2022

Practica 2 - Classificació

GRUPO GPA202-0930

DANIEL ALCOVER NETO – 1562889
ALEXANDRE GALVANY PARDO – 1566586
JORDI GONZÁLEZ GONZÁLEZ - 1526878

ÍNDEX

Introducció a la Pràctica 2
Llibreries Utilitzades 2
Apartat B – Classificació Numèrica 3
1. EDA (Exploratory Data Analysis) 3
2. Preprocessing 6
3. Model Selection 8
4. Model Selection 9
5. Metric Analysis 10
6. Hyperparameter Search 11
Apartat A – Comparativa de Models 12

Introducció a la Pràctica

En aquesta pràctica se'ns proposen una sèrie d'objectius a complir:

- Aplicar diferents models classificadors:
  - Regressor logístic.
  - SVM.
- Entendre les millores d'aplicar kernels.
- Avaluar correctament l'error del model.
- Visualitzar les dades juntament amb el model.
- Ser capaç d'aplicar tècniques de classificació en casos reals i validar els resultats.

Aleshores per tal de dur a terme tots aquests objectius hem utilitzat Kaggle.

Introducció al Dataset

Kaggle ens ha proporcionat una base de dades relacionada amb la Qualitat de l'Aigua la qual ens proporciona dades que aplicant els models de classificació ens poden ajudar a millorar la vida de les persones que tenen dificultats per obtenir aigua de qualitat així com la pròpia economia dels països amb més dificultats ja que una bona qualitat de l'aigua millora la salut de les persones i per tant redueix la despesa sanitària.

Dins de la base de dades trobem dades molt interessant com poden ser:

- Potability
- pH Value
- Turbidity

Podeu trobar tots els arxius en el nostre repositori: https://github.com/Jordigg2000/APC_LAB

Llibreries Utilitzades

- **Numpy**: És una llibreria que està especialitzada en el càlcul numèric i anàlisis de dades per a un volum normalment elevat de dades com és en el nostre cas.
- **Scikit-learn**: És una llibreria molt útil dins del apartat de ML ja que ens proporciona algorismes d'aprenentatge tant supervisats com no supervisats. Aquests algorismes ens permetran fer un estudi més profund en les nostres dades.
- **Matplotlib**: Llibreria que ens servirà per poder tenir una bona visualització de les dades ja que ofereix gràfics que faran que aquestes siguin comprensibles. Alguns exemples serien diagrames de dispersió, histogrames, gràfics de barres...
- **Scipy**: És una llibreria ideal per l'anàlisi de dades que volem fer ja que ens permetrà manipular les dades i visualitzar-la a través de una ample llista de comandes d'alt nivell de Python.
- **Pandas**: Aquesta llibreria ens serà molt útil ja que ens permetrà netejar i analitzar les dades. Esta especialitzada en dades que estan desordenades del món real com és en el nostre cas.
- **Missingno**: Aquesta llibreria ens ofereix un conjunt d'eines de visualització i utilitats sobre les dades que falten d'una forma flexible i fàcil d'utilitzar per tal d'obtenir un resum visual ràpid de la integritat (o la manca d'aquesta) del conjunt de dades.

Apartat B – Classificació Numèrica

1. EDA (Exploratory Data Analysis)

a) Quants atributs té la vostre base de dades? Quin tipus d'atributs són?

La nostre de base de dades assignada Water Quality està formada per un total de 10 atributs:

- **pH Value** (***float64***): pH de l'aigua
- **Hardness** (***float64***): capacitat que té l'aigua de precipitar sabó causada pel calci i el magnesi.
- **Solids** (***float64***): Indica quant de mineralitzada està l'aigua.
- **Chloramines** (***float64***): Desinfectants usats en la majoria de sistemes d'aigua públics.
- **Sulfate** (***float64***): Substancies que es troben en minerals, roques i en el sòl.
- **Conductivity** (***float64***): Conductivitat elèctrica que té l'aigua.
- **Organic_carbon** (***float64***): Mesura de la quantitat total de carboni en compostos orgànics en aigua pura.
- **Trihalomethanes** (***float64***): Productes químics els quals poden ser trobats en aigües tractades amb clor.
- **Turbidity** (***float64***): Indica la qualitat de la descàrrega de residus respecte a la matèria col·loïdal.
- **Potability** (***int64***): Indica si l'aigua és potable o no.

En la següent taula podem veure estadístiques dels atributs numèrics de la nostre base de dades:

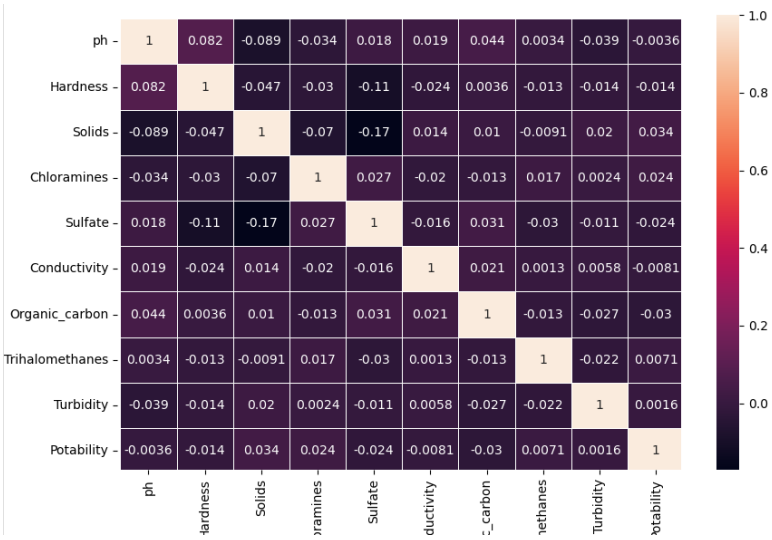
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
count	2785.000	3276.000	3276.000	3276.000	2495.000	3276.000	3276.000	3114.000	3276.000	3276.000
mean	7.081	196.369	22014.093	7.122	333.776	426.205	14.285	66.396	3.967	0.390
std	1.594	32.880	8768.571	1.583	41.417	80.824	3.308	16.175	0.780	0.488
min	0.000	47.432	320.943	0.352	129.000	181.484	2.200	0.738	1.450	0.000
25%	6.093	176.851	15666.690	6.127	307.699	365.734	12.066	55.845	3.440	0.000
50%	7.037	196.968	20927.834	7.130	333.074	421.885	14.218	66.622	3.955	0.000
75%	8.062	216.667	27332.762	8.115	359.950	481.792	16.558	77.337	4.500	1.000
max	14.000	323.124	61227.196	13.127	481.031	753.343	28.300	124.000	6.739	1.000

b) Com és el target, quantes categories diferents existeixen?

En la nostre base de dades el target només té una categoria, la de poder predir a partir dels atributs que tenim si l'aigua és potable o no ho és.

c) Pot haver una correlació entre X i Y?

La correlació entre les dades és la següent:



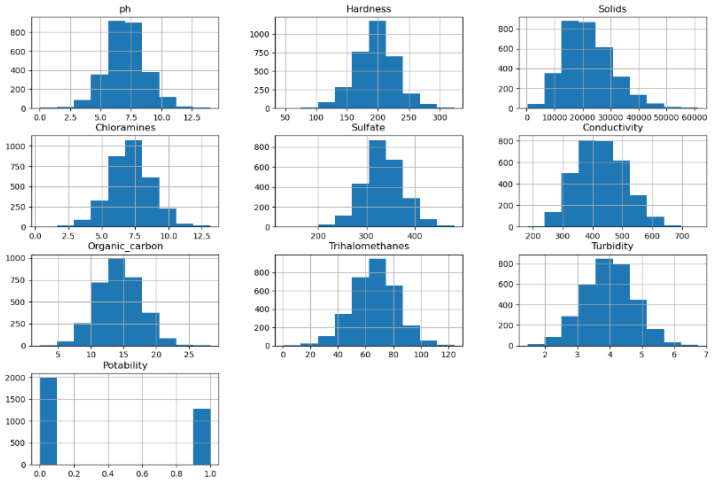
Com veiem la correlació que existeix entre les diferents dades és molt baixa ja que cap té una correlació superior al 0.1, inclús algunes correlacions són negatives referents a la potabilitat que és l'atribut objectiu i els diferents atributs independents.

d) Estan balancejades les etiquetes (distribució similar entre categories)? Creus que pot afectar a la classificació la seva distribució?

Les dades del nostre dataset no estan balancejades ja que els rangs de valors dels diferents atributs és molt ampli. Tot i això no creiem que aquest fet faci que la correlació en un futur millor i que afecti a la classificació.

Distribució de les dades

Veiem quina ha estat les distribució de les dades:

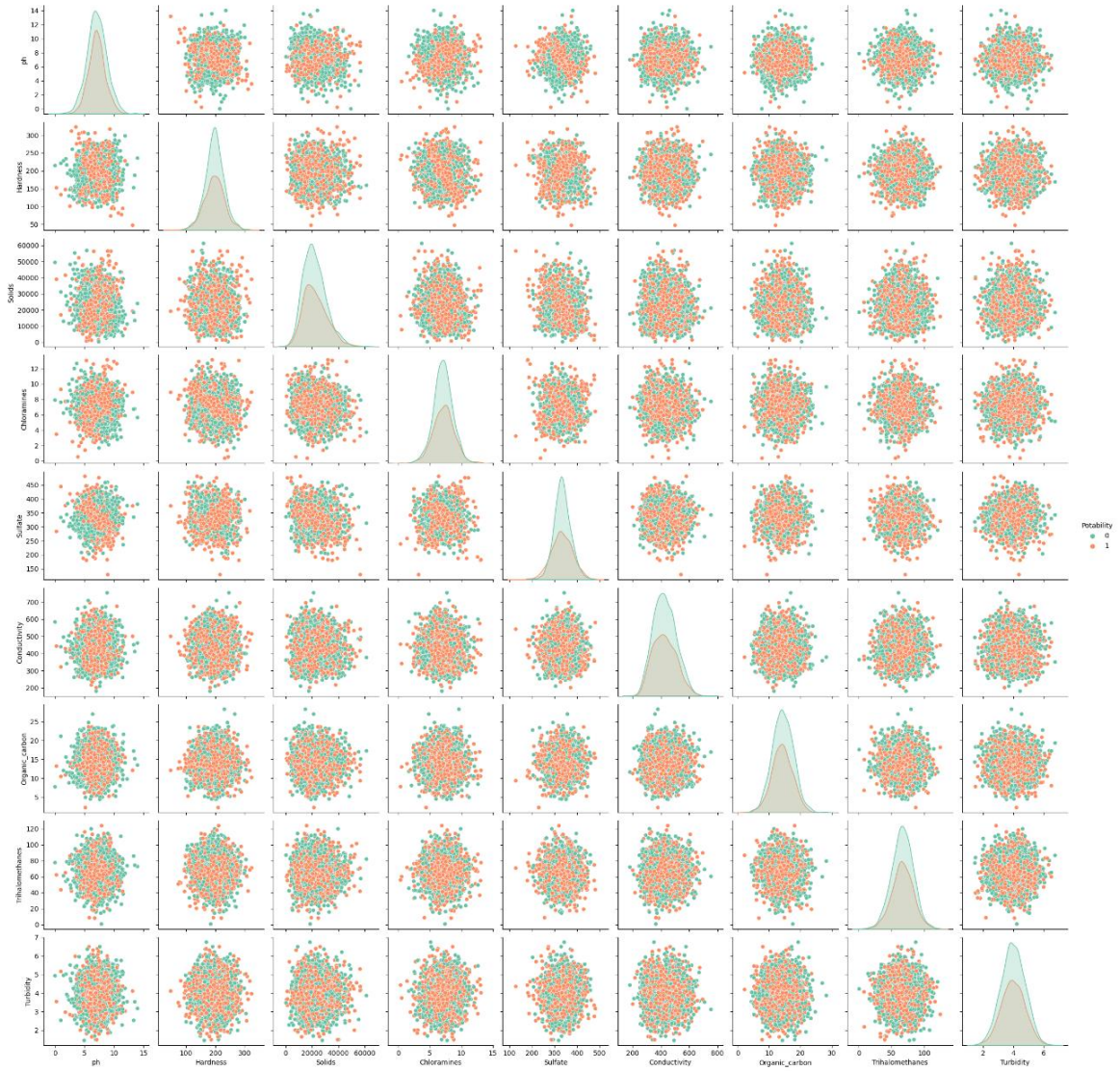


Com veiem les distribucions que segueixen la majoria d'atributs amb excepció de la Potability, que és un atribut binari, són distribucions Gaussianes o normals.

