

# Dokumentazioa Proiektua

Ieltzu Irazu, Mikel de Velasco, Jorge Nieto

24 de Marzo de 2015



# 1 Azaleko Orria / Aurkibidea

1. AZALEKO ORRIA / AURKIBIDEA .....	2
2. ESPARRU TEORIKOA .....	3
3. ESPARRU ESPERIMENTALA .....	4
4. ONDORIOAK / ETORKIZUNERAKO LANA .....	6
5. BIBLIOGRAFIA .....	6
6. BALORAZIO SUBJETIBOA .....	7

## 2 Esparru Teorikoa

1. Dokumentazio eta sintesi lana landutako ikasketa teknikari buruz, alegia:

- (a) Support Vector Machines(SVM)
- (b) **Multilayer Perceptron (MP)**
- (c) Bayes Network (BayesNet)

Gure kasuan **Multilayer Perceptron** estimatzailea ikastea tokatu zaigu. **Multilayer Perceptron** ez da estimatzaile simple bat eta horregatik **OneR** estimatzailearekin konparatuko dugu (estimatzaile simple dena).

### 2.1 Multilayer Perceptron

‘Multilayer Perceptron’ ingurune aldaketen aurrean aldatzen duen sare neuronal artifiziala da, eta sarrerako datu batzuetatik irteera egoki batzuetara aldatzen du. ‘Multilayer Perceptron’ sarea hiru geruza mota desberdinetan banatzen da:

- (a) **Sarrera geruza:** Sarean sarrera patroiak sartzen duten neurona multzoa. Neurona hauek ez dute prozesamendurik.
- (b) **Geruza ezkutuan:** Erdibideko neuronak dira, non bakoitzaren sarrerak aurreko geruza batetik datoz eta irteerak hurrengo geruza batera doaz.
- (c) **Irteera geruza:** Zeinen irteera-balioak sare guztiaren irteerekin bat datozen neuronak.

‘Multilayer Perceptron’ atzerako hedapen ikasketa entrenatzeko teknika erabiltzen duen sarea da. Metodo honek ‘**Estandar Linear Perceptron**’ metodoaren aldaketa bat da eta linelaki banatzeko modukoak ez diren datuak bereiz ditzake.

#### ATZERAKO HEDAPENEAREKIN IKASTEN:

Ikasketa datuak prozesatu eta gero haien pisua aldatzen denean gertatzen da, espero zen errore kopurua egon den errore kopuruarekin konparatuz.

**Algoritmoa:** Errorea Kalkulatu  
(Nodo Kopurua:  $N$  | Data Puntua:  $j$  | TargetValue:  $d$  | Perceptroiak emandako balioa:  $y$ )

**Aurre-baldintzak:**  $n \in N$

**Eraitza:**  $e_n(j)$

**Begin :**  
$$e_n(j) = d_n(j) - y_n(j)$$

**End**

Errorea irteeratik sarrera joango balitz bezala ulertu daiteke, horregatik hartzen du backpropagation izena.

#### PERCEPTROIA ENTRENATU:

Perceptroiek hiperplano bat definitzen dute, eta neurona sarea perceptroiaaren hiperplano hori implementatzen dute. Datu batzuk sarreratzat erabiliz, pisu balioak kalkulatu egin daitezke konexiorik gabe eta konektatzean

pertzeptroiak irteera balioak kalkulatu ditu.

Pertzeptroiari ez zaio lagin osoa pasatzen lehenengo aldian, honen zati bat ematen zaio. Sareak bere aldagaiak kasu bakoitzaren ostean eguneratzea da helburua, sarea gutxika definituz eta hobetuz. Errorea ez da lagin osoaren gainean kalkulatzeko, baizik eta jasotako instantzia bakoitzeko errore hori txikitzen ahalegintzen da.

2. Diseinua laburbildu eta atazen banaketa taldekideen artean eman. Javan garatu den programaren diseinua eta inplementazioari buruzko xehetasun aipagarriak.

Gure Diseinu txostena beste pdf batean egin dugu. Bere izena Design.txt da eta pdf hau dagoen karpeta berean dago.

### 3 Esparru Esperimentala

1. Datuak: datuen deskribapen kualitatibo eta kuantitatiboa
2. ariketako emaitzak sartu.

	Pre-processed Data		
	Train	Dev	Test
Nominal attributes	2	2	2
Numeric attributes	1	1	1
Attributes Total	3	3	3
Instances V1	11326	45	0
Instances V2	11202	17363	0
Instances total	22528	17408	24220

2. Aurre-prozesamendua: erabilitako filtroak aukeratzeko motibazioa azaldu eta 3. ariketako emaitzak eman.

	Class	Hold Out (train vs. dev)		
		Precision	Recall	F-Measure
Baseline	V1	0.333333333333	0.377777777777	0.354166666666667
	V2	0.9983868179985	0.99804181305	0.9982142857142858
	W.Avg.	0.8803191444948786		
Model	V1	0.0	0.0	0.0
	V2	0.9974149816176471	1.0	0.9987058180667797
	W.Avg	0.9961241451685142		

3. Emaitza esperimentalak: ereduak eskaintzen duten kalitatea aztertu. ...

		Resubstitution error			10-Fold Cross Validation		
	Class	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Baseline	V1	0.61136 9719539 1598	0.92401 7236830 5338	0.735861610 113107	0.6254901 96078431 4	0.533022 60135432 24	0.575566212 4305588
	V2	0.96202 1978021 978	0.76618 2391037 9836	0.853005943 6811849	0.8244395 95318389 2	0.872956 41519341 85	0.848004624 9851217
	W.Avg	0.8196513709397332			0.7704330807854539		
Model	V1	0.0	0.0	0.0	0.0	0.0	0.0
	V2	0.71526 9431089 7436	1.0	0.834002423 3222872	0.7152694 31089743	1.0	0.834002423 3222872
	W.Avg	0.5965364388571999			0.5965364388571999		

#### 4. Exekutagarrien exekuzio adibide bat (ikus 5.1. atala).

Txostenean iradokitzen zen bezala, ataza hau hiru partetan banandu da eta parte bakotzarekin **.jar** exekutagarri bat atera egin dugu.

##### (a) Aurreprozesatzailea:

Atal honetako **.jar** exekutagarria **Preprocess.jar** fitxategia da, eta programa ondo funtziona dezan, lau parametro behar ditu gutxienez:

- Development **.arff** fitxategia. Kasu honetan, **dev.p3.obfuscated.arff** fitxategia.
- Aurreko fitxategia aurreprozesatu ondoren, **.arff** fitxategi berria sortu nahi den helbidea. Adibidez **devaurre.arff** fitxategia sortzen genuen, baina edozein izen har dezake.
- Entrenamendu **.arff** fitxategia. Kasu honetan, **train.p3.resampled.obfuscated.arff** fitxategia.
- Hirugarren parametroan adierazi den fitxategia aurreprozesatu ondoren, **.arff** fitxategi berria sortu nahi den helbidea. Adibidez **trainaurre.arff** fitxategia sortzen genuen, baina edozein izen har dezake.
- Azkenik, beharrezko den parametro bat sar daiteke. Parametro honek, **1-100** arteko zenbaki bat izan behar da eta fitxategien instantzien % portzentai jakin batean lan egin nahi den adierazten da. Adibidez fitxategi osoaren instantzien **%80**-rekin lan egin nahi bada parametro honetan, 80 zenbakia sartzen da.

Beraz, exekutatzeko, konsolatik honako komandoa egikaritu beharko zen:

```
> java -jar Preprocess.jar dev.p3.obfuscated.arff devResult.arff train.p3.resampled.obfuscated.arff trainResult.arff
```

##### (b) Modeloa egitea:

Atal honetako **.jar** exekutagarria **GetModel.jar** fitxategia da, eta programa ondo funtziona dezan, bi parametro behar ditu, parametro horiek, aurreko atalean sorturiko bi **.arff** fitxategi berriak izango dira:

- Aurreprozesatutako development **.arff** fitxategia. Kasu honetan, **devaurre.arff** fitxategia.
- Aurreprozesatutako entrenamendu **.arff** fitxategia. Kasu honetan, **trainaurre.arff** fitxategia.

Beraz, exekutatzeko, konsolatik honako komandoa egikaritu beharko zen:

```
> java -jar GetModel.jar devaurre.arff trainaurre.arff
```

Exekutatu ondoren, programa honek, 'modeloak' katalogoa sortzen du eta bere barnean, **BaselineModel.model** eta **MultilayerPerceptronModel.model** modelo fitxategi bitarrak sortzen ditu.

Horretaz aparte, 'ficheros' katalogo berria ere sortzen du eta bertan bi modeloen ebaluaketa idazten du, **EvaluationBaseline.txt** eta **EvaluationMultilayerPerceptron.txt** hurrenez hurren.

(c) Sailkatzailea:

Atal honetako .jar exekutagarria **Classify.jar** fitxategia da, eta programa ondo funtziona dezan, hiru parametro behar ditu, parametro horietako bat, aurreko atalean sorturiko modeloetako bat izan behar da:

- i. Aurreko exekutagarrian (**GetModel.jar**) sorturiko **.model** fitxategi bitarra izango da. Kasu honetan, **BaselineModel.model** edo **MultilayerPerceptronModel.model** fitxategi bitarrak.
- ii. Bigarren parametroa ebaluatu nahi dugun fitxategiaren path-a izango da. Gure kasuan **test.p3.obfuscated.noclass.arff** izango da
- iii. Hirugarren parametro bezala ebaluatu ondoren sortutako fitxategia kokatu nahi dugun lekuaren path-a izango da. Kasu honetan, **TestPredictionsBaseline.arff** edo **TestPredictionsMultilayerPerceptron.arff**

Beraz, exekutatzeko, konsolatik honako komandoa egikaritu beharko zen:

```
> java -jar Classify.jar MultilayerPerceptronModel.model test.p3.obfuscated.noclass.arff TestPredictionsMultilayerPerceptron.arff
```

Exekutatu ondoren, programa honek, hirugarren parametroan (**TestPredictionsMultilayerPerceptron.arff** edo **TestPredictionsBaseline.arff**) pasaturiko izenarekin emaitzak gordeko ditu.

## 4 Ondorioak eta etorkizunerako lana

Lan honetan Multilayer Perceptron-ren parametro asko ekortu ditugu eta beraz oso modelo sendoa lortu dugu. Nahiz eta parametro asko ekortu izan, konturatu gara ez dela asko hobetzen modeloa, eta modelo honetan ekorketa egiterakoan beste parametro bat kontuan izan behar zela konturatu gara.

Eskatzen zeneko ia-ia guztia egin dugu eta beraz ez ditugu asko egiteko, baina etorkizuneko lan bezala beste Baseline algoritmoak ekortzea eta konparaketa berriak egitea izango zen, eta ahal izanez gero Multilayer Perceptronen ekorketan kontuan hartzen ez dugun ‘errorRate’ parametroa kontuan hartzea izango zen. Hobetzeko beste gauza bat aurreprozesamendua izango zen. Klaseen artean diferentzia asko zegoenez, estimatzailea ezin zuen ondo ikasi, ematzia ez oso onak ateraz.

Nahiz eta azkenen bertsioan ‘Multilayer Perceptron’ ekorketan, klase minoritarioaren ‘fmeasure’ balioarekin konparatu izana, tarteko bertsio batean beste era batera egiteko aukera eman genuen eta ekorketaren emaitzak gorde genituen. Bi ekorketa egiterakoa (‘fmeasure’ edo ‘errorRate’) eta konparatzerakoan, askorik ez zela hobetzen konturatu ginen, batzutan batek egiten zuen hobeto beste batzutan besteak... Beraz ez genuen ondorio handirik atera hortik. Egia da aurreprozesamendua hobetu beharra dagoela, eta horren ondorioz emaitza askoz hobetuko dala, gure uztez hori izan da arazorik nagusia.

## 5 Bibliografia

Bibliografia edukia indartu edo sostengatzeko emanda dago. Ezinbestekoa da iturri bibliografikoak aipatzea erabili edo horietara jotzen garen puntuan bertan. Ez ahaztu irudien iturria aipatzen. Testuan zehar esplizituki aipatu ez den iturririk ez sartu Bibliografia atalean. Iturria aipatzean zehaztu ahalbait gehien (liburuko kapitulua edo atala eman), honela beste irakurle batek sakondu ahal izango du iturri horiek kontsultatu.

- **MULTILAYER PERCEPTRON:** Data Mining - Practical Machine Learning Tools and Techniques (3rd Ed) (Page 232)
- **WEKA WIKISPACES:** [weka.wikispaces.com](http://weka.wikispaces.com)
- **WEKAREN DOKUMENTAZIOA:** Weka programa instalatzerakoan dagoen dokumentazioa ( [interneteko kopia](#) )

- **MULTILAYER PERCEPTRON:** [en.wikipedia.org/wiki/Multilayer\\_perceptron](https://en.wikipedia.org/wiki/Multilayer_perceptron) (Ideia bat eukitzeko)
- **MULTILAYER:** [en.wikipedia.org/wiki/Perceptron](https://en.wikipedia.org/wiki/Perceptron) (Ideia bat eukitzeko)

## 6 Balorazio Subjektiboa

(borondatezkoa) eranskin batean atazari buruzko hausnarketa egin. Honako puntuak lagungarriak izan daitezke baina ez dira derrigorrezkoak, beste batzuk ere sar daitezke.

1. Atazarekin lortu nahi ziren helburuak lortu dituzue? (ikusi 1. atala)
2. Batazbestean zenbat denbora eman duzue atazeaz lanean? Desglosatu: ikasketarako denbora, bilaketa bibliografikoa, softwarearen diseinua eta inplementazioa, txostena.
3. Talde-lana erabilgarria izan da ataza ebazteko?
4. Zer sortarazi dizue interesik handiena? Ikasleengan interesa eta motibazioa pizteko iradokizunak eman.

- **Ieltzu:**

Nire ustez Atazarekin lortu behar genueneko helburuak lortu dira. Denbora asko eman dut bereziki txostena egiten, gauza asko azaldu behar direlako. Ikasketarako denbora nahikotxo ere eman dut. Multilayer Perceptron nola doan jakitea denbora asko eramanez dit. Diseinua egiteko ez da arazo larriarik egon. Diseinua egiterakoan beharbada modeloa egiteko atala zailena izan da, luzera handienekoa da ere. Talde-lana oso garrantzitsua izan da. Beti bezala, gure taldean oso ondo moldatzen gara hirurok batera eta pozik nago egindako lanarekin. Interes handiena beharbada lan handia izatearena izango da, ez gurelako hain handia den bestelako lanik egin arlo honetan.

- **Jorge:**

Nire ustez atazako helburu guztiak bete ditugu. Denbora asko kostatu zitzaidan arazo guztiak konpontzea, izan ere hasiera batean ez nuen oso ondo ulertzen nola zihuan metodoa (Multilayer Perceptron). Denbora asko eman nuen interneten irakurtzen eta bideoak ikusten ondo ulertu ahal izateko. Azkenean Multilayer Perceptron ekorketa oso ona egin dugu eta denbora asko behar du ekortzeko, probak egiterakoan arazo bat izan zen (Memoria, denbora...). Dokumentazioari dagokionez ere denbora asko behar izan genuen guztia prestatzeko. Diseinua egiterakoan ez genituen arazorik izan, beti bezala hasiera batean diseinua guztiok egin eta gero lana ondo banatu genuen. Gure artean oso ondo moldatzen gara eta ez dugu inolako arazorik izan taldean lana egiterakoan.

Nire ustez oso praktika interesgarria izan da, guk ezagutzen ez genuen metodo bati buruz informazioa bilatu, diseinatu, inplementatu eta probak egin ditugulako. Aurreko praktikatik ikasitakoa hobetzeko ere balio izan du.

- **Mikel:**

Nire partez esan behar dut txostenean aipatzen zen guztiarekin bete dugula, baina uzte dut praktika hau ez zegoela guztiz prestatuta 'Multilayer Perceptron' klasifikatzailearekin lan egiteko, ekorketa egiterakoan klase minoritarioaren 'fmeasure'-rekin konparatzen gurelako eta praktika burutzearekin ulertu dudaren partez, modelo hau 'errorRate'-arekin lan egiten du, eta ekorketa kontuan hartzea merezi zuela uzte dut. Ekorketan nahiko parametro kontuan izan ditugu, eta kontuan hartu ez ditugunak, metodoa beste era batera erabiltzeko zelako izan da.

Dokumentazioa egiterakoan,  $\text{\LaTeX}$  erabili dugunez, hasieran zaila egin zaigu baina denbora joan ahala gero

eta azkarrago egiten genuen, egin nahi genuen guztia lortu ez arren, eta nahiko denbora emanez lenguaia honen “Kaixo Mundua” ikasteko. Hala ere behar den guztia dokumentatuta geratu dela uzte dut.

Taldean lan egiterakoan ez dugu inolako arazorik izan, beste proiektu batzuetan taldekide berak izan gare-lako eta badakigulako nola egiten duten bestek lan eta nola antolatu behar garen egin beharreko ataza aurrera eramateko.

Praktika osoa kontuan hartuz, interesgarria izan dela dirudit, klasifikatzaile berri bat nola funtzionatzen den ikusi dugulako. Beste taldekideen klasifikatzaileak nola funtzionatzen duten ikasteko gurarekin geratu naiz. Eta azkenik, etorkizun batean informazioa bilatu eta ulertzeko lagungarria izan dela.