IBM DATA SCIENCE CAPSTONE PROJECT

Analysis of Bank Note Authentication

By Anuska Saha

Nov 2021

Table of Contents

1.EXECUTIVE SUMMARY

Banknotes are one of the most important assets of a country. Some miscreants introduce fake notes which bear a resemblance to original note to create discrepancies of the money in the financial market. It is difficult for humans to tell true and fake banknotes apart especially because they have a lot of similar features. Fake notes are created with precision, hence there is need for an efficient algorithm which accurately predicts whether a banknote is genuine or not. This paper proposes machine learning techniques to evaluate authentication of banknotes. Supervised learning algorithms such as Support Vector Machine (SVM) and Unsupervised learning algorithm like K-Means Clustering are used for differentiating genuine banknotes from fake ones. The study also shows the comparison of these algorithms in classification of banknotes.


£20
£20
Bank of England
£10
Bank of England
£5
LONDON
FOR THE GOVERNOR AND COMPANY
OF THE BANK OF ENGLAND
CHIEF CASHIER
England
5
Pounds

2.INTRODUCTION

2.1 NATURE OF ANALYSIS

Business Analysis

- Despite a decrease in the use of currency due to the recent growth in the use of electronic transactions, cash transactions remain very important in the global market.
- Banknotes are used to carry out financial activities. To continue with smooth cash transactions, entry of forged banknotes in circulation should be preserved.
- There has been a drastic increase in the rate of fake notes in the market. Fake money is an imitation of the genuine notes and is created illegally for various motives. These fake notes are created in all denominations which brings the financial market of the country to a low level.

2.2 ANALYTICAL APPROACH

- In the recent years, Soft computing techniques have been widely used to solve problems that are difficult to solve using conventional mathematical methods. Supervised learning techniques are widely used in classification problems. This paper evaluates supervised machine learning algorithms to classify genuine and fake notes, and compares algorithms on the basis of accuracy, sensitivity, and specificity. Consider someone wants to deposit money in the bank. The notes that are to be deposited are given to a human being to check for their authenticity

2.3 PROBLEM STATEMENT

Security aspects of banknotes have to be considered and security features are to be introduced to mitigate fake currency. Hence, there is a dire need in banks and ATM machines to implement a system that classifies a note as genuine or fake. As the fake notes are prepared with precision, it is difficult to differentiate them from genuine ones. A recognition system must be installed to detect legitimacy of the note. The system should extract the features of the note using image processing techniques. These features will be given as input to the machine learning algorithm which will predict if the note is true or fake.

2.4 MOTIVATION OF OUR RESEARCH

Despite a decrease in the use of currency due to the recent global expansion in electronic financial transactions, transactions in real money continue to be very important in the global market . While performing transactions in real money, touching and counting notes by hand is still a common practice in daily life, but the use of various types of automated machines has become essential for large-scale and safe transactions. Such automated self-service machines include automated teller machines (ATMs) for money deposits and withdrawals, as well as financial transactions , banknote counters and coin counters , mostly used in banks, and automatic vending machines, into which money is inserted to purchase goods . These devices must be equipped with four essential functions: banknote recognition, counterfeit banknote detection, serial number recognition, and fitness classification. While limited surveys have been conducted in previous studies on the areas of banknote recognition and counterfeit banknote recognition, this paper is the first survey of its kind to review all four areas. This lack of research is ascribable to the fact that banknote recognition studies have been mostly carried out in industrial settings rather than for academic purposes.

METHODOLOGY

3.1 SCOPE OF OUR RESEARCH & ANALYSIS

Banknote recognition generally concerns classification of banknotes by denomination, i.e., the currency amount of a note of a specific country. This classification also enables recognition of the year of printing and input direction of the classified denomination. In some studies, the scope of recognition is extended to simultaneous recognition of two or more national currencies.

Counterfeit banknote detection generally concerns methods for distinguishing between genuine and fake notes. As shown in the example of a genuine and a counterfeit USD 100 bill in Figure 1a,b, respectively, a validation check is done by examining anti-counterfeiting features

3.2 EXPLANATION OF THE DATA SOURCES

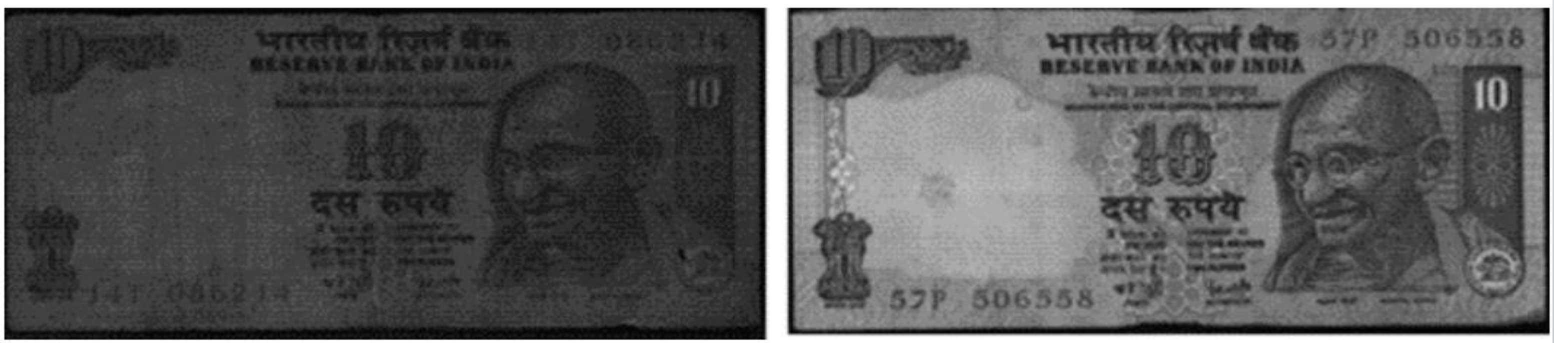
A banknote serial number is a unique alphanumerical identifier engraved on each banknote in the banknote production process. It contains the name of the issuing bank and serial information of each denomination ; The Figure2 shows the serial number of a USD 100 bill. Since each banknote has its own unique serial number, it can be used to trace its source and circulation route and can thus be efficiently used to detect counterfeit banknotes. Example of serial number code (USD 100 bill). A banknote serial number is a unique alphanumerical identifier engraved on each banknote in the banknote production process. It contains the name of the issuing bank and serial information of each denomination ; Figure 2 shows the serial number of a USD 100 bill. Since each banknote has its own unique serial number, it can be used to trace its source and circulation route and can thus be efficiently used to detect counterfeit banknotes. Related technologies are described in detail in next coming Section

Figure 1. Example of genuine and counterfeit banknotes (USD 100 bill): (a) a genuine banknote; (b) a counterfeit banknote


(a) (b)

. Figure 2. Example of serial number code (USD 100 bill

Fitness classification of banknotes generally concerns methods for classifying banknotes according to their physical conditions, such as soiling. As shown in the example of two INR 10 bills in Figure 3, banknotes of the same denomination may exhibit fit or unfit conditions, which include soiling and creases (Figure 3a), depending on circulation intensity and climate conditions. In order to maintain the fitness of banknotes in circulation, automated self-service terminals, such as ATMs, need to be equipped with a fitness classification function to sort out and retrieve unfit banknotes. Retrieving unfit banknotes is also necessary for preventing banknote classification errors.



(a) (b)

4.COLLECTION OF DATA

4.1 OVERVIEW OF DATA

This dataset is about distinguishing genuine and forged banknotes. Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. A Wavelet Transform tool was used to extract features from these images.

4.2 DATASET DESCRIPTION

- The dataset used to train the models is taken from UCI machine learning repository . Data were extracted from genuine and counterfeit banknote images. The dataset has 1372 instances. There are 5 attributes out of which 4 are the features and one is the target attribute. The dataset contains a balanced ratio of both classes which is 55:45(genuine: counterfeit). The target class contains two values: 0 and 1 where 0 represents genuine note and 1 represents fake note.

Table 1. A tabular Dataset Description

Dataset Characteristics	Multivariate	Number of Instances	1372
Attributes Characteristics	Real	5	Number of Attribute

Table 2. Dataset description

Attribute Name	***Value Type***	***Description***
Variance of Wavelet Transformed Image	Continuous	Variance finds how each pixel varies from the neighboring pixels and classifies them into different regions
Skewness of Wavelet Transformed image	Continuous	Skewness is the measure of the lack of symmetry
Kurtosis of Wavelet Transformed image	Continuous	Kurtosis is a measure of whether the data are heavy tailed or light-tailed relative to a normal distribution
Entropy of image	Continuous	Image is used to describe the amount of information which must be coded for, by a compression algorithm
Class	Integer	Class contains two values 0 representing genuine note and 1 representing fake note

5.MODELLING

5.1 MACHINE LEARNING SUPERVISED ALGORITHMS APPLIED

- Logistic Regression CV
- Linear SVC
- Decision Tree Classifier
- Random Forest classifier
- K Nearest Neighbours
- CatBoost Classifier
- XGBoostClassifier

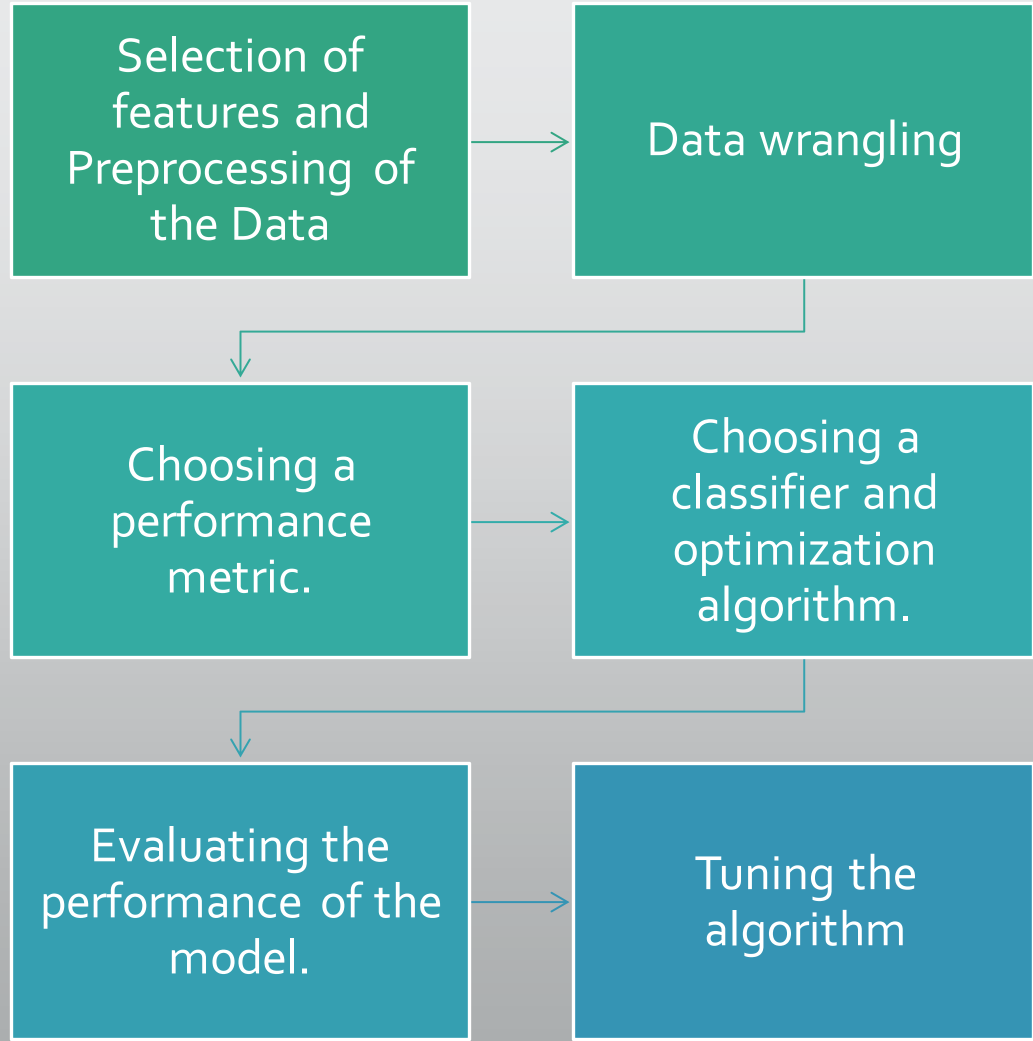
5.1.1Comparative Analysis of Algorithms in Supervised Classification

- Choosing a Classification Algorithm

Choosing an appropriate classification algorithm for a particular problem task requires practice: each algorithm has its own quirks and is based on certain assumptions. The "No Free Lunch" theorem: no single classifier works best across all possible scenarios. In practice, it is always recommended that you compare the performance of at least a handful of different learning algorithms to select the best model for the particular problem; these may differ in the number of features or samples, the amount of noise in a dataset, and whether the classes are linearly separable or not.

Eventually, the performance of a classifier, computational power as well as predictive power, depends heavily on the underlying data that are available for learning.

5.1.2.The five main steps that are involved in training a machine learning algorithm can be summarized as follows:



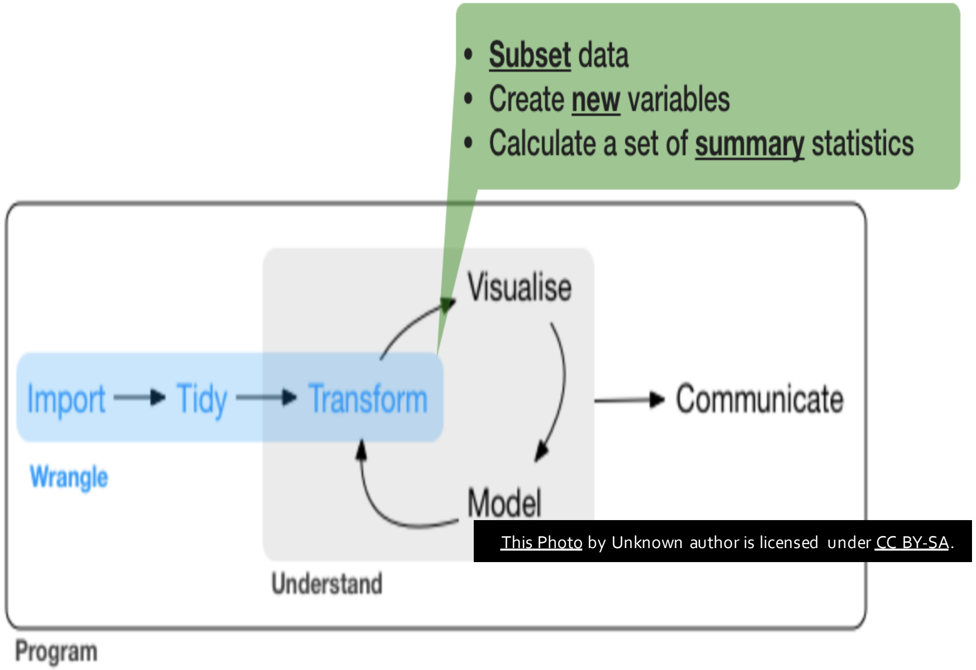
Data Preprocessing

- Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data preprocessing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.
- **Steps in Data Preprocessing in Machine Learning**
  - **1. Acquire the dataset**
  - **2. Import all the crucial libraries**
  - **3. Import the dataset**
  - **4. Identifying and handling the missing values**
  - **5.Splitting the dataset**
  - **6.Feature Scaling**

- In the Data Preprocessing technique we are accessing the data set **"BankNote_Authentication.csv".**
- Importing the libraries such as pandas ,numpy, seaborn ,matplotlib
- Checking for the null values
- Splitting the training and testing Dataset.To evaluate how well a trained model performs on unseen data, we will further split the dataset into separate training and test datasets. Splitting data into 80% training and 20% test data
- We loaded the StandardScaler class from the preprocessing module and initialized a new StandardScaler object that we assigned to the variable sc. Using the fit method, StandardScaler estimated the parameters μ (sample mean) and (standard deviation) for each feature dimension from the training data. By calling the transform method, we then standardized the training data using those estimated parameters μ and . Note that we used the same scaling parameters to standardize the test set so that both the values in the training and test dataset are comparable to each other.

Data Wrangling

Data wrangling involves processing the data in various formats like - merging, grouping, concatenating etc. for the purpose of analysing or getting them ready to be used with another set of data. . In this chapter we will look at few examples describing these methods.



Measuring our classifier using Binary classification performance metrics

- A variety of metrics exist to evaluate the performance of binary classifiers against trusted labels. The most common metrics are accuracy, precision, recall, F1 measure, and ROC AUC score. All of these measures depend on the concepts of true positives, true negatives, false positives, and false negatives. Positive and negative refer to the classes. True and false denote whether the predicted class is the same as the true class.
- For our Banknote classifier, a true positive prediction is when the classifier correctly predicts that a note is authentic. A true negative prediction is when the classifier correctly predicts that a note is fake. A prediction that a fake note is authentic is a false positive prediction, and an authentic note is incorrectly classified as fake is a false negative prediction.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) in Python is the first step in your data analysis process developed by "**John Tukey**" in the 1970s. In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. By the name itself, we can get to know that it is a step in which we need to explore the data set.

EDA with SQL

We'll take a look at our data:

In [2]:

```
df.head()
```

Out[2]:

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

```
df.tail()
```

	Variance	Skewness	Curtosis	Entropy	Class
1368	0.40614	1.3492	-1.4501	-0.55949	1
1369	-1.3887	-4.8773	6.4774	0.34179	1
1370	-3.7503	-13.4586	17.5932	-2.7771	1
1371	-3.5637	-8.3827	12.393	-1.2823	1
1372	-2.5419	-0.65804	2.6842	1.1952	1

- Checking for NULL values

```
dataset.isnull().sum()
```

```
variance    0
skewness    0
curtosis    0
entropy     0
class       0
dtype: int64
```

```
[ ] dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1372 entries, 0 to 1371
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   variance  1372 non-null   float64
 1   skewness  1372 non-null   float64
 2   curtosis  1372 non-null   float64
 3   entropy   1372 non-null   float64
 4   class     1372 non-null   int64  
dtypes: float64(4), int64(1)
memory usage: 53.7 KB
```

```
[ ] dataset.describe()
```

	variance	skewness	curtosis	entropy	class
count	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000
mean	0.433735	1.922353	1.397627	-1.191657	0.444606
std	2.842763	5.869047	4.310030	2.101013	0.497103
min	-7.042100	-13.773100	-5.286100	-8.548200	0.000000
25%	-1.773000	-1.708200	-1.574975	-2.413450	0.000000
50%	0.496180	2.319650	0.616630	-0.586650	0.000000
75%	2.821475	6.814625	3.179250	0.394810	1.000000
max	6.824800	12.951600	17.927400	2.449500	1.000000

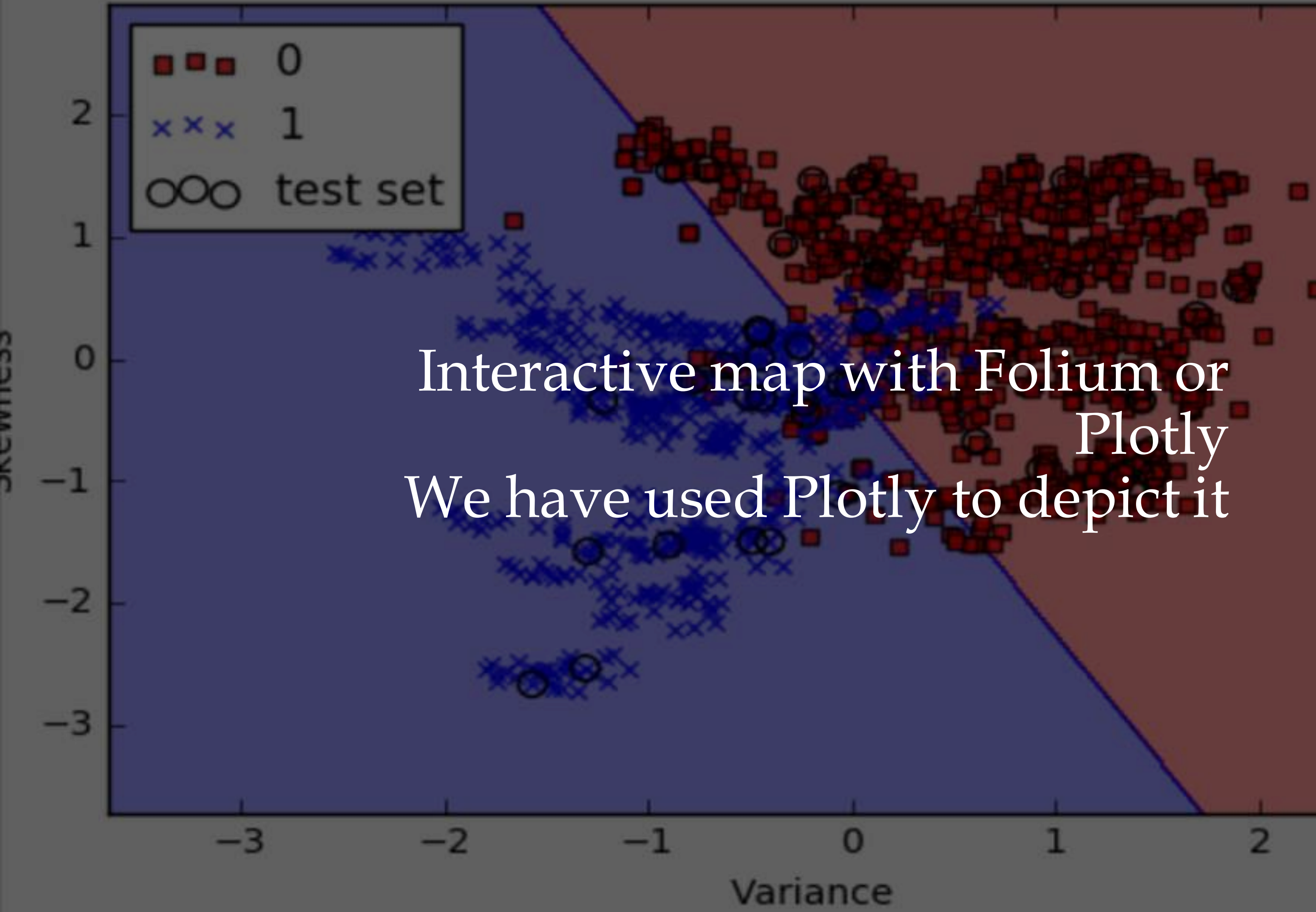
```
[118] import pandasql as ps
```

```
q1 = """SELECT  count(variance)as "variance of Wavelet Transformed image"
FROM df group by class"""
grouped_df = ps.sqldf(q1, locals())
grouped_df
```

	variance of Wavelet Transformed image
0	762
1	610

```python
q1 = """SELECT  count(skewness)as "skewness of Wavelet Transformed image"
FROM df group by class"""
grouped_df = ps.sqldf(q1, locals())
grouped_df
```

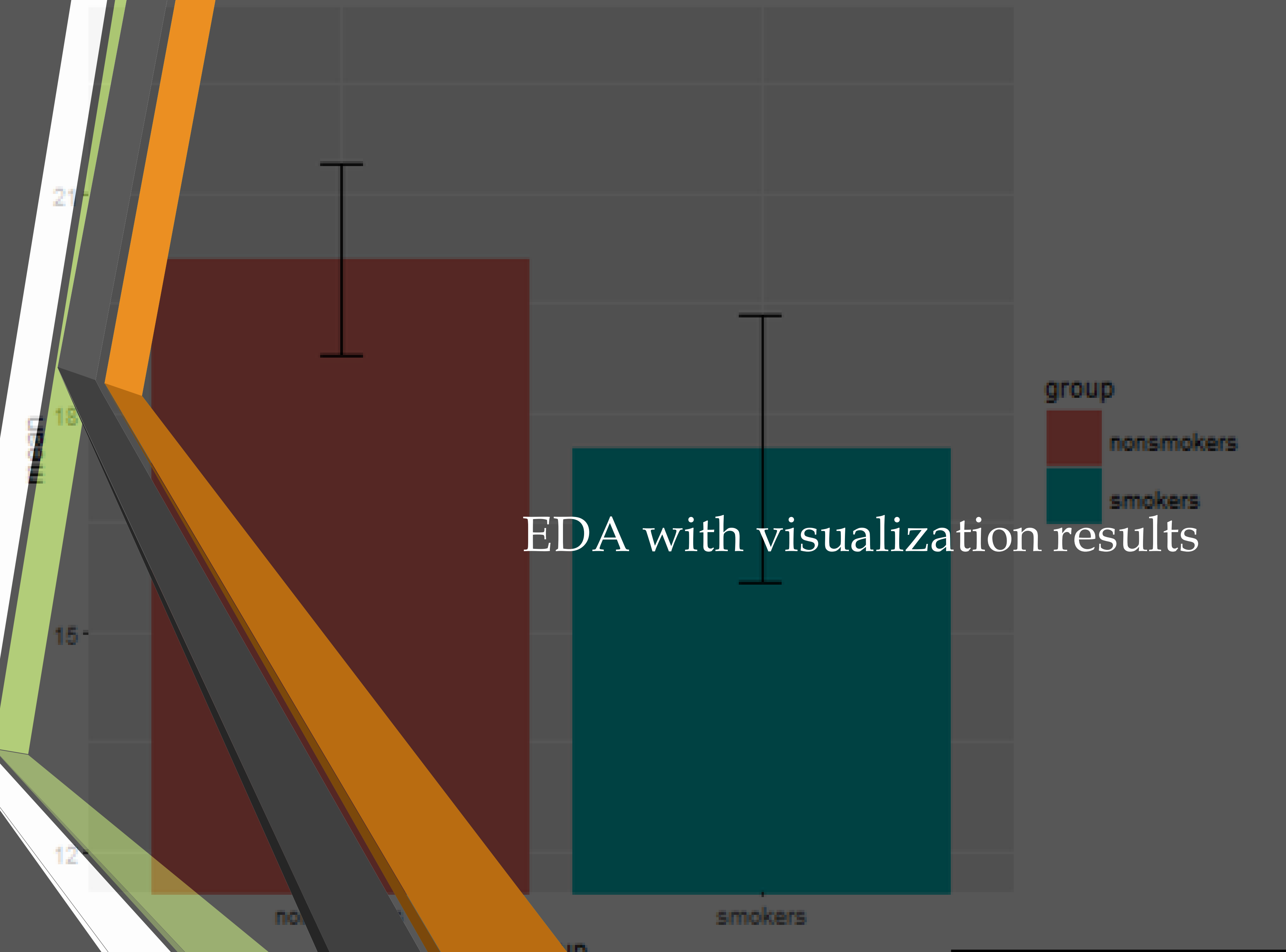
	skewness of Wavelet Transformed image
0	762
1	610

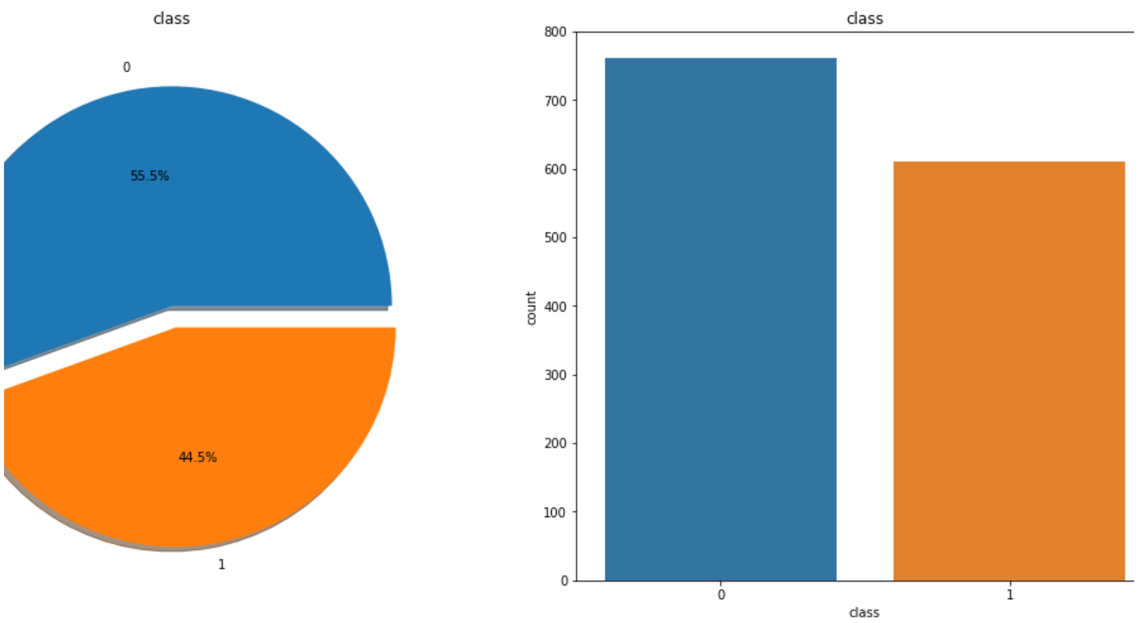


Interactive map with Folium or Plotly

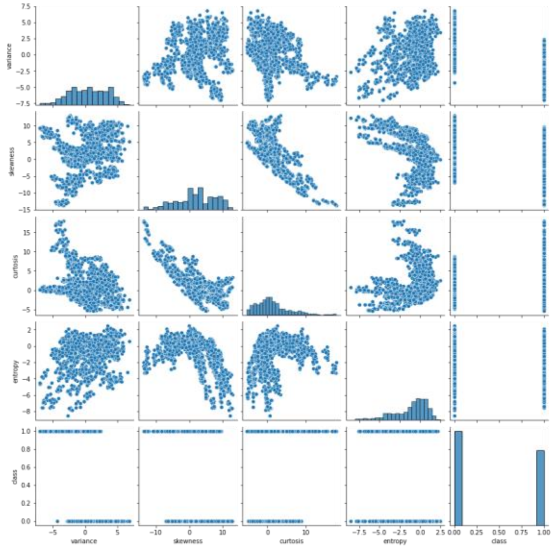
We have used Plotly to depict it

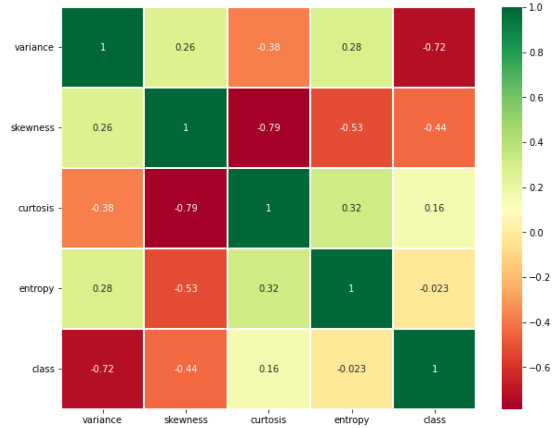
EDA with visualization results





Separating dependent and independent variables



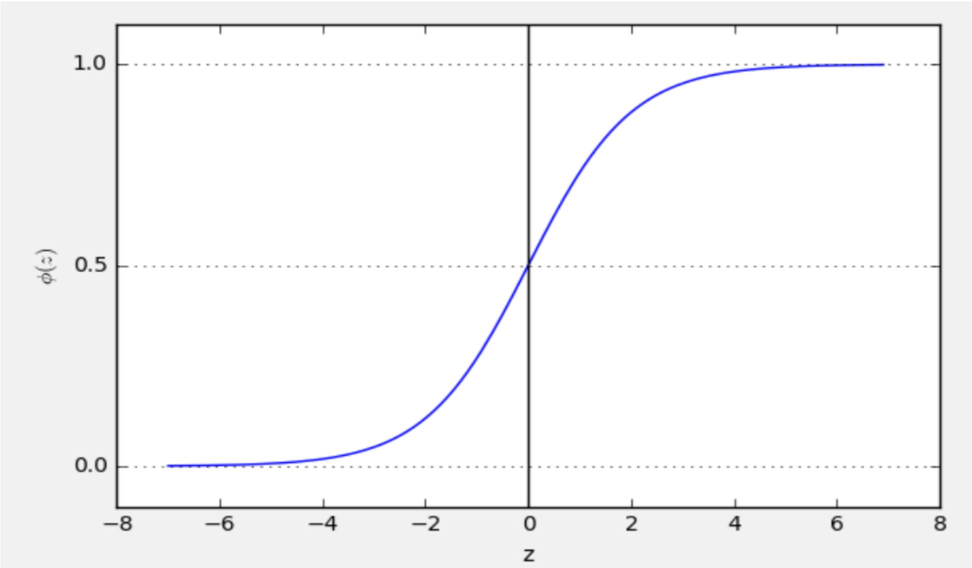


Supervised Classification Algorithm Applied

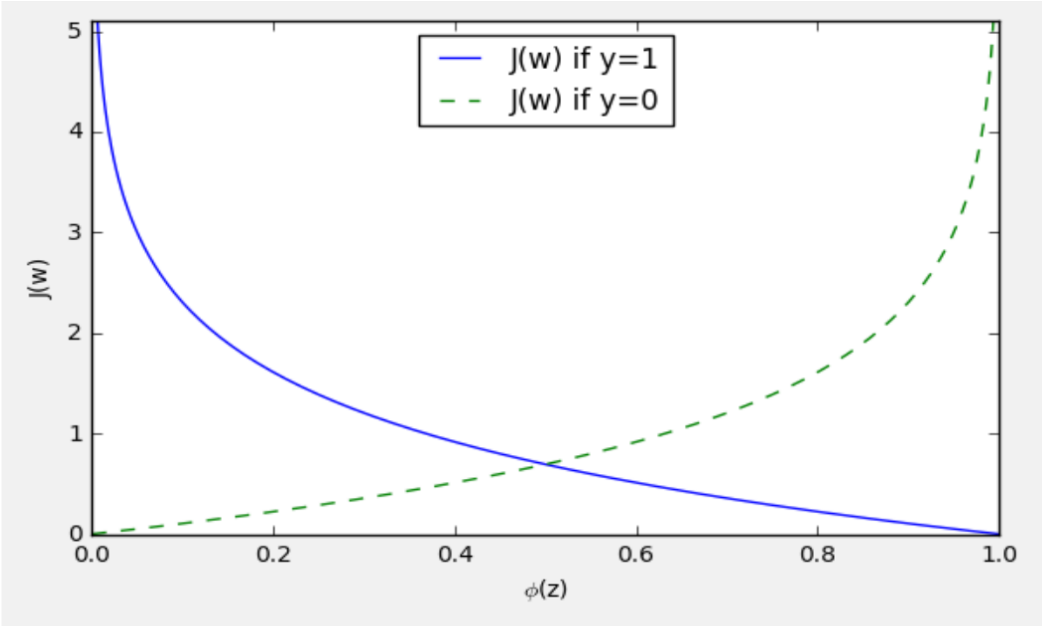
➢ Using Logistic Regression

- Logistic regression is a classification model that is very easy to implement but performs very well on linearly separable classes. It is one of the most widely used algorithms for classification in industry. The logistic regression model is a linear model for binary classification that can be extended to multiclass classification via the OvR technique.

Exploratory Data interactive visual analytics methodology

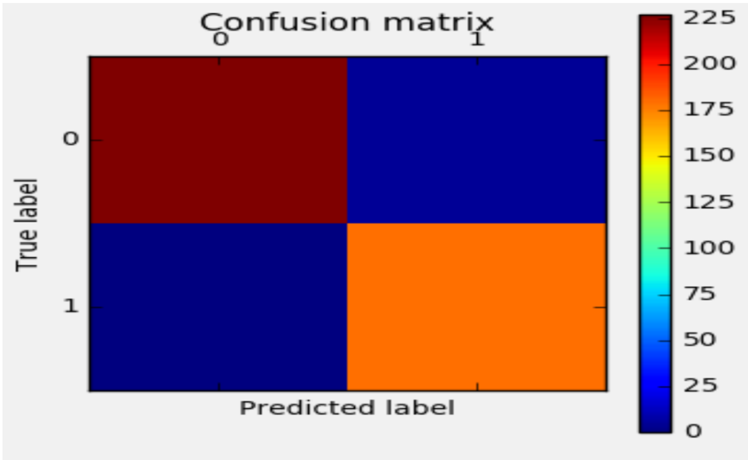


The sigmoid function used in the Logistic Regression

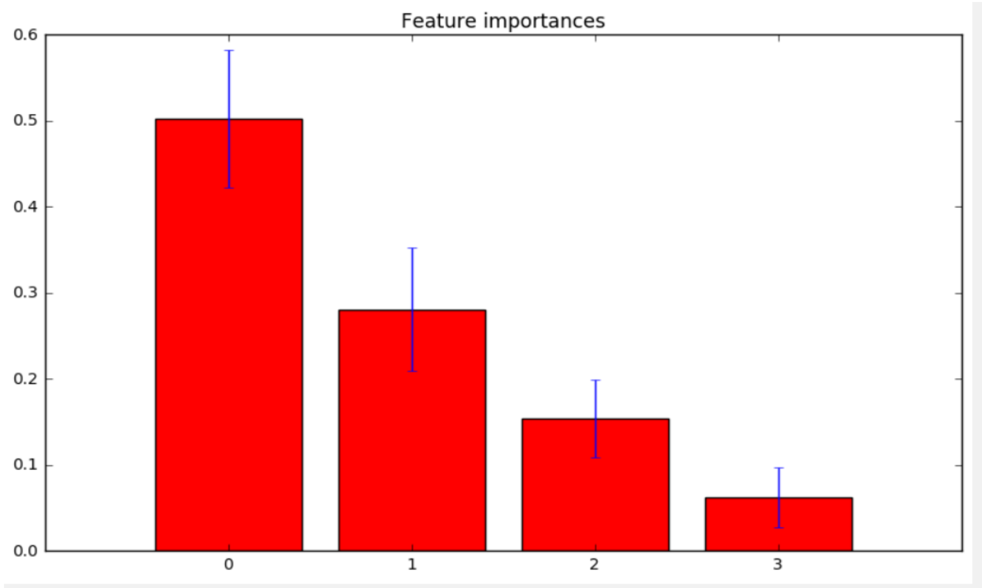


Learning the weights of the logistic cost function

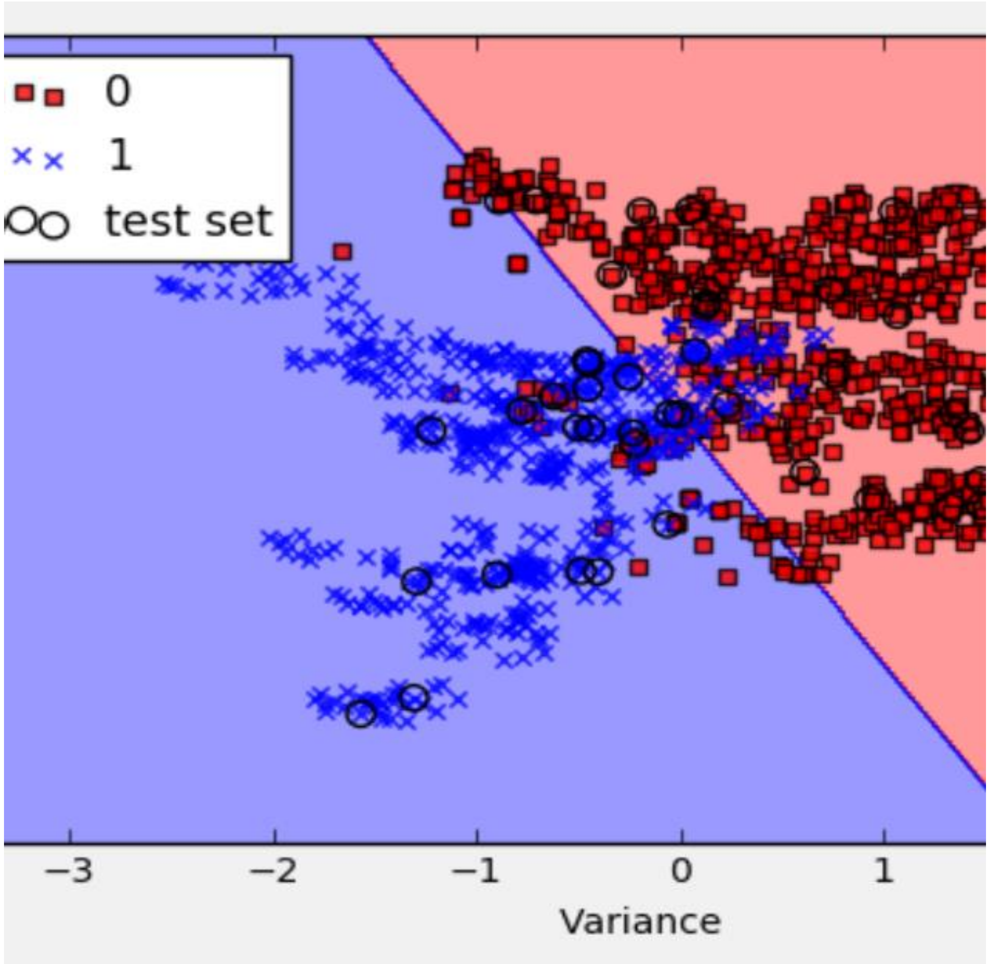
- The confusion matrix indicates that there were 227 true negative predictions, 180 true positive predictions, 0 false negative predictions, and 5 false positive prediction.



- Finding the most important features with forests of trees

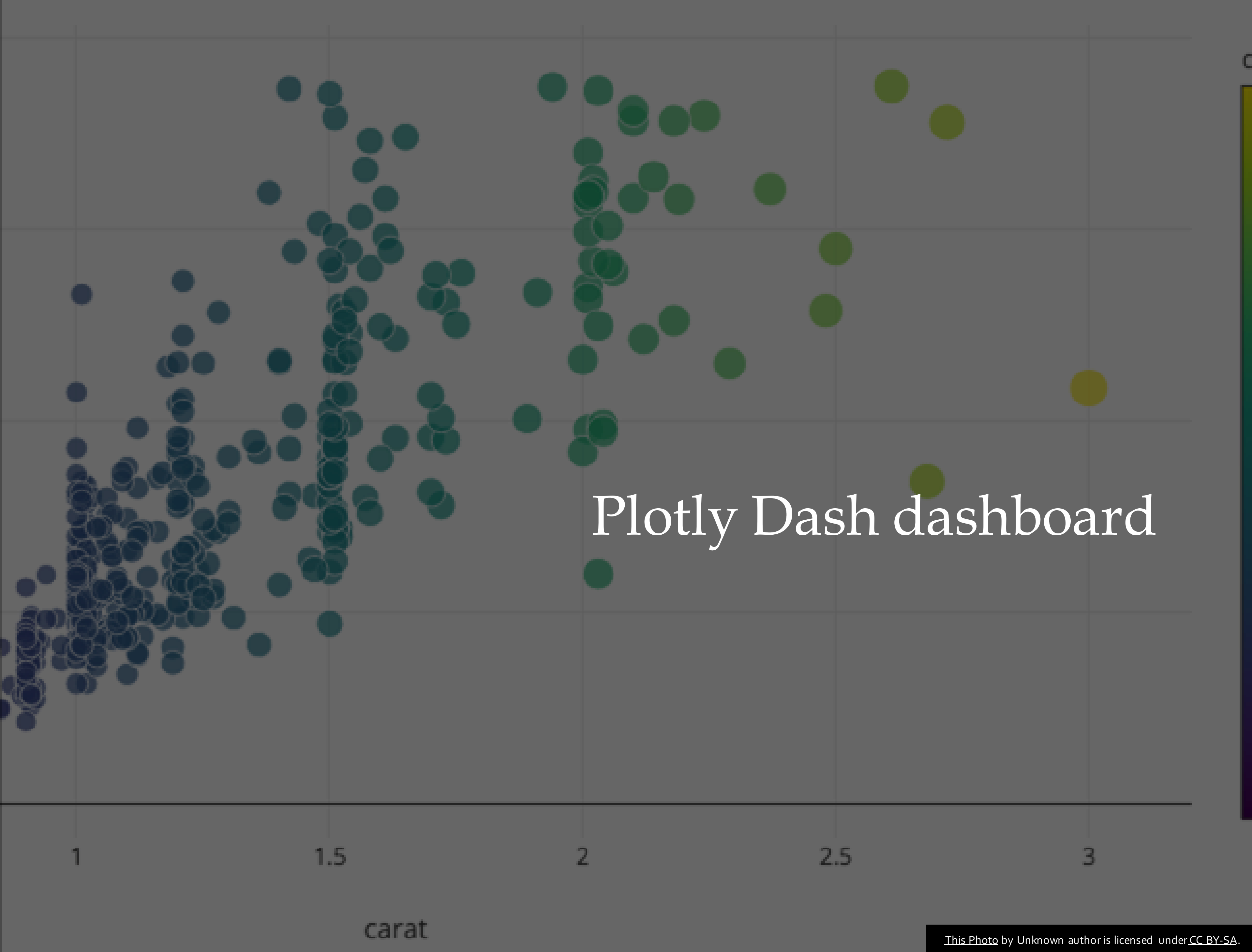


- Plotting our model decison regions

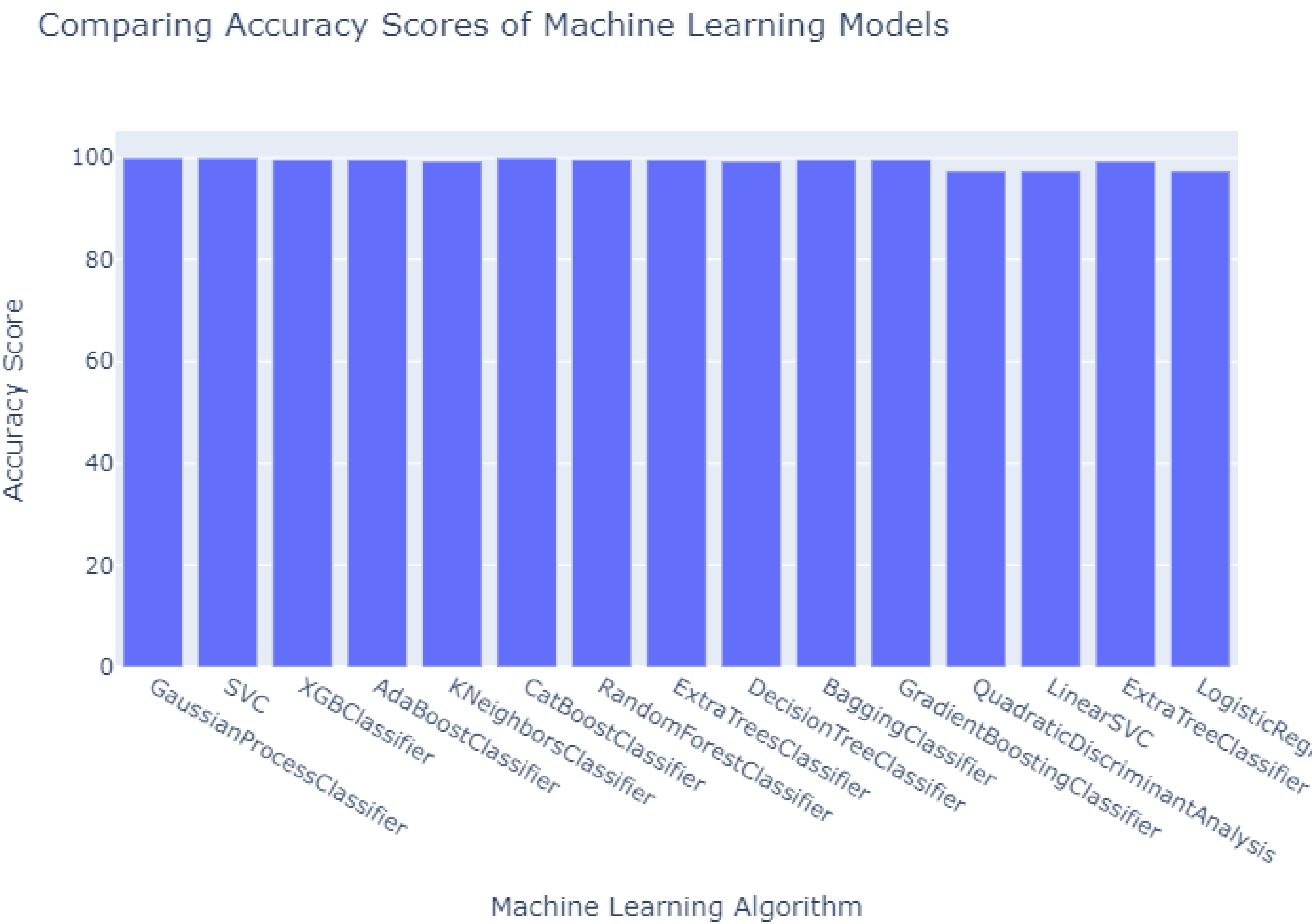


Comparing With Other Supervised algorithms

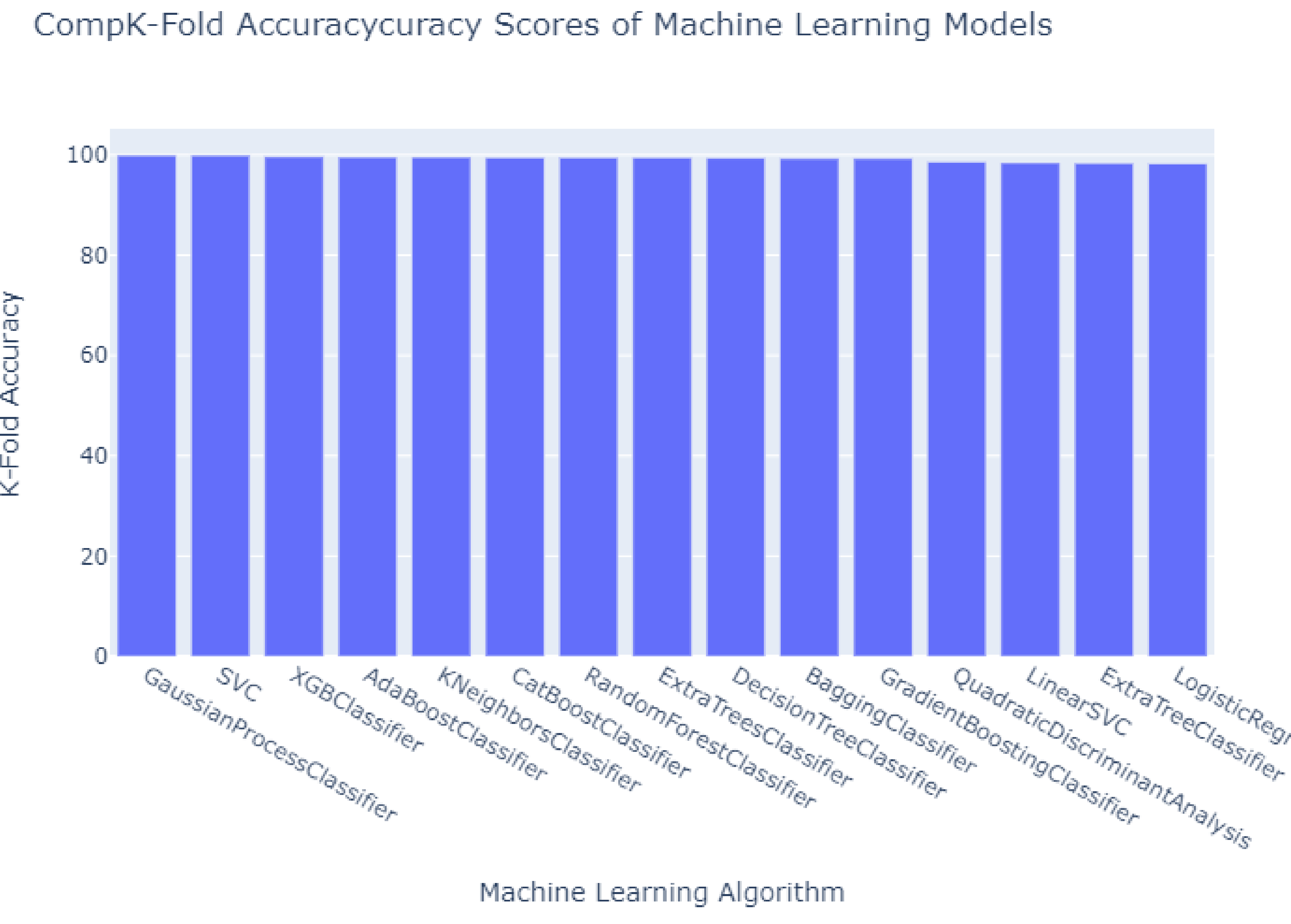
- Comparing Accuracy Scores of Machine Learning Models
- CompK-Fold Accuracycuracy Scores of Machine Learning Models



Plotly Dash dashboard

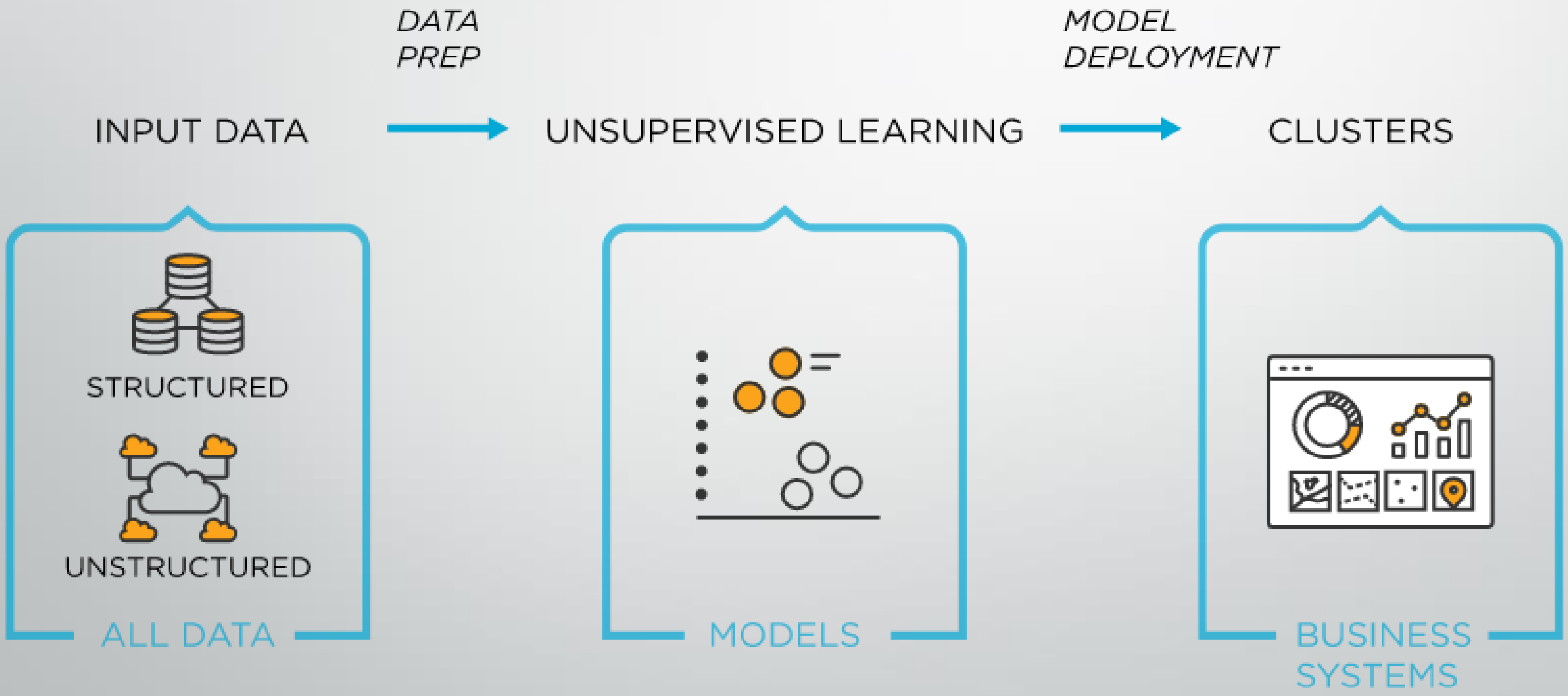


Comparing Accuracy Scores of Machine Learning Models
Accuracy Score
0
20
40
60
80
100
GaussianProcessClassifier
SVC
XGBClassifier
AdaBoostClassifier
KNeighborsClassifier
CatBoostClassifier
RandomForestClassifier
ExtraTreesClassifier
DecisionTreeClassifier
BaggingClassifier
GradientBoostingClassifier
QuadraticDiscriminantAnalysis
LinearSVC
ExtraTreeClassifier
Machine Learning Algorithm


CompK-Fold Accuracycuracy Scores of Machine Learning Models
K-Fold Accuracy
0
20
40
60
80
100
GaussianProcessClassifier
SVC
XGBClassifier
AdaBoostClassifier
KNeighborsClassifier
CatBoostClassifier
RandomForestClassifier
ExtraTreesClassifier
DecisionTreeClassifier
BaggingClassifier
GradientBoostingClassifier
QuadraticDiscriminantAnalysis
LinearSVC
ExtraTreeClassifier
Machine Learning Algorithm

	MLA Name	Accuracy Score	K-Fold Accuracy
0	GaussianProcessClassifier	100.000000	99.909091
1	SVC	100.000000	99.909091
2	XGBClassifier	99.636364	99.636364
3	AdaBoostClassifier	99.636364	99.545455
4	KNeighborsClassifier	99.272727	99.545455
5	CatBoostClassifier	100.000000	99.454545
6	RandomForestClassifier	99.636364	99.454545
7	ExtraTreesClassifier	99.636364	99.453711
8	DecisionTreeClassifier	99.272727	99.363636
9	BaggingClassifier	99.636364	99.272727
10	GradientBoostingClassifier	99.636364	99.271893
11	QuadraticDiscriminantAnalysis	97.454545	98.632193
12	LinearSVC	97.454545	98.450375
13	ExtraTreeClassifier	99.272727	98.360300
14	LogisticRegressionCV	97.454545	98.268557

5.2 Unsupervised Machine learning Algorithm



Unsupervised learning is a type of machine learning in which the algorithm is not provided with any pre-assigned labels or scores for the training data. As a result, unsupervised learning algorithms must first self-discover any naturally occurring patterns in that training data set.

➤ Using K-Means Clustering

Building a K-Means Clustering model to detect if a banknote is real or fake.

Step 1

- Gather and Assess the data.

Step 2

- Run K-Means.

Step 3

- Re-run K-means several times to to see if we get similar results, which can tell if the K-Means model is stable or not.

Step 4

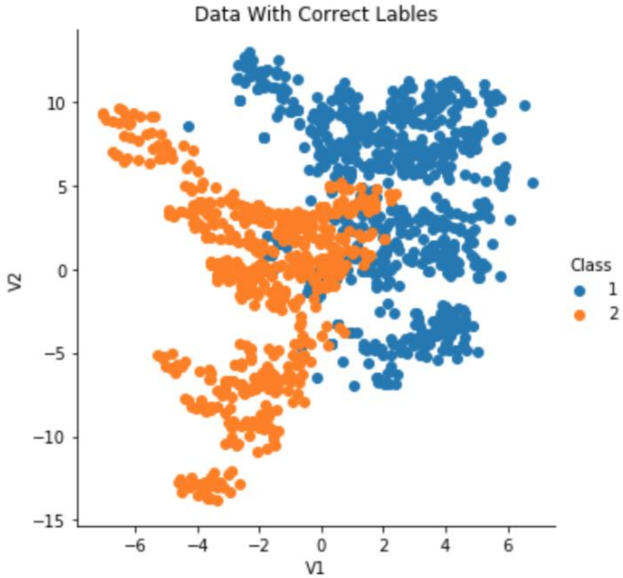
- Analyze the K-Means computing results

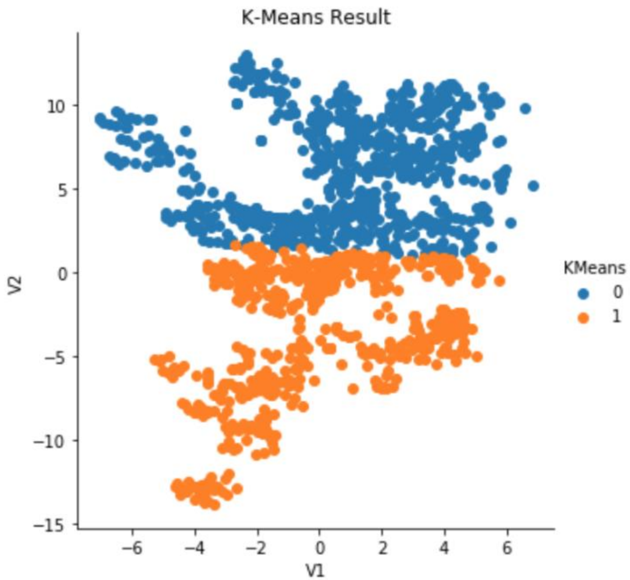
Step 5

- Calculate the accuracy of the result!

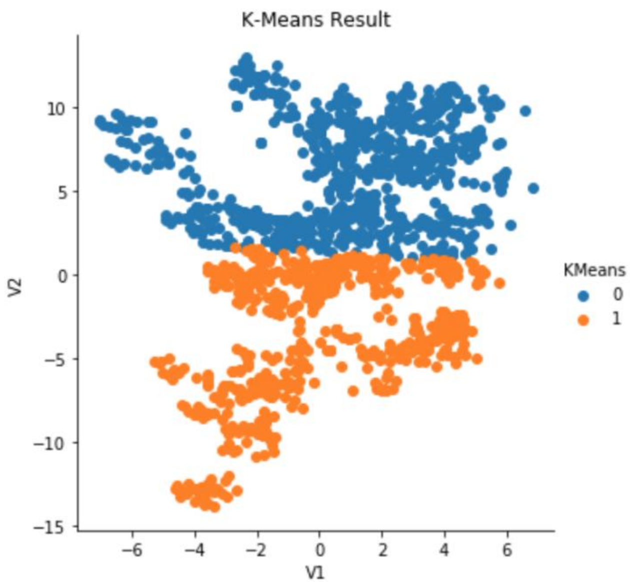
- The first step in building K-Means is to assess if this dataset is suitable for K-Means; if not, then we should choose other clustering models. After seeing this plot, I found the data distribution in the graph is neither too wide, nor too centered at one place, therefore it is worth trying to computing K-Means on this dataset. But, there is no obvious cluster in spherical shapes so we should expect the K-Means model won't work perfectly here.
- Third step is to run K-Means several times since K-Means will be randomly choosing initial places to be centroids, and then they will be changing their places according to the average distances from the members of each cluster, which will be set as the new centroids. Therefore, we normally will get different results every time we rerun K-Means, but if the results are too different among many tests, then it means that K-Means might not be suitable for this dataset, since it is not stable. Here, after running K-Means 9 times, the results we got are very similar, which means the K-Means here are stable.

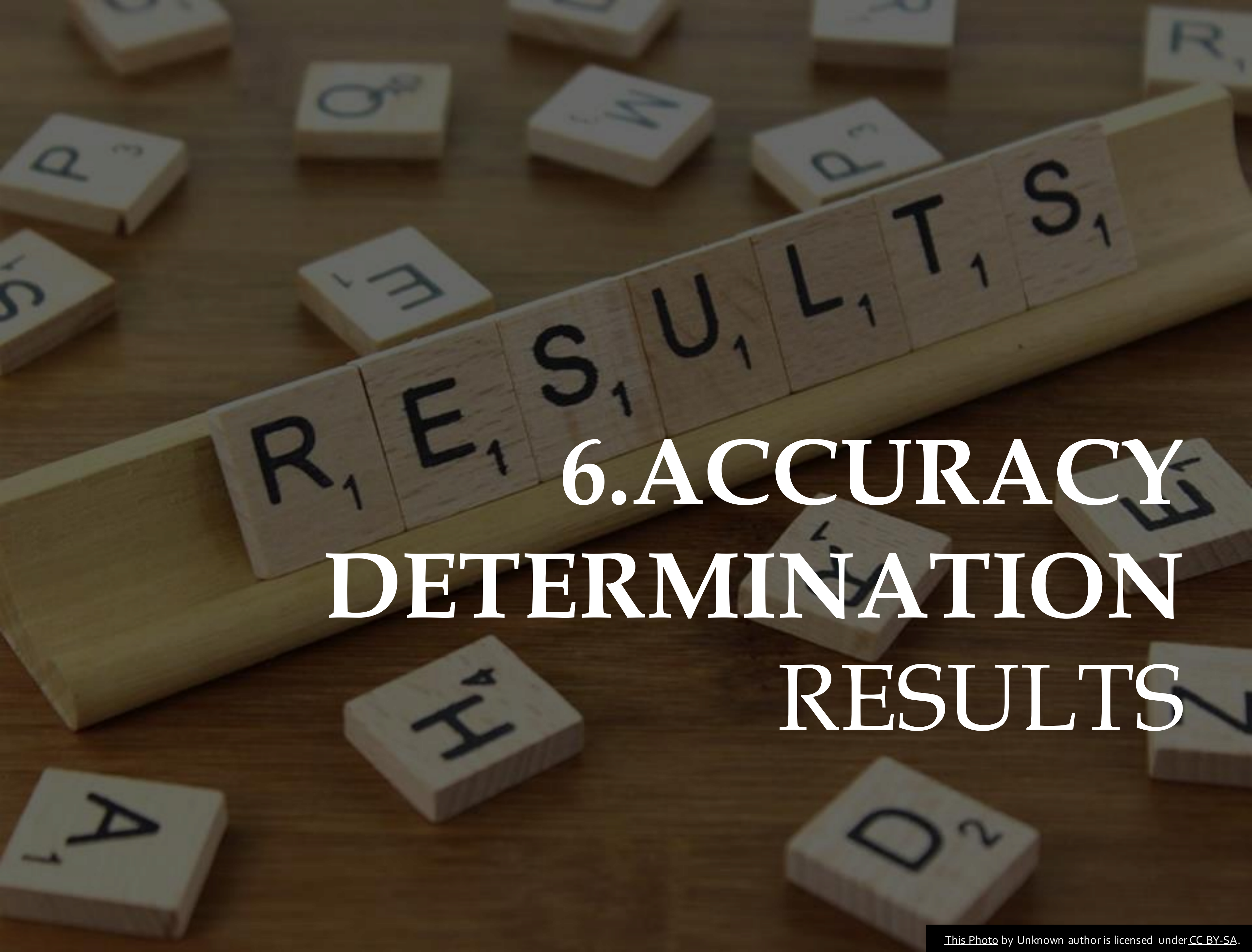
- We can see the K-Means one tends to be divided by a horizontal line at V2(skewness) = 1, whereas the original one tends to be divided by a slightly slant vertical line at V1(variance) =0. Which showed one drawback of K-Means which is that K-Means gives more weight to the bigger clusters.(The group 1 in K-Means tends to include the bigger cluster in the below position.) Let's calculate the accuracy of this K-Means clustering model:


Data With Correct Lables
V2
V1
10
5
0
−5
−10
−15
−6
−4
−2
0
2
4
6
Class
1
2


K-Means Result
V2
V1
10
5
0
−5
−10
−15
−6
−4
−2
0
2
4
6
KMeans
0
1


Data With Correct Lables
V2
V1
Class
1
2


K-Means Result
V2
V1
KMeans
0
1



6.ACCURACY DETERMINATION RESULTS

6.1.In logistic Regression Precision and Recall

Precision is the fraction of positive predictions that are correct. For instance, in our Banknote Authentication classifier, precision is the fraction of notes classified as authentic that are actually authentic.

Precision is given by the following ratio:

P = TP / (TP + FP)

Sometimes called sensitivity in medical domains, recall is the fraction of the truly positive instances that the classifier recognizes. A recall score of one indicates that the classifier did not make any false negative predictions. For our Banknote Authentication classifier, **recall is the fraction of authentic notes that were truly classified as authentic.**

Recall is calculated with the following ratio:
R = TP / (TP + FN)
Individually, precision and recall are seldom informative; they are both incomplete views of a classifier's performance.
Both precision and recall can fail to distinguish classifiers that perform well from certain types of classifiers that perform poorly. A trivial classifier could easily achieve a perfect recall score by predicting positive for every instance. For example, assume that a test set contains ten positive examples and ten negative examples.
A classifier that predicts positive for every example will achieve a recall of one, as follows:
R = 10 / (10 + 0) = 1
A classifier that predicts negative for every example, or that makes only false positive and true negative predictions, will achieve a recall score of zero. Similarly, a classifier that predicts that only a single instance is positive and happens to be correct will achieve perfect precision.
Scikit-learn provides a function to calculate the precision and recall for a classifier from a set of predictions and the corresponding set of trusted labels.

Calculating our Banknote Authentication classifier's precision and recall:

Our **classifier's precision is 0.988**; almost all of the notes that it predicted as authentic were actually authentic. Its recall is also high, indicating that it correctly classified approximately 98 percent of the authentic messages as authentic.

Calculating the F1 measure

The F1 measure is the harmonic mean, or weighted average, of the precision and recall scores. Also called the f-measure or the f-score, the F1 score is calculated using the following formula:

F1 = 2PR / (P + R)

The F1 measure penalizes classifiers with imbalanced precision and recall scores, like the trivial classifier that always predicts the positive class. A model with perfect precision and recall scores will achieve an F1 score of one. A model with a perfect precision score and a recall score of zero will achieve an F1 score of zero. As for precision and recall, scikit-learn provides a function to calculate the F1 score for a set of predictions. Let's compute our classifier's F1 score.

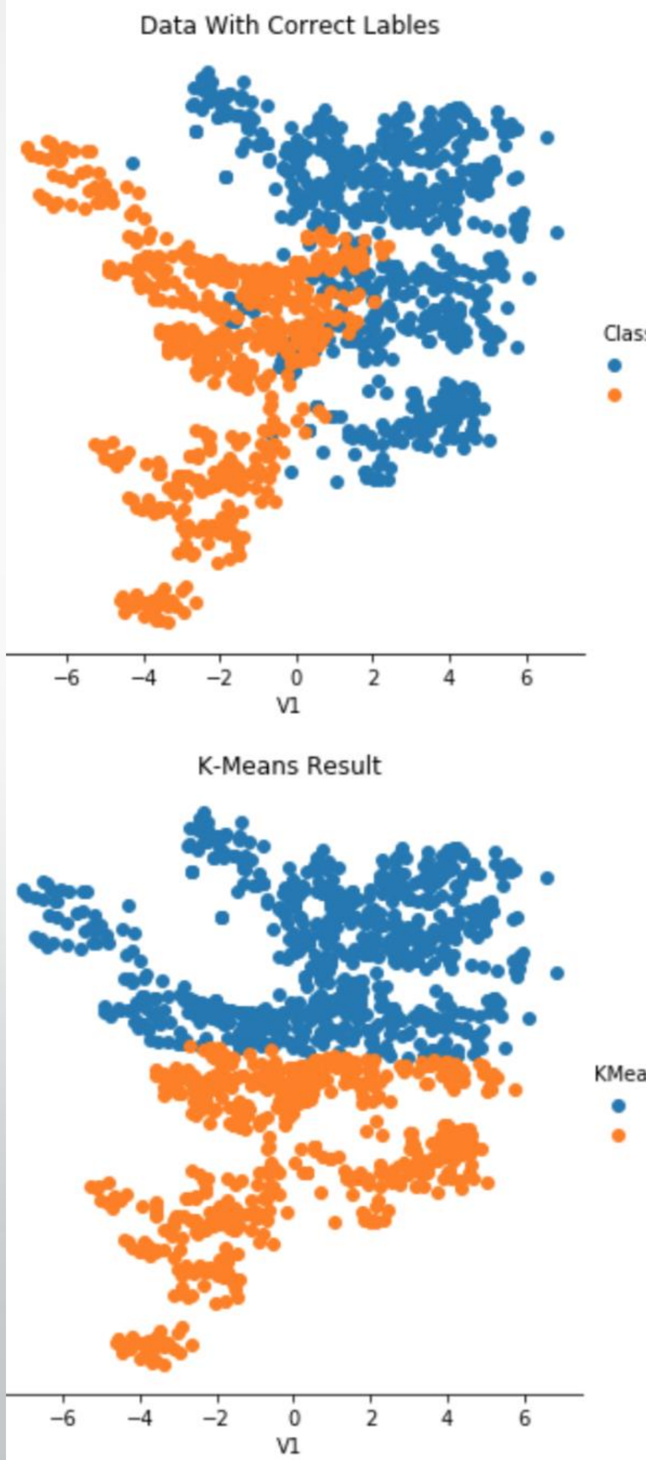
The arithmetic mean of our classifier's precision and recall scores is **0.98**. As the difference between the classifier's precision and recall is small, the F1 measure's penalty is small. Models are sometimes evaluated using the F0.5 and F2 scores, which favor precision over recall and recall over precision, respectively.

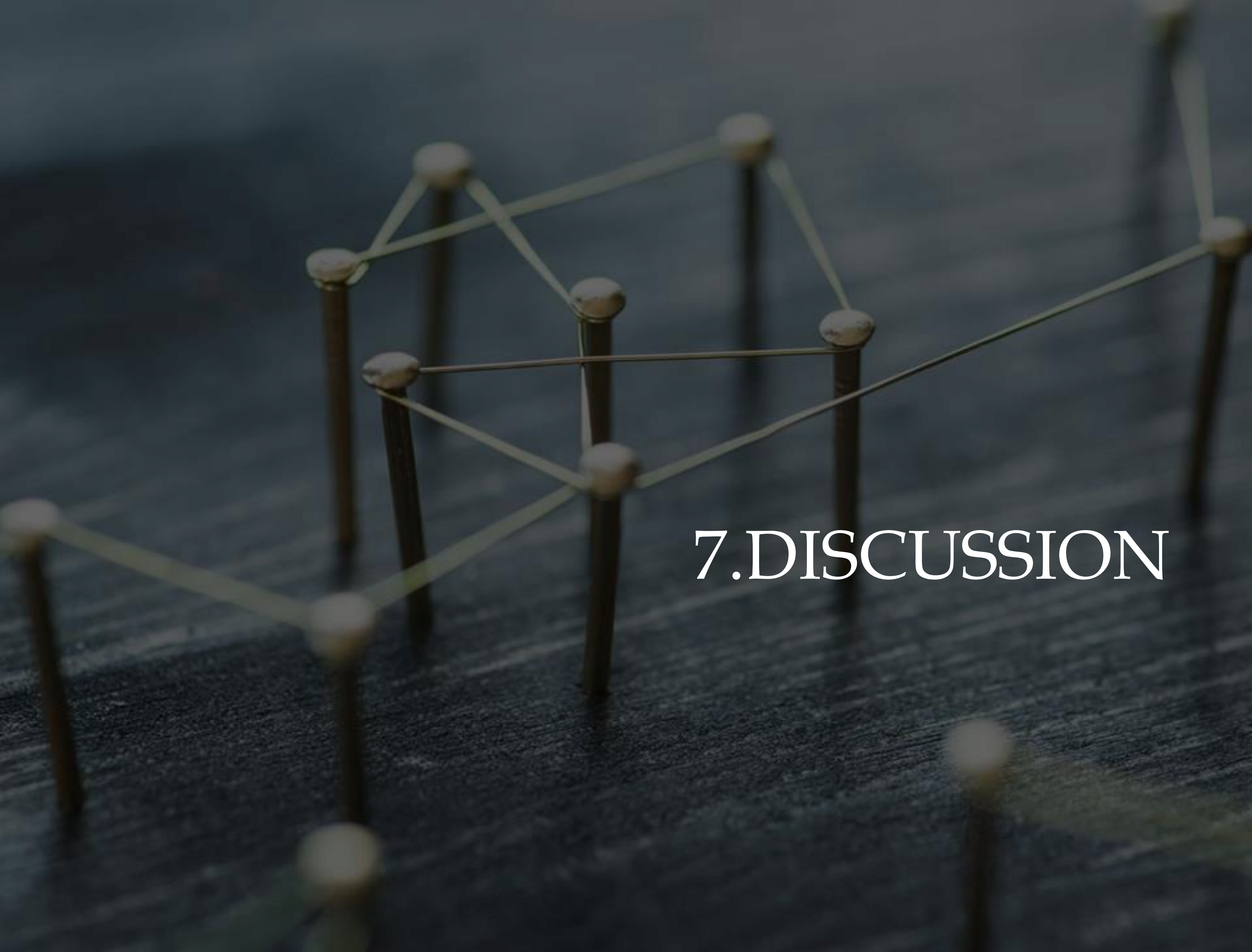
```
from sklearn.metrics import roc_auc_score, roc_curve, auc
roc_auc_score(y_test,lr.predict(X_test_std))

0.98922413793103448
```

6.2.Comparisons among accuracies among supervised classification Algorithms

	MLA Name	Accuracy Score	K-Fold Accurac
	GaussianProcessClassifier	100.000000	99.909091
	SVC	100.000000	99.909091
	XGBClassifier	99.636364	99.636364
	AdaBoostClassifier	99.636364	99.545455
	KNeighborsClassifier	99.272727	99.545455
	CatBoostClassifier	100.000000	99.454545
	RandomForestClassifier	99.636364	99.454545
	ExtraTreesClassifier	99.636364	99.453711
	DecisionTreeClassifier	99.272727	99.363636
	BaggingClassifier	99.636364	99.272727
0	GradientBoostingClassifier	99.636364	99.271893
1	QuadraticDiscriminantAnalysis	97.454545	98.632193
2	LinearSVC	97.454545	98.450375
3	ExtraTreeClassifier	99.272727	98.360300
4	LogisticRegressionCV	97.454545	98.268557

6.3.K-Means Clustering Accuracy





7.DISCUSSION

7.1.ANALYSIS AND DISCUSSION THROUGH ML TECHNIQUES

Classification of banknotes has been a neglected research area in comparison with banknote recognition and counterfeit banknote classification. However, with increasing penetration of automated self-service machines, such as ATMs, mechanical breakdown caused by unfit banknotes has become a serious maintenance and repair issue. Banknotes with high soiling levels also pose problems to banknote counters because they trigger false recognition and false rejection problems in banknote recognition, counterfeit banknote detection, and serial number recognition. Banks ensure circulation of only fit banknotes by withdrawing unfit banknotes from circulation by means of continuous fitness classification. In this context, fitness classification of banknotes has recently been attracting more attention. The following two issues should be considered in banknote fitness classification studies: - Most methods for fitness classification classify banknotes into two classes: fit and unfit banknotes. However, such a binary classification has the inherent problem of requiring subjective judgment without any clear-cut quantifiable criteria. Therefore, experts are usually involved to perform visual assessment of the soiling level of banknotes, or densitometers are used to distinguish fit and unfit banknotes depending on the measured values. - Besides the binary classification of fit and unfit banknotes, it is also important to ensure reproducibility of the assigned fitness level when the same banknote is put into a machine repeatedly.

8.CONCLUSION

One way of finding a good bias-variance tradeoff is to tune the complexity of the model via regularization. Regularization is a very useful method to handle collinearity (high correlation among features), filter out noise from data, and eventually prevent overfitting. The concept behind regularization is to introduce additional information (bias) to penalize extreme parameter weights. The most common form of regularization is the so-called L2 regularization (sometimes also called L2 shrinkage or weight decay).

The parameter C that is implemented for the LogisticRegression class in scikit-learn comes from a convention in support vector machines, which will be the topic of the next section. C is directly related to the regularization parameter, which is its inverse. Consequently, decreasing the value of the inverse regularization parameter C means that we are increasing the regularization strength.

9.ACKNOWLEDGEMENTS

- https://towardsdatascience.com/k-means-clustering-project-banknote-authentication-289cfe773873
- https://www.kaggle.com/discussion
- https://notebook.community/Aniruddha-Tapas/Applied-Machine-Learning/Classification/Logistic%20Regression%20for%20Banknote%20Authentication#Choosing-a-classification-algorithm