Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Verteilte System

Prof. Dr. Eirini Ntoutsi
Tai Le Quy,Damianos Melidis

Gull Jabeen	10010282
Fawad Abbasi	10008146

Data Mining I - SS17
Data Mining Project 1- Supervised Learning

Table of Contents

Scope:

In this project we have gone through the whole Knowledge Discovery in Databases (KDD) process, from dataset selection to preparation, mining and interpretation of the data mining results (also called patterns).The main focus of the project is on supervised learning.

Dataset:

This data set was created by MC (Monte Carlo1) simulations for the registration of high energy gamma particles by atmospheric gamma telescope. The features of the dataset are pulses created by the incoming Cherenkov photons on the photomultiplier tubes of the telescope. Depending on the received energy, the primary gamma, several Cherenkov photons get collected, in patterns(called the shower image), allowing to discriminate statistically those caused by primary *gammas signal* from the images of hadronic showers initiated by cosmic rays (class: signal) or *background signal* found in the upper atmosphere (class: background).

Tasks

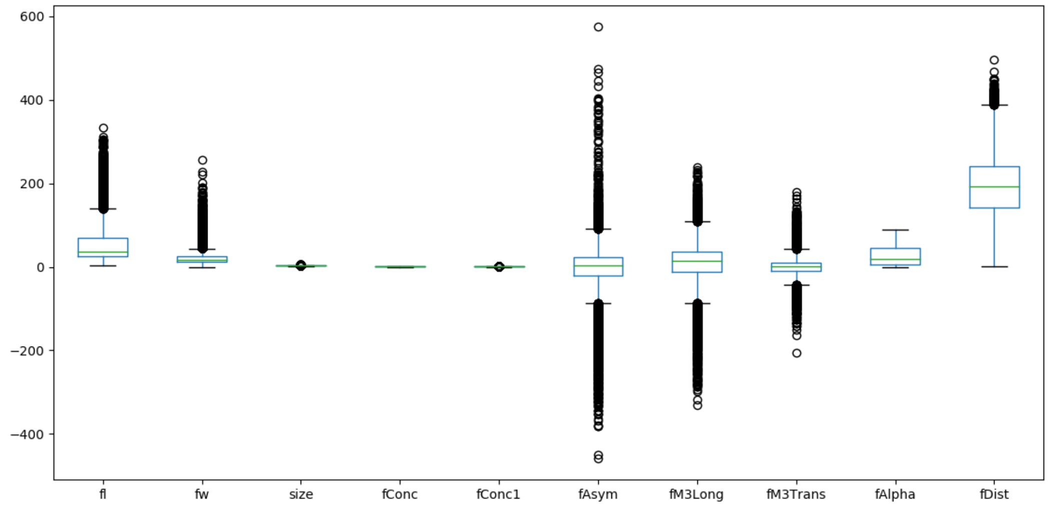
➢ Dataset understanding

○ Uni-variate analysis

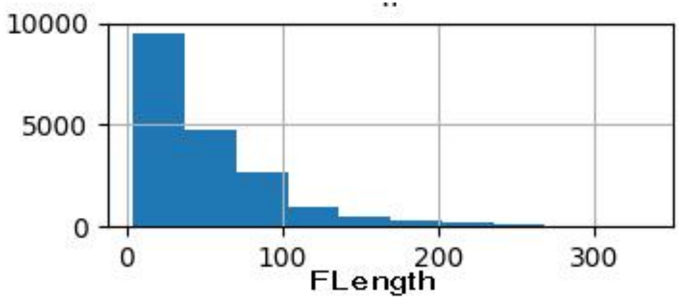
Uni-variate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and its major purpose is to describe; it takes data, summarizes that data and finds patterns in the data.

Below are the results for uni-variate analysis for our project.

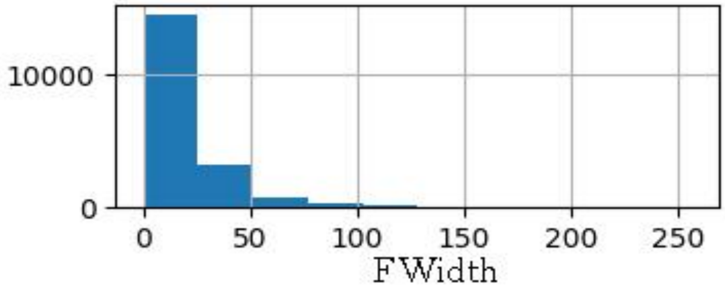
- ***Box Plot***



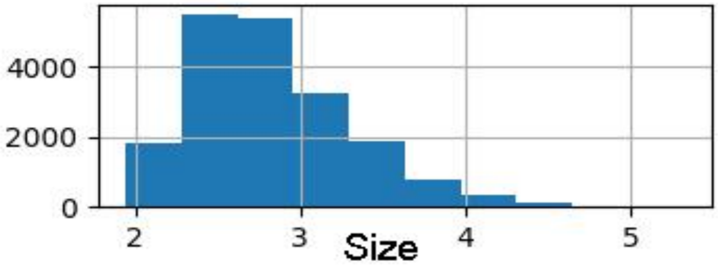
- ***Flength:***
  - ~85% drawn from the distribution are within 0 to 100.
  - Rest of the values lies between 100 and 250
  - None of the value is after 250.
  - Histogram is right skewed



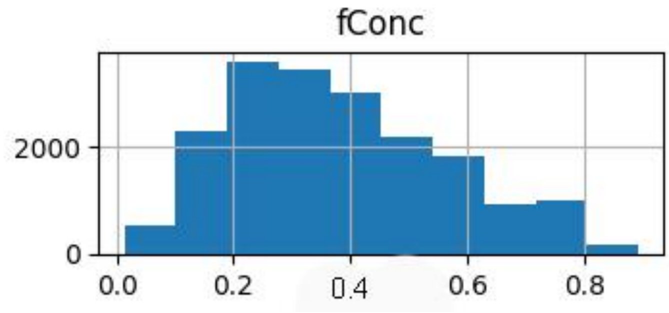
- ***Fwidth:***
  - ~98% values are between 0 to 75.
  - Others values lies between 75 and 120.
  - None of the value after 120.
  - Histogram is right skewed



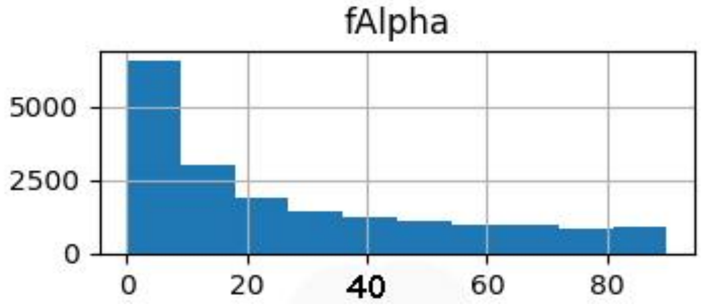
- ***Fsize:***
  - The distribution of fSize is highest in the bin with fSize values from 2.4-2.8.
  - The bin (fSize: 2.8-3.5) next to it has a frequency close to the former and stands second highest.
  - The distribution of fSize then goes on reducing constantly to the right.
  - Histogram is right skewed.



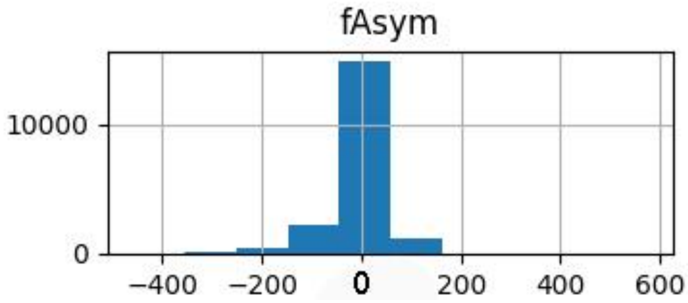
- ***fConc:***
  - fConc is highest in the bin with fConc values from 0.19-0.35.
  - fConc reduces on both sides.
  - Histogram is left skewed.



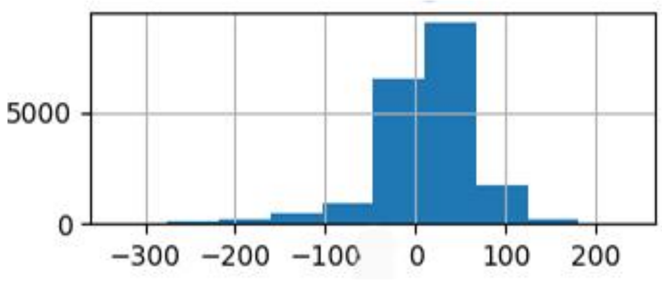
- ***fAlpha:***
  - Distribution of fAlpha is highest in the first bin with fAlpha values from 0 to 15.
  - The distribution of fAlpha then goes on reducing to the right.
  - The histogram is right skewed.



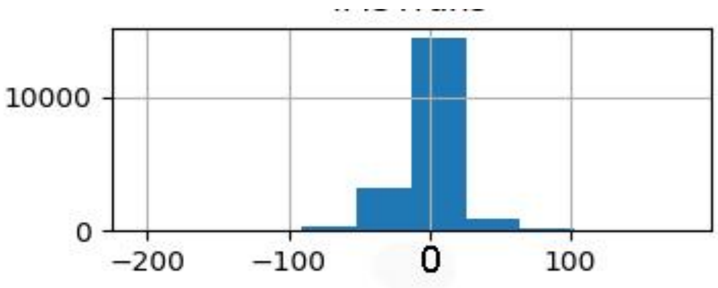
- ***fAsym:***
  - fConc is highest in the bin with fAsym values from -50 - 50
  - The distribution of fAsym on all bins on either sides of this bin is much lower.



- ***fM3Long:***
    - Distribution of fM3Long is highest with fM3Long values from -50 to 50.
    - The histogram is left skewed.

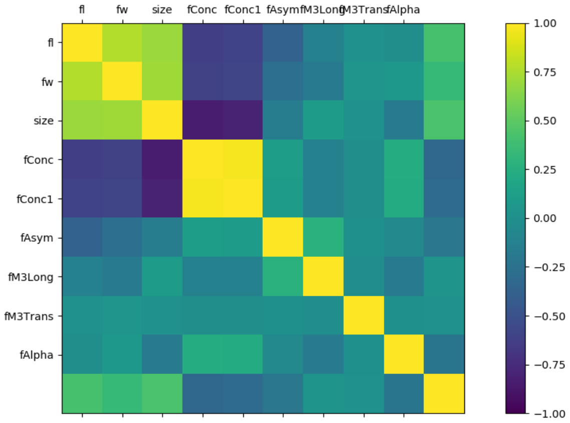


- ***fM3Trans:***
    - The distribution of fM3Trans is highest in the bin with fM3Trans values from -50 to 25.
    - The bin between -25 to -50 has the second highest frequency distribution.
    - The histogram is symmetric with normal distribution.



Bi-variate analysis (Correlations)

Bivariate analysis is the simultaneous analysis of two variables (attributes). It explores the concept of relationship between two variables, whether there exists an association and the strength of this association, or whether there are differences between two variables and the significance of these differences. There are three types of bivariate analysis.



- **Positive linear- correlation**

  The following pairs of attributes have a positive correlation with each other.

  - fLength vs fWidth
  - fM3Long vs fM3Trans
  - fSize vs fDist
  - fSize vs fM3Long
  - fSize vs fM3Trans
  - fLength vs fSize
  - fLength vs fAsym
  - fLength vs fM3Long
  - fLength vs fM3Trans
  - fWidth vs fSize
  - fWidth vs fM3Long
  - fWidth vs fM3Trans
  - fConc vs fConc1
  - fAsym vs fM3Long

- **Negative linear- correlation**

  The following pairs of attributes have a negative correlation with each other.

  - fLength vs fConc
  - fLength vs fConc1
  - fWidth vs fConc
  - fWidth vs fConc1
  - fSize vs fConc
  - fSize vs fConc1
  - fConc vs fM3Long
  - fConc vs fM3Trans
  - fConc1 vs fM3Long
  - fConc1 vs fM3Trans

- **No correlation**
  - fLength vs fAlpha
  - fLength vs fDist
  - fWidth vs fAsym
  - fWidth vs fAlpha
  - fWidth vs fDist
  - fSize vs fAsym
  - fSize vs fAlpha
  - fConc vs fAsym
  - fConc vs fAlpha

- fConc vs fDist
- fConc1 vs fAsym
- fConc1 vs fAlpha
- fConc1 vs fDist
- fAsym vs fM3Trans
- fAsym vs fAlpha
- fAsym vs fDist
- fM3Long vs fAlpha
- fM3Long vs fDist
- fM3Trans vs fAlpha
- fM3Trans vs fDist
- fAlpha vs fDist
- fLength vs Class
- fWidth vs Class
- fSize vs Class
- fConc vs Class
- fConc1 vs Class
- fAsym vs Class
- fM3Long vs Class
- fM3Trans vs Class
- fAlpha vs Class
- fDist vs Class

Checking for imbalance: Class distribution

The classes (g, h) are not equally distributed. The distribution of g is double to the distribution of the class h.

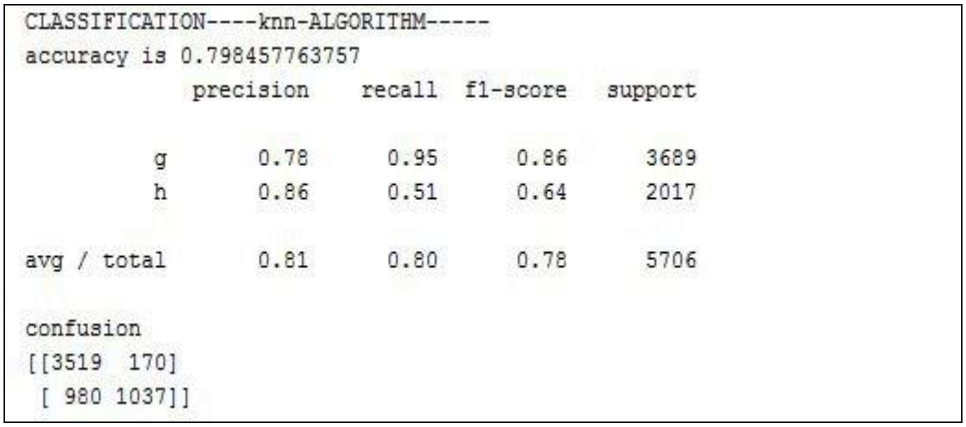
Preprocessing/Transformation

Final dataset:

- fLength
- fWidth
- fSize
- fConc
- fConc1
- fAsym
- fM3Long
- fM3Trans
- fAlpha
- fDist

➢ Classification

• Knn Algorithm



```
CLASSIFICATION----knn-ALGORITHM-----
accuracy is 0.798457763757
             precision    recall  f1-score   support

          g       0.78      0.95      0.86      3689
          h       0.86      0.51      0.64      2017

avg / total       0.81      0.80      0.78      5706

confusion
[[3519  170]
 [ 980 1037]]
```

• *Performance measures*

- Accuracy = 0.798 = 79.8%
- Sensitivity = 0.95 = 95%
- Specificity = 0.51 = 51%
- F1-score = 0.86 = 86%

• *Confusion Matrix:*

	Predicted			Total
Actual		g	h	
	g	3519	170	3689
	h	980	1037	2017
Total		4499	1207	5706

- More appropriate
  - sensitivity
  - specificity
  - F1-score

Accuracy is not giving a correct measure .The number of predicted examples for ‘g’ and ‘h ’as predicted by Knn algorithm are 4278 and 1428 respectively. Hence there is an imbalance.

- Accuracy is a bad measure

  Despite a low specificity value of 51%, accuracy of the classifier is still 79.8% which is much higher. Therefore accuracy is not a good evaluation measure when there is a class imbalance in this scenario.

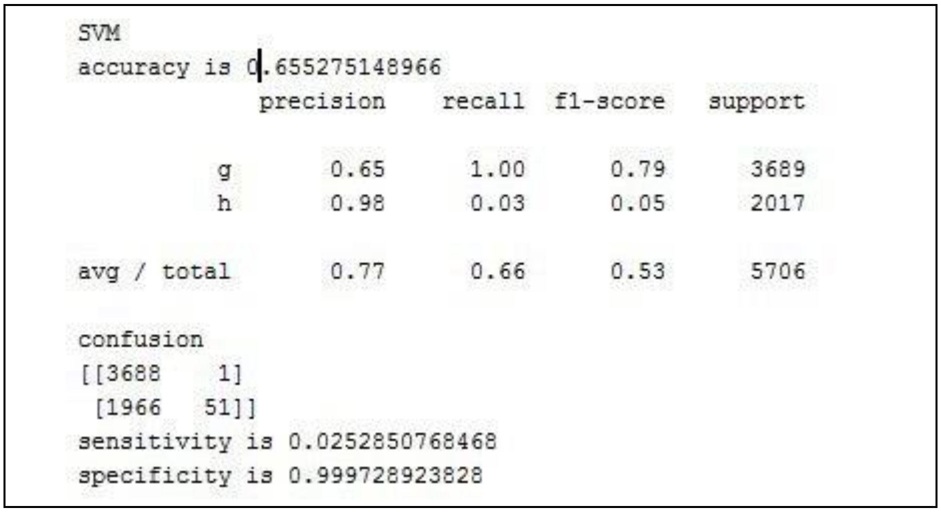
- Performance of the Learner

  The learner with a high sensitivity value of 95% works well in classifying positive values (or ‘g’). On the other hand, with a lower specificity (or true negative rate) value of 51% does not perform well in classifying the negative values (or ‘h’). Due to this reason, the learner attains a higher recall of 95% but a lower precision of 86% thereby having an average harmonic mean (or F1-score) of 86%. This can also be visualized in the ROC curve with a high AUC of 0.85. The overall performance of the learner is pretty good with an accuracy of 79.8%.

- Model visualization

  The decision boundary for Knn classifier consists of arbitrary shaped boundaries. It does not form axes parallel boundaries in contrast to decision trees.

Svm Algorithm



```
SVM
accuracy is 0.655275148966
             precision    recall  f1-score   support

          g       0.65      1.00      0.79      3689
          h       0.98      0.03      0.05      2017

avg / total       0.77      0.66      0.53      5706

confusion
[[3688    1]
 [1966   51]]
sensitivity is 0.0252850768468
specificity is 0.999728923828
```

Performance measures

- Accuracy = 0.655 = 65.5%
- Sensitivity = 0.025 = 2.5%
- Specificity = 0.03 = 3%
- F1-score = 0.79 = 79%

- ***Confusion Matrix:***

	Predicted			Total
Actual		g	h	
	g	3688	1	3689
	h	1966	51	2017
Total		5654	52	5706

- More appropriate
  - sensitivity
  - specificity
  - F1-score

  Accuracy is not giving a correct measure. there is an imbalance.

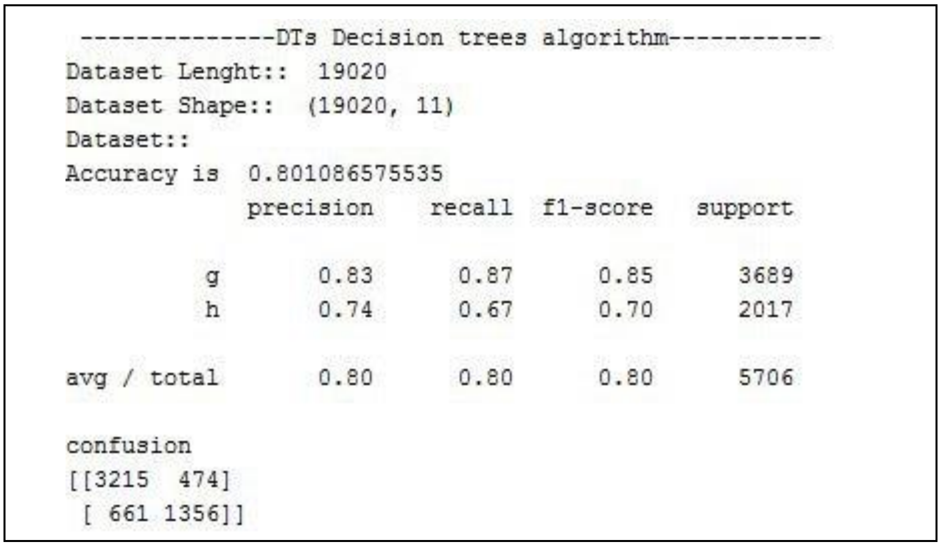
- Accuracy is a bad measure

  Despite a very low specificity value of 3%, accuracy of the classifier is still 65% which is much higher. Therefore accuracy is not a good evaluation measure when there is a class imbalance in this scenario.

- Model visualization

  The decision boundary for SVM classifier consists of arbitrary shaped boundaries. It does not form axes parallel boundaries in contrast to decision trees.

- **Decision Tree Algorithm**



```
 --------------DTs Decision trees algorithm-----------
Dataset Lenght::  19020
Dataset Shape::  (19020, 11)
Dataset::
Accuracy is  0.801086575535
             precision    recall  f1-score   support

          g       0.83      0.87      0.85      3689
          h       0.74      0.67      0.70      2017

avg / total       0.80      0.80      0.80      5706

confusion
[[3215  474]
 [ 661 1356]]
```

- ***Performance measures***
    - Accuracy = 0.801 = 80.1%
    - Sensitivity = 0.87 = 87%
    - Specificity = 0.67 = 67%
    - F1-score = 0.85 = 85%
- ***Confusion Matrix:***

	Predicted			Total
Actual		g	h	
	g	3215	474	3689
	h	661	1356	2017
Total		3876	1830	5706

- More appropriate
    - sensitivity
    - specificity
    - F1-score

  Accuracy is not giving a correct measure. There is an imbalance.

- Accuracy is a bad measure

  Accuracy is not a good evaluation measure when there is a class imbalance in this scenario.

- Model visualization

  The decision boundaries for the decision tree classifier are parallel to the axes and form hyper rectangles as opposed to the ones in case of the other two classifiers.

Interpretation

Knn Algorithm

It is a lazy algorithm. What this means is that it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This means the training phase is pretty fast. Lack of generalization means that KNN keeps all the training data. More exactly, all the training data is needed during the testing phase. (Well this is an exaggeration, but not far from truth). This is in contrast to other techniques like SVM where you can discard all non-support vectors without any problem.

- **SVM**

  "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems.

- **Decision Trees Classifier:**

  The decision tree classifier builds a classification model based on the data from the training set. It builds a decision tree with each feature which is likely to give maximum gain being the node and performing a split at that node.

References:

https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/

https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

http://www.statisticshowto.com/univariate/

http://www.saedsayad.com/bivariate_analysis.htm

http://machinelearningmastery.com/visualize-machine-learning-data-python-pandas/

http://machinelearningmastery.com/quick-and-dirty-data-analysis-with-pandas/

https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope