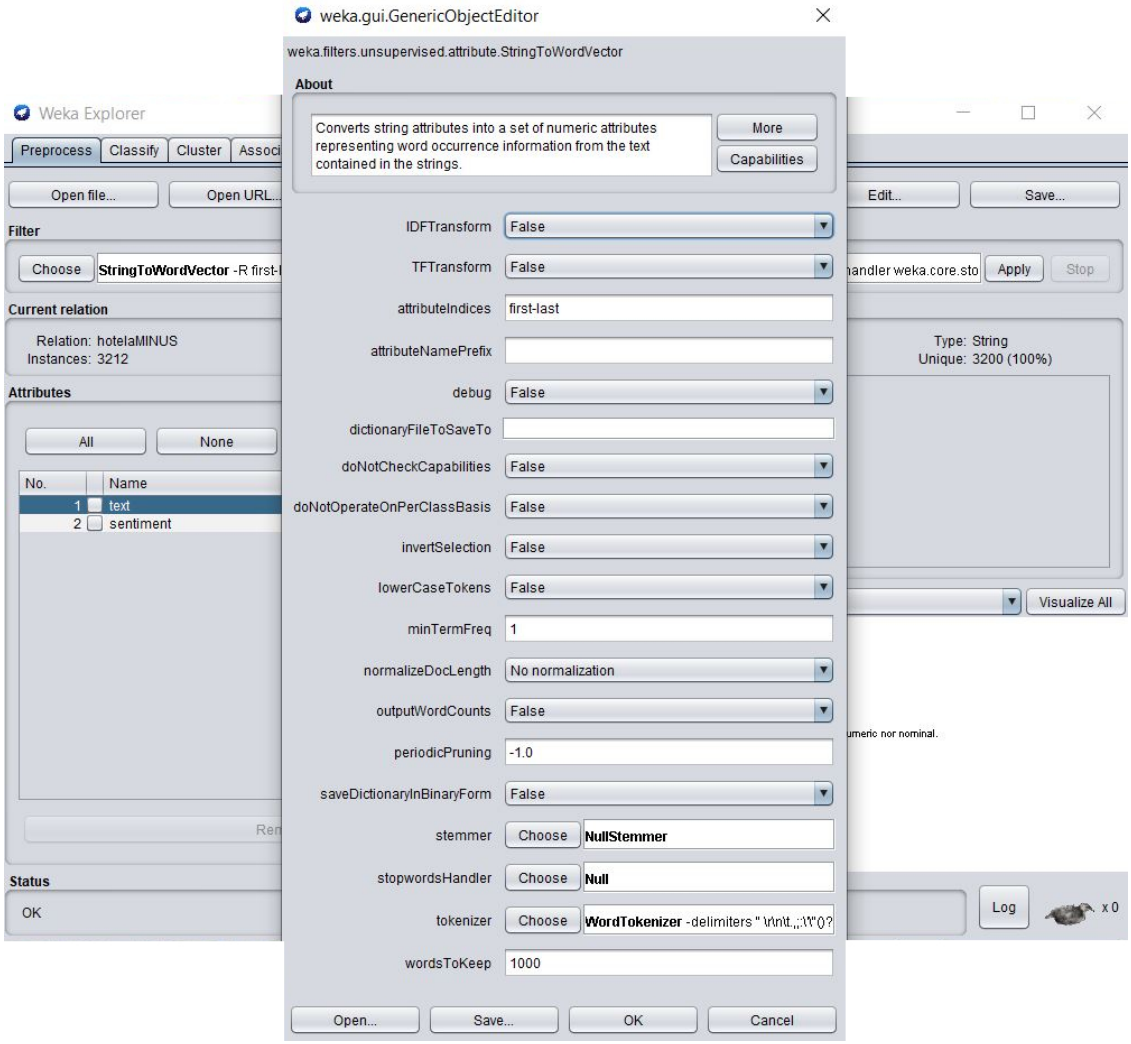
Honek, ordea, ez du multisailkatzaileak erabiltzea metodo txarra denik esan nahi. Proba gehiago egin litezke, konbinazio desberdin gehiago erabiliz, emaitza hobetzen den kasuren bat aurkitu arte. Baina orain arteko emaitzak ikusita, saiakera gehiago egiteak ez du itxaropen handirik ematen. Horregatik, multisailkatzaileekin denbora gehiago galdu beharrean, beste metodoren bat erabiltzea hobe izango dela erabaki da.

6.8 Azken irtenbidea

Proba eta kalkulu asko egin dira honaino iristeko. Emaitza onik eman ez duten proba eta kalkulu asko. Egoera ikusita, azken karta jokatzea erabaki da. Itxaropen guztiak amaierarako gorde den teknika berezi batean jarri dira, TF-IDF metodoan hain zuzen. Metodo honen azalpena jakiteko jo berriro kontzeptuetara, zehazki 5.8 kapitulura.

Metodo hau erabiltzeko hitz-zakua (*bag-of-words*) berriro sortu behar da, garapenaren lehen pausoan egin den antzera, ”StringToWordVector” filtroa erabiliz. Baina oraingoan, filtroaren aukeretan aldaketa batzuk behar dira. Aldaketak egiteko interfazea 6.10 Irudian ikus daiteke.

Dena dela, hori egin aurretik, HOTELA datu-basean aldaketa txiki bat egitea erabaki da: letra larri guztiak letra xehe bihurtzea. Horretarako, Notepad++ programarekin ”Ctrl+A” sakatuz fitxategiko testu guztia hautatu da, eta jarraian ”Ctrl+U” sakatuz hizki guztiak letra xehe bihurtu dira. Guztiak letra larri bihurtu nahi izango balira, ”Ctrl+Shift+U” sakatuz egin liteke. Garapenaren hasieran, WEKAk letra larriak eta xeheak kontuan hartzen ez zituelakoan, txikikeria honi ez zaio garrantzirik eman. Baina geroago, hitz-zakuan adibidez ”Lovely” eta ”lovely” hitzak bakoitza atributu desberdin bat bezala adierazita dagoela ikusi da. Konponketa hau oso aldaketa txikia den arren, hau egiteak emaitza hobetzen lagundu beharko luke.



6.10 Irudia: WEKAn *StringToWordVector* filtroaren aukerak aldatzen.

TF-IDF metodora itzuliz, hitz-zakua sortzerakoan teknika hau aplikatzeko, filtroaren aukeren artean ”IDFTransform” eta ”TFTransform” aktibatu behar dira, biak ”True” jarrita. Hori eginez, algoritmoak atributuen maiztasun-balioak kalkulatuko ditu.

Horretaz gain, teknika konfiguratzeko, defektuzko beste bi aukera ere aldatu dira. Horietako bat ”minTermFreq” aldagaiari 5 balioa ematea izan da, mantenduko diren hitzak klase berean 5 aldiz edo gehiagotan errepikatzen direnak bakarrik izan daitezen. Esate baterako, ”lovely” hitza zakuan sartu ahal izateko, esaldi positibo, edota neutro, edota negatibo guztien artean gutxienez 5 aldiz agertu behar da. Ez du balio, adibidez, esaldi positiboen artean 4 aldiz, neutroen artean 3 aldiz, eta negatiboen artean 2 aldiz agertzeak. Beste aldaketa ”wordsToKeep” aldagaiari 200 balioa ematea izan da. Horrela, klase bakoitzeko 200 hitz bakarrik hartuko dira.

Teknika honen probak HOTELA datu-basean bakarrik egin dira, bi datu-baseekin egiteko denbora gehiegi beharko litzatekeelako. HOTELA datu-baseko emaitzak hobetzea lortzen bada, orduan egingo dira probak POLITIKA datu-basearekin ere, bestela ez.

Filtroa aplikatu ondoren, nabarmentzekoa da atributu kopurua 1651etik 280ra murriztu dela. Badaezpada, berriro gogoratu beharra dago atributuak hitz desberdinen kopurua direla. Instantzia-kopuruak puntu honetan 3212 izaten jarraitzen du, hau da, datu-baseko iruzkin-kopurua.

Sailkatzaileekin probak egiten hasi aurretik, datu-baseari SMOTE filtroa ere pasatzea erabaki da, instantzia-kopurua orekatzeko. Hemendik abiatuta bi datu-multzo sortu dira: bata SMOTE behin aplikatuta eta bestea SMOTE hirutan aplikatuta. Oraingoan SMOTE behin baino gehiagota aplikatu ahal izatea lortu da, ordenagailuan WEKAri memoria gehiago emateko aldaketa batzuk egiteari esker.

SMOTE behin aplikatutako datu-multzoak 3588 instantzia (2192 positibo, 644 neutro, 752 negatibo) dauzka. SMOTE hirutan aplikatutakoak, berriz, 4984 instantzia (2192 positibo, 1288 neutro, 1504 negatibo). Bi datu-multzo horiei, banan-banan, betiko sailkatzaileak aplikatu zaizkie, eta lortu diren emaitzak 6.13 Taulan adierazi dira.

HOTELA datu-basea (TF-IDF + SMOTE)							
NORMAL 280	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.641	0.688	0.727	0.841	0.981	0.865	0.949
Neutral	0.062	0.22	0.37	0.238	0.042	0.208	0.064
Negative	0.89	0.802	0.743	0.721	0.658	0.773	0.503
Avg.	0.589	0.593	0.666	0.707	0.745	0.728	0.697
HOTELA datu-basea (TF-IDF + SMOTE x3)							
NORMAL 280	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.28	0.259	0.728	0.803	0.977	0.765	0.945
Neutral	0.631	0.491	0.444	0.491	0.428	0.463	0.196
Negative	0.969	0.939	0.763	0.783	0.813	0.88	0.678
Avg.	0.579	0.524	0.665	0.716	0.786	0.722	0.671

6.13 Taula: HOTELA datu-baseko emaitzak TF-IDF metodoarekin.

Ikus daitekeen bezala, oso emaitza onak lortu dira. Baina hobekuntza hobeto ikusteko, 6.14 Taulan, TF-IDF gabeko emaitzekin alderatu da.

	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
SMOTE	0.602	0.579	0.637	0.696	0.734	0.738	0.673
TF-DF + SMOTE	0.589	0.593	0.666	0.707	0.745	0.728	0.697

6.14 Taula: TF-IDF metodoarekin eta gabe lortutako emaitzen alderaketa.

Orain argi ikusten da emaitzak oso antzekoak direla, baina TF-IDF egiteak emaitza apur bat hobetu duela. Eta nola ez, SMOTE hirukoitzeko TF-IDFak orain arteko emaitzarik onena eman du, %78.6 alegia.

Halako emaitza onak ikusita, hemendik abiatuta pauso gehiago ematea pentsatu da, eta TF-IDFdun bi datu-multzoei multisailkatzaileak aplikatzea erabaki da. Multisailkatzaile horiek *AdaBoostM1* eta *Bagging* izan dira berriro, baina defektuzko aukerekin eta sailkatzaile indartsuekin, hasiera batean egin den bezala.

Multisailkatzaileekin probak egin ostean, lortu diren emaitzak 6.15 Taulan adierazi dira.

HOTELA datu-basea (TF-IDF + SMOTE)						
NORMAL 280	AdaBoostM1			Bagging		
(SMOTE)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.891	0.98	0.865	0.932	0.988	0.865
Neutral	0.255	0.037	0.208	0.116	0.019	0.238
Negative	0.891	0.628	0.773	0.689	0.616	0.758
Avg.	0.729	0.737	0.728	0.734	0.736	0.73
HOTELA datu-basea (TF-IDF + SMOTE x3)						
NORMAL 280	AdaBoostM1			Bagging		
(SMOTE x3)	REPTree	RandomForest	SMO	REPTree	RandomForest	SMO
Positive	0.866	0.977	0.765	0.912	0.977	0.765
Neutral	0.5	0.436	0.463	0.366	0.428	0.463
Negative	0.794	0.811	0.88	0.793	0.813	0.88
Avg.	0.75	0.787	0.722	0.735	0.786	0.722

6.15 Taula: HOTELA datu-baseko emaitzak TF-IDF metodoarekin eta multisailkatzaileekin.

Taulako datuak ikusita, emaitzak nahiko berdin mantendu dira. Lortutako hobekuntza bakarra SMOTE hirukoitzaren datu-multzoa %78.6tik %78.7ra igotzea izan da. Oso igoera txikia izan da, baina pozik egoteko moduko emaitzak dira.

Pauso honetan, TF-IDF metodoarekin emaitzak hobetzea lortu da. Baina oraindik ez da guztiz frogatu ea benetan metodo hori eraginkorra izan den. Izan ere, oraindik ez da TF-IDFrik gabe SMOTE hirukoitza egitea probatu. Aukera posible hori mahai gainean ikusita, hurrengo pauso gisa SMOTE hirukoitz arrunt bat probatzea erabaki da, emaitzak alderatzeko.

Hasiera-hasierako hitz-zakuari SMOTE filtroa hiru aldiz aplikatu, eta ondoren banan-banan betiko sailkatzaileekin probak egin ostean, lortu diren emaitzak 6.16 Taulan adierazi dira.

HOTELA datu-basea (SMOTE x3)							
NORMAL 1652	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.282	0.259	0.807	0.806	0.991	0.76	WEKA
Neutral	0.589	0.38	0.462	0.527	0.445	0.693	OUT
Negative	0.959	0.875	0.664	0.822	0.822	0.972	OF
Avg.	0.566	0.476	0.675	0.739	0.799	0.806	MEMORY

6.16 Taula: HOTELA datu-baseko emaitzak, SMOTE soilik eta hirutan aplikatuta.

Ordenagailuak denbora luzea behar izan du exekuzioak egiteko, eta *DecisionTable* azkenean ezin izan da egin, baina bistan den bezala, oso emaitza onak lortu dira.

TF-IDF teknika erabiltzearen eta ez erabiltzearen arteko desberdintasuna argiago ikusteko, 6.17 Taulan konparaketa bat egin da.

	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
SMOTE3	0.566	0.476	0.675	0.739	0.799	0.806	OUT OF MEM
TF-DF + SMO3	0.579	0.524	0.665	0.716	0.786	0.722	0.671

6.17 Taula: TF-IDF metodoarekin eta gabe lortutako emaitzen alderaketa, SMOTE hirukoitzaz.

Oraingoan zalantzarik gabe esan daiteke TF-IDF erabiltzeak ez duela merezi, TF-IDF gabe askoz emaitza hobea lortu baita.

Eta azken proba horrekin, garapen osoko emaitzarik onena lortu da, %80.6 alegia. Hau ikusita, SMOTE zenbat eta gehiagotan erabili orduan eta emaitza hobea lortuko dela pentsa daiteke. Baina ez, pentsamendu hori okerra da. Izan ere, SMOTEk egiten duena klase minoritarioko instantzia-kopurua bikoiztea da. SMOTE hiru aldiz aplikatuta datu-multzoa nahiko orekatuta geratu da, baina laugarrenez aplikatuko balitz desoreka sortuko litzateke, klase minoritarioa instantzia gehiegi izatera pasako litzatekeelako. Hortaz, %80.6ko asmatze-tasarekin, HOTELA datu-baseko probak bukatutzat ematea erabaki da.

Azkenik, proiektua guztiz bukatutzat jotzeko, HOTELA datu-basearekin egindako azken proba POLITIKA datu-basearekin ere egitea falta da. Beraz, hasierako hitz-zakuari SMOTE filtroa hiru aldiz aplikatu zaio, eta betiko sailkatzaileekin lortu diren emaitzak 6.18 Taulan adierazi dira.

POLITIKA datu-basea (SMOTE x3)							
NORMAL 1652	3-NN	7-NN	NaiveBayes	J48	RandomForest	SMO	DecisionTable
Positive	0.986	0.986	0.913	0.87	0.942	1	0.855
Neutral	0.519	0.333	0.778	0.759	0.88	0.889	0.639
Negative	1	0.96	0.98	0.86	0.9	1	0.68
Avg.	0.844	0.775	0.89	0.832	0.91	0.965	0.737

6.18 Taula: POLITIKA datu-baseko emaitzak, SMOTE soilik eta hirutan aplikatuta.

SMOTE bakarrarekin asmatze-tasarik altuena %89.4 zen, baina oraingoan %96.5era igotzea lortu da. Datubase honetan ere, SMOTE hiru aldiz baino gehiago aplikatzeak ez du merezi, instantzia-kopurua zeharo desorekatuko litzatekeelako. Izan ere, SMOTE hirukoitzarekin instantziak ”positibo=138, neutro=108, negatibo=100” dira, eta SMOTE berriro egingo balitz negatiboak 200era pasako lirateke.

Horrenbestez, hauek izan dira proiektua egitetik atera diren emaitza eta ondorio guztiak.

7. KAPITULUA

Azken ondorio eta hausnarketak

HOTELA datu-basean lortu den emaitzarik onena %80.6 izan da. POLITIKA datu-basean, berriz, %96.5. Bi emaitza horiek metodo berdinarekin lortu dira, baina, hala ere, bien artean desberdintasun handia dago. Horren zergatia hainbat arrazoi desberdin izan daitezke.

Horietako bat datu-baseek hizkuntza desberdina daukatela izan daiteke. HOTELAko iruzkinak ingelesez daude, eta POLITIKAkoak, berriz, gazteleraz. Hizkuntza bakoitzak bere berezitasunak dauzka, eta sentimendu-analisia egiterakoan denek ez dute eraginkortasun bera izango. Gizakiek hizkuntza batzuk beste batzuk baino errazago ikasten dituzte. Hizkuntza batzuk hilabete pare batean ikas daitezke, beste batzuetarako urteak behar dira. Makinekin gauza bera gertatzen da. Hizkuntza batzuk ikasteko saiakera eta datu gehiago behar dira.

Beste arrazoietako bat garapenean zehar aipatu dena da. HOTELA datu-basea, ingelesez dagoen arren, Indian hitz egiten den ingeles kaskarrean dagoela. Adibide gisa, azter dezagun datu-basetik ateratako esaldi hau: ”actually the purpose of hotel booking is for my cousin ,and not for me ,he went alsone and stayed alone he sais thjat the hotel is very very bad at that time he suffired witj heay rain also in chennai unfortunately he could not stayed for second day and he returns”.

Esaldi horrek hiztegiari hainbat ostiko ematen dizkio. Gramatikaren aldetik hain dago gaizki, ezen pare bat aldiz irakurri beharra dagoen zer esan nahi duen ulertzeko. Dena dela, hitz-zakua sortzerakoan n-gramarik erabili ez denez, hitzen arteko ordena eta harremana ez da kontuan hartu. Beraz, proiektu honetan gramatika gaizki egoteak ez du inolako eraginik izan.

Baina zoritxarrez, gramatika ez da gaizki dagoen gauza bakarra. Ortografia- zein tipografia-akatsak nonahi daude, eta emaitzen kalterako eragiten dute. Izan ere, ”alsone”, ”sais”, ”thjat”, ”suffired”, ”witj” eta ”heay” hitzek ”alone”, ”says”, ”that”, ”suffered”, ”with” eta ”heavy” izan beharko lukete. Gaizki idatzitako hitz bakoi-

tzagatik atributu berri bat sortuko da. Hitz-zakua ezertarako balio ez duten atributuz beteko da, eta ondorioz, benetako atributuek agerpen-kopuru murritzagoa izango dute.

Esan beharra dago ere, datu-baseko esaldi guztiak ez daudela horren gaizki. Hiru mila esalditik gora dagoenez, bakoitza banan-banan egiaztatzea ez da ezta burutik pasa. Baina hala ere, begiratu batean esaldi asko ongi idatzita daudela ikusten da.

Azkenik, hobetze aldera, hitz singularrak eta pluralak hitz bakarrean biltzeko metodoren bat ere bilatu zitekeen. Hau da, esate baterako, ”room” eta ”rooms” hitzak, bi atributu desberdin izan beharrean, atributu bakarra izatea. Horretarako, hitz bakoitzaren erroa bilatu eta erro berdina duten hitzak atributu bakar gisa hartu beharko lirateke kontuan. Hori egiteak hitz-zakuaren kalitatea apur bat handituko zukeen.

Bestalde, garapenaren amaieran, SMOTE gehiagotan aplikatzeak emaitza hobetu beharrean okertu egingo lukeela aipatu da. Ongi egongo litzateke hori hitzez baino emaitzekin frogatzea, baina horretarako RAM memoria handiagoa duen ordenagailu bat beharko litzateke.

Lana borobiltzeko, datuen entrenamenduarekin lortutako emaitzak oso onak direla esan beharra dago. %80.6 eta %96.5 oso asmatze-tasa onak dira, jendearen iritzi orokorra jakiteko behintzat. Datu hauekin, hotel bateko jabeak adibidez, bere mila bezeroen iritzia amen batean orokortu dezake, eta zerbitzuarekiko poztasun-maila zenbatekoa den amen batean jakin. Hau edozein arlotan aplikatu daiteke, baina arlo bakoitzerako entrenamendua datu desberdinekin egin beharko da, eta batzuekin emaitza hobeak lortuko dira besteekin baino.

Etorkizunera begira, sentimenduen analisia gero eta garrantzi gehiago hartzen ari da, eta ez litzateke erokeria bat izango laster negozioen munduan guztiz ezinbestekoa izatera iritsiko dela esatea. Gainera, denbora aurrera joan ahala, teknika hobeak garatuko dira. Sentimenduen analisiak etorkizun handia dauka, eta arlo hau lantzeak benetan merezi du.

Bibliografia

[1] MONKEYLEARN. *Everything There Is to Know about Sentiment Analysis (2020-01-02)*. https://monkeylearn.com/sentiment-analysis/.

[2] MEDIUM (SONI, YASH). *Machine Learning for dummies — explained in 3 mins! (2017-07-24)*. https://becominghuman.ai/machine-learning-for-dummies-explained-in-2-mins-e83fbc55ac6d.

[3] THE UNIVERSITY OF WAIKATO. *Weka 3 — Data Mining with Open Source Machine Learning Software in Java*. https://www.cs.waikato.ac.nz/ml/weka/.

[4] HO, DON. *Notepad++*. https://notepad-plus-plus.org/.

[5] GNU PROJECT. *GIMP - GNU Image Manipulation Program*. https://www.gimp.org/.

[6] OVERLEAF. *Overleaf, Online LaTeX Editor*. https://www.overleaf.com/.

[7] EUSKAL HERRIKO UNIBERTSITATEA. *Webaren bidezko Posta*. https://webposta.ehu.eus/.

[8] JITSI.ORG. *Jitsi Meet*. https://meet.jit.si/.

[9] KAGGLE. *Kaggle: Your Machine Learning and Data Science Community*. https://www.kaggle.com/.

[10] MACHINE LEARNING MASTERY (BROWNLEE, JASON). *A Gentle Introduction to the Bag-of-Words Model (2017-10-09)*. https://machinelearningmastery.com/gentle-introduction-bag-words-model/.

[11] WIKIPEDIA. *Hizkuntzaren prozesamendu*. https://eu.wikipedia.org/wiki/Hizkuntzaren_prozesamendu.

[12] WIKIPEDIA. *Cross-validation (statistics)*. https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[13] WIKIPEDIA. *K auzokide hurbilenak*. https://eu.wikipedia.org/wiki/K_auzokide_hurbilenak.

[14] WIKIPEDIA. *Naive Bayes sailkatzaile*. https://eu.wikipedia.org/wiki/Naive_Bayes_sailkatzaile.

[15] OCTAVIAN'S BLOG. *Decision Trees — C4.5 (2011-03-25)*. https://octaviansima.wordpress.com/2011/03/25/decision-trees-c4-5/.

[16] TOWARDS DATA SCIENCE (YIU, TONY). *Understanding Random Forest (2019-06-12)*. https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[17] MICROSOFT (PLATT, JOHN). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines (1998-04-XX)*. https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.

[18] WIKIPEDIA. *Decision table*. https://en.wikipedia.org/wiki/Decision_table.

[19] THE UNIVERSITY OF AUCKLAND (RIDDLE, PATRICIA). *Reduced Error Pruning (1998-05-15)*. https://www.cs.auckland.ac.nz/~pat/706_98/ln/node90.html.

[20] QUANTDARE (GARRIDO, ANA PORRAS). *What is the difference between Bagging and Boosting? (2016-04-20)*. https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/.

[21] THE UNIVERSITY OF WAIKATO. *weka.attributeSelection*. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/package-summary.html.

[22] WIKIPEDIA. *Oversampling and undersampling in data analysis*. https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis.

[23] MONKEYLEARN (STECANELLA, BRUNO). *What is TF-IDF? (2019-06-10)*. https://monkeylearn.com/blog/what-is-tf-idf/.