El fet de que siguin distribucions normals ens permet fer una predicció amb més probabilitat que si aquestes distribucions no fossin Gaussianes.

Relació entre els atributs i la seva Potability

Per tal de veure la relació que existeix entre els diferents atributs de les nostres dades i quan aquestes maquen si l'aigua es potable o no tenim les següents gràfiques:



- **Dades de color taronja**: Representen que per aquella combinació de valors de certs atributs l'aigua no és potable.
- **Dades de color verd**: Representen que per aquella combinació de valors de certs atributs l'aigua és potable.

2. Preprocessing

a) Estàn les dades normalitzades? Caldria fer-ho?

Les dades no estàn normalitzades ja que hi ha valors molt diferents entre els diferents atributs i és per això que seria adient fer una normalització de les dades per tal de que totes estiguin a una escala que sigui millor treballar amb elles.

Com veurem a la següent taula hem hagut de normalitzar tots els atributs de la nostre base de dades ja que els rangs dels atributs no tenien una similitud amb la que poder treballar de la millor forma. L'únic atribut que no ha estat normalitzat ha estat la Potability.

Dades a normalitzar:

```
columnes = ['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate', 'Conductivity','Organic_carbon','Trihalomethanes','Turbidity']
```

I aquests són els resultats un cop hem normalitzat les dades:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	0.571	0.336	0.544	0.680	0.669	0.313	0.700	0.286	0
1	NaN	0.297	0.301	0.492	NaN	0.719	0.497	0.451	0.577	0
2	NaN	0.641	0.322	0.699	NaN	0.415	0.562	0.533	0.304	0
3	NaN	0.606	0.356	0.603	0.647	0.318	0.622	0.808	0.601	0
4	NaN	0.485	0.290	0.485	0.515	0.379	0.359	0.254	0.496	0
...	...	...	...	...	...	...	...	...	...	...

b) En cas que les normalitzeu, quin tipus de normalització serà més adient per les vostres dades?

En el nostre cas hem hagut de normalitzar les dades les quals tenien valors molt diferents entre atributs. Per tal de fer això hem utilitzat el mètode de normalització de **minmax**.

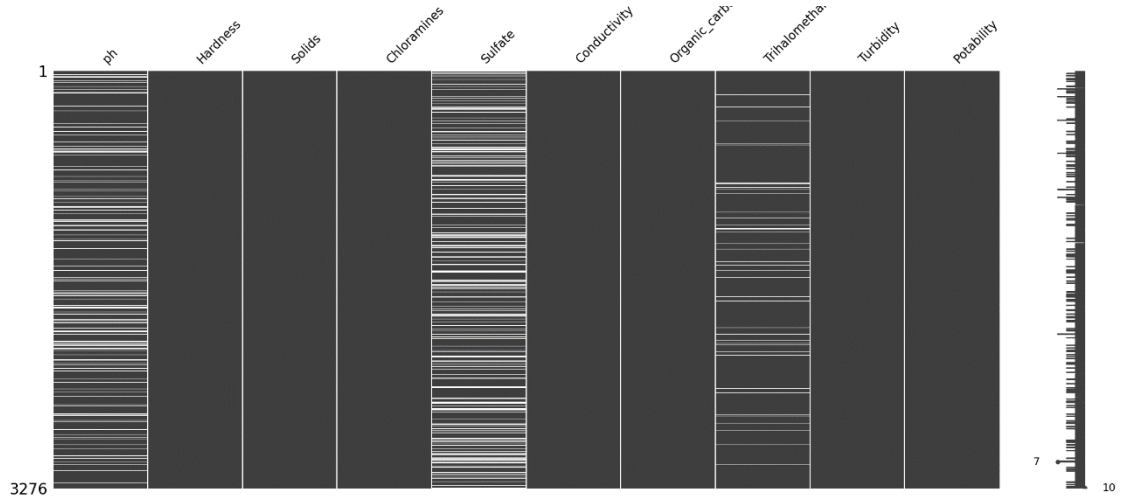
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

c) Teniu gaires dades sense informació? Els NaNs a pandas?

Per tal de veure si tenim gaires dades sense informació o NaN el que farem serà utilitzar la llibreria de missingno la qual ens permet visualitzar en una matriu aquells atributs els quals tenen dades NaN.

(Aquesta llibreria és externa, l'haurem d'instalar, per exemple amb pip install missingno)

Els espais en blanc que veiem en les gràfiques de cadascun dels atributs ens indica que falta informació sobre aquell atribut en concret per aquell recull de dades.



Aleshores tal i com podem veure en aquestes gràfiques tenim que els atributs que contenen dades sense informació són:

- pH.
- Sulfate.
- Trihalomethanes.

d) Teniu dades categoriques? Quina seria la codificació amb més sentit? (OrdinalEncoder, OneHotEncoder, d'altres?)

No, en el nostre cas tenim dades continues ja que aquestes dades no es poden definir en unes categories especifiques com podrien ser exemples d'edat o sexe.

En el nostre cas tenim dades numèriques les quals representen valors d'elements químics o de propietats de l'aigua.

3. Model Selection

a) Quins models heu considerat?

El models que hem seleccionat per tal de fer la anàlisis del nostre dataset són:

- KNN.
- SVM.
- Naive Bayes.
- Random Forest.
- Decision Tree.
- Logistic Regression.

b) Quin creieu que serà el més precís?

Model	Accuracy (%)
KNN	**64,02%**
SVM	**68,14%**
Random Forest	**69,51%**
Decision Tree	**64,02%**
Logistic Regression	**62,80%**

El més precís de tots els models que hem arribat a aplicar és el de Random Forest ja que tenim un accuracy del 69,51% tot i que molt a prop s'ha trobat el model de SVM (68,14).

c) Quin serà el més ràpid?

El més ràpid de tots els models seria el KNN, encara que es ràpid d'entrenat però lent fent les prediccions.

Respecte als altre models alguns són molt ràpids al entrenar, però tarden més l'hora de fer prediccions i viceversa.

d) Seria una bona idea fer un `ensemble`? Quins inconvenients creieu que pot haver-hi?

Fer un 'ensemble' sí que es una bona idea, ja que tots els models utilitzats son molt bons i combinant-los podrien ser millors. L'inconvenient que té és que triga molt més en entrenar i predir.

4. CrossValidation

a) Per què és important cross-validar els resultats?

Per a que el model no es sobre ajusti a les dades d'entrenament i per tant es produeixi overfitting i així es pugui generalitzar millor.

D'aquesta forma també podem obtenir una estimació més precisa del model.

b) Separa la base de dades en el conjunt de train-test. Com de fiables seran els resultats obtinguts? En quins casos serà més fiable, si tenim moltes dades d'entrenament o poques?

Si tenim moltes dades d'entrenament, els resultats seran més fiables en la majoria dels casos, tot i que un sobre entrenament del model podria comprometre el rendiment ja que podria patir d'alta variància, és a dir, centrar-se massa en detalls que són molt específics i no en generalitzar i treure patrons.

En aquests casos podríem tenir que estem tenint un overfitting del nostre model i no seria capaç de generalitzar amb les dades de test que li entréssim.

De la mateixa forma el fet de que aquestes dades siguin de qualitat és en igual mesura igual de important que la quantitat de dades que li passem al model per entrenar-lo.

D'altre banda en el cas de tenir poques dades d'entrenament podria comportar a una situació de underfitting del model ja que aquest no podria treure tendències de les dades que ha rebut i per tant no podrà fer models precisos.

c) Quin tipus de K-fold heu escollit? Quants conjunts heu seleccionat (quina k)? Com afecta els diferents valors de k?

Hem utilitzat en el nostre cas 5 conjunts.

La selecció de k en el nostre model ens afecta en el sentit de que si escollim menys de 5 conjunts el model és sobre ajusta massa i pot acabar provocant overfitting i en el cas de que en seleccionem masses el model es sobre ajusta menys, però observem que el temps tant de l'entrenament com de la predicció augmenta.

d) Es viable o convenient aplicar `LeaveOneOut`?

En el nostre cas no hem considerat que sigui una bona opció ja que el nostre dataset és massa gran i per tant faria que el model trigués massa en entrenar i en predir.

5. Metric Analysis

a) Podrieu explicar i justificar quina de les següents mètriques serà la més adient pel vostre problema? accuracy_score, f1_score o average_precision_score.

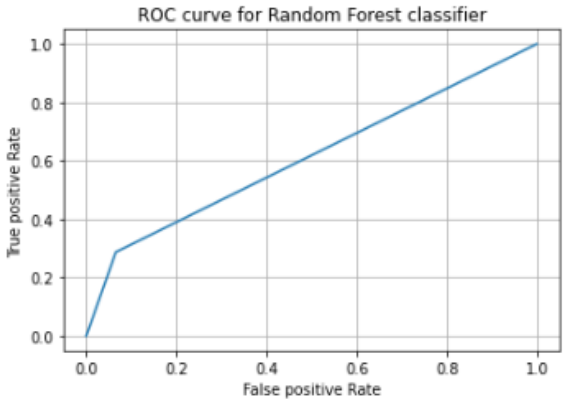
L'anàlisi de mètriques són un seguit de paràmetres que ajuden a calcular el rendiment d'una classificació duta a terme per qualsevol algorisme de classificació, com el "random forest" o la regressió logística.

Les mètriques que hem calculat han estat:

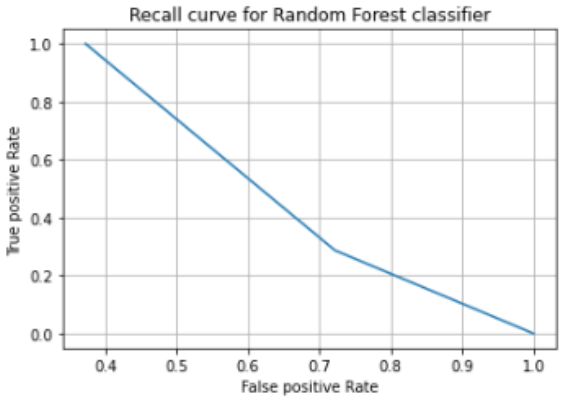
- **Accuracy_score**: aquesta mètrica calcula quantes observacions tant positives com negatives s'han classificat correctament. S'ha d'usar en problemes que estan equilibrats.
- **F1_score**: aquesta combina a parts iguals la precisió amb el recall. Per tant, té tant en compte una cosa com l'altre per treure un valor entre 0 i 1.
- **Average_precision_score**: aquesta també combina en una corba precisió i recall. Això fa que tinguem ambdues en una sola representació i ajudi a oferir una possible solució al dilema entre precisió i recall. Amb aquesta representació podem trobar el punt mitjà de combinació d'ambdues que seria el punt òptim per oferir el millor model possible.

b) Mostreu la Precisió-Recall Curve i la ROC Curve. Quina és més rellevant pel vostre dataset? Expliqueu amb les vostres paraules, la diferencia entre una i altre.

ROC de Random Forest:



Precision Recall Curve de Rando Forest:



Reports de classificació:

Random Forest:

```
In [18]: print(metrics.classification_report(y_test,pred_rf))
```

	precision	recall	f1-score	support
0	0.69	0.93	0.79	412
1	0.72	0.29	0.41	244
accuracy			0.69	656
macro avg	0.71	0.61	0.60	656
weighted avg	0.70	0.69	0.65	656

Regressió Logística:

```
In [19]: print(metrics.classification_report(y_test,pred_lg))
```

	precision	recall	f1-score	support
0	0.63	1.00	0.77	412
1	0.00	0.00	0.00	244
accuracy			0.63	656
macro avg	0.31	0.50	0.39	656
weighted avg	0.39	0.63	0.48	656

6. Hyperparameter Search

a) Quines formes de buscar el millor paràmetre heu trobat? Són costoses computacionalment parlant?

La forma més adequada de buscar aquests paràmetres que hem trobat ha estat calcular el model moltes vegades amb diferents paràmetres d'entrada.

El que hem fet ha estat calcular el model molts cops canviant els paràmetres d'entrada i finalment, extreure quina combinació d'entrades ha donat els millors resultats.

El problema d'aquest sistema és que, computacionalment parlant, és molt cost, ja que s'està calculant el model molts cops.

Aquest mètode pot no ser gaire eficient ni computacionalment ni temporalment. Hi ha altres possibles mètodes més eficients. Hi ha gaires llibreries que ofereixen altres mètodes.

b) Si disposem de recursos limitats (per exemple, un PC durant 1 hora) quin dels dos métodes creieu que obtindrà millor resultat final?

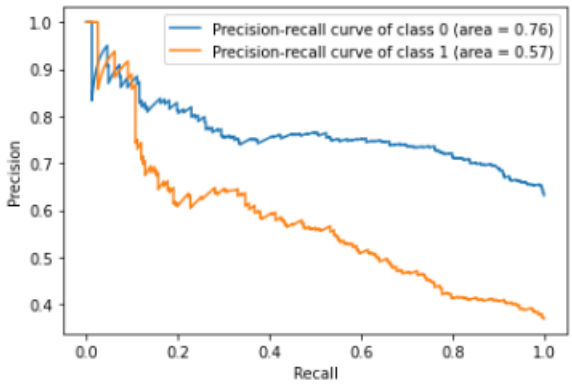
En el cas de disposar de recursos limitats el mètode que obtindria millors resultats en aquests temps i recursos limitats seria el de Randomized Parameter Optimization.

Apartat A – Comparativa de Models

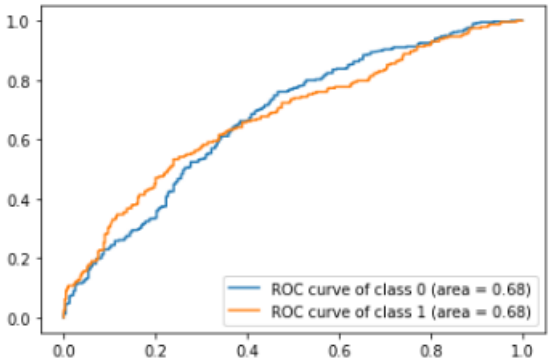
Hem fet el anàlisis de la nostre base de dades amb els següents models:

- SVM.
- Regressió logística.

Aquests són els resultats que hem arribat a obtenir per al Precission-Recall Curve:



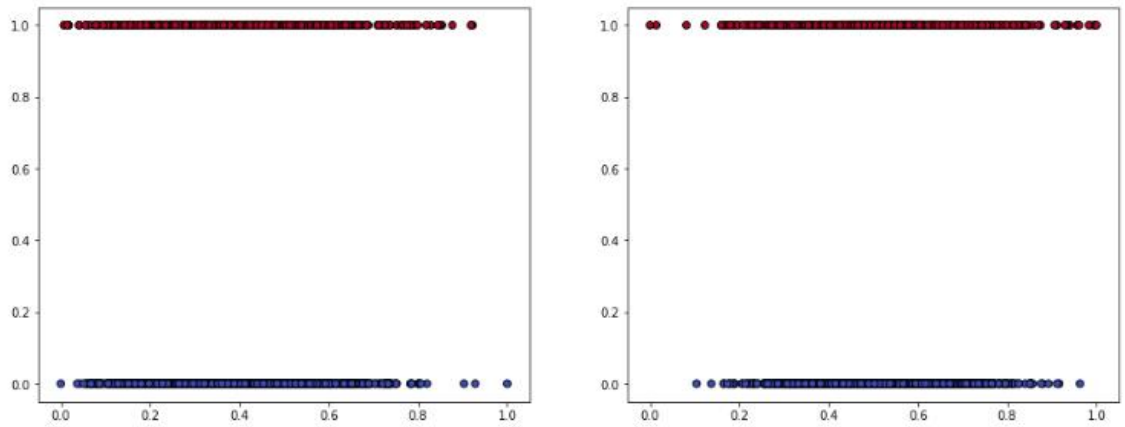
Per el ROC Curve aquests són els resultats que hem obtingut:



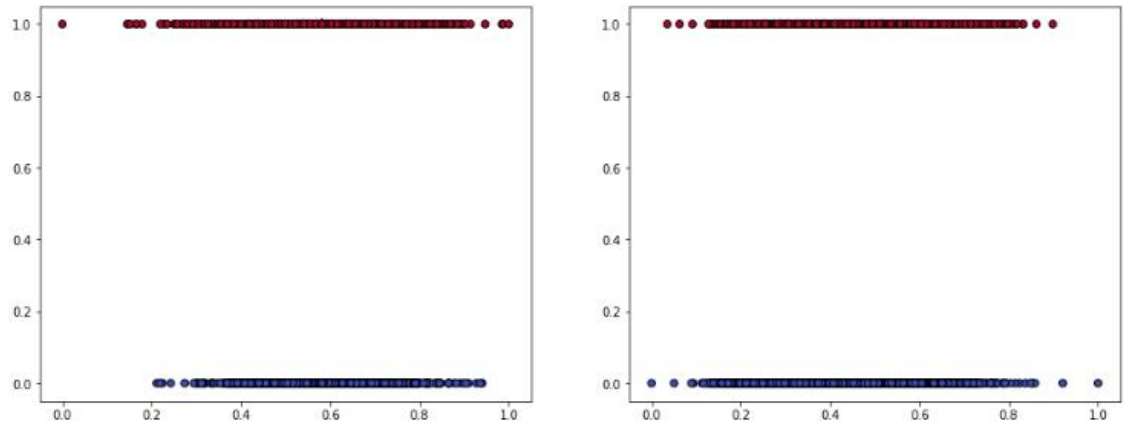
I aquestes són les estadístiques que obtenim sobre la classificació en funció dels dos models:

```
Correct classification Logistic  0.5 % of the data:  0.6092796092796092
Correct classification SVM       0.5 % of the data:  0.6526251526251526
Correct classification Logistic  0.7 % of the data:  0.6032553407934893
Correct classification SVM       0.7 % of the data:  0.6683621566632757
Correct classification Logistic  0.8 % of the data:  0.6310975609756098
Correct classification SVM       0.8 % of the data:  0.6585365853658537
```

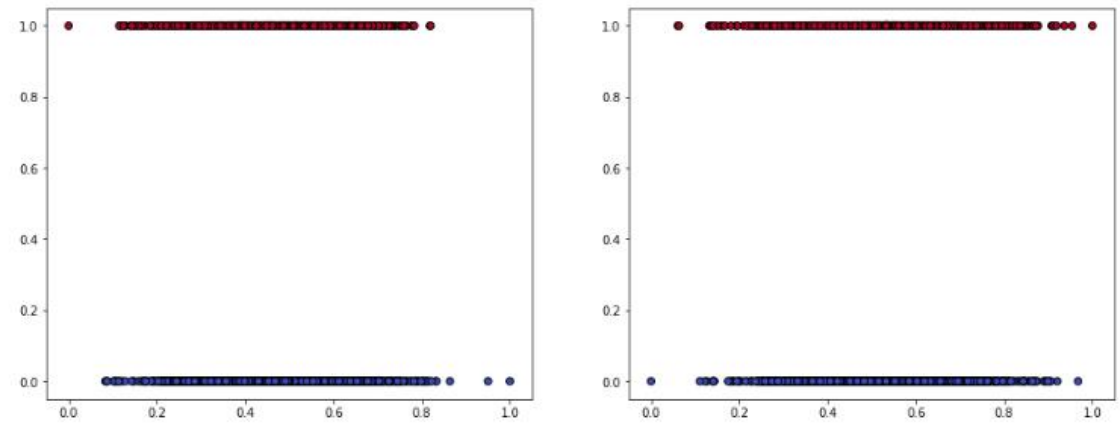
Classificació dels Atributs Solids & Chloramines:



Classificació dels Atributs Sulfate & Conductivity:



Classificació dels Atributs Organic_carbon & Trihalomethanes:



Classificació de l'Atribut Turbidity: