**Data Intensive Computing**

**Portuguese Banking Institution**

**Sudheer Anugoju , (**[**sudheera@buffalo.edu**](mailto:sudheera@buffalo.edu) **, UBIT #50442552),MS**

**Vishnu Sudhamsh Konkimalla([vkonima@buffalo.edu](mailto:vkonima@buffalo.edu) , UBIT #50441739), MS**

**Project Phase 1**

**Problem Statement:**

A Portuguese banking institution's direct marketing campaign is covered by the entries in this data collection. Calling was used to carry out the marketing effort. Prior to a client declining or accepting a term deposit membership, it frequently takes more than one call to reach them. The classification's objective is to forecast the client's decision to subscribe to the term deposit (yes/no) (variable y).

**Data Source:**

The data is related to the Bank marketing, which has 41188 rows and 20 columns.

The attributes in the data are,

* Input variables

1 - age (numeric)  
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')  
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)  
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')  
5 - default: has credit in default? (categorical: 'no','yes','unknown')  
6 - housing: has housing loan? (categorical: 'no','yes','unknown')  
7 - loan: has personal loan? (categorical: 'no','yes','unknown')

* Related with the last contact of the current campaign:  
  8 - contact: contact communication type (categorical: 'cellular','telephone')  
  9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', …, 'nov', 'dec')  
  10 - dayofweek: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
* Other Attributes:  
  11 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)  
  12 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)  
  13 - previous: number of contacts performed before this campaign and for this client (numeric)  
  14 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
* Social and Economic Context Attributes:  
  15 - emp.var.rate: employment variation rate - quarterly indicator (numeric)  
  16 - cons.price.idx: consumer price index - monthly indicator (numeric)  
  17 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)  
  18 - euribor3m: euribor 3 month rate - daily indicator (numeric)  
  19 - nr.employed: number of employees - quarterly indicator (numeric)
* Output variable (desired target):

20 - y - has the client subscribed a term deposit? (binary: 'yes','no')

A screenshot of a computer

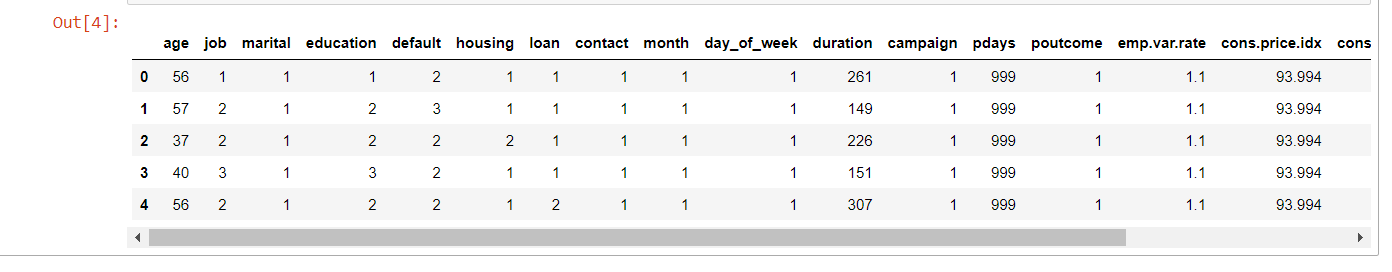
Description automatically generated with medium confidence

**Data Cleaning/Processing:**

The Data cleaning techniques we used to make the improper data to a meaningful data, by this the model performs better when training and testing.

1. **Labelling the Values to Numeric:**

Here we are changing the categorical data to numeric.



1. **Dropping Unnecessary Columns with Single Values:**

We removed the column ‘**pdays**’ which is having single values in the column.

Table

Description automatically generated

1. **Replace Null Values if there are any:**

Checking the null values in the data and replacing those with mode values of the respective column.

Graphical user interface

Description automatically generated with medium confidence

1. **Rescale dataset columns to the range from 0 to 1:**

We used Min Max normalization scaling to standardize the range of independent variables.

Graphical user interface, application

Description automatically generated

1. **Finding correlation to find the Dependencies and Remove over Headed Columns:**

Used the correlation function to check the Dependencies and remove the columns which are over headed. Dropped columns are [‘euribor3m’,’ nr.employed’,’ emp.var.rate’].

A screenshot of a computer

Description automatically generated with medium confidence

1. **Convert Normalized data's datatypes to float:**

Here converted the normalized data into float datatype which will be used for the machine learning algorithm efficiently.

**Graphical user interface

Description automatically generated**

1. **Renaming the Fields:**

Renaming the fields so that we can remove the unwanted characters in the column names.

**Graphical user interface, application

Description automatically generated**

1. **Remove rows whose age is less than 20:**

From the real world scenarios people who’s age less than 20 will not have any loan as they will not have any source of income to apply for a loan.

So removing the columns whose age is less than 20.

Table

Description automatically generated

1. **Removing records with Duplicate Data:**

Removed the duplicated data so which helps in the improve the data.

Graphical user interface, application

Description automatically generated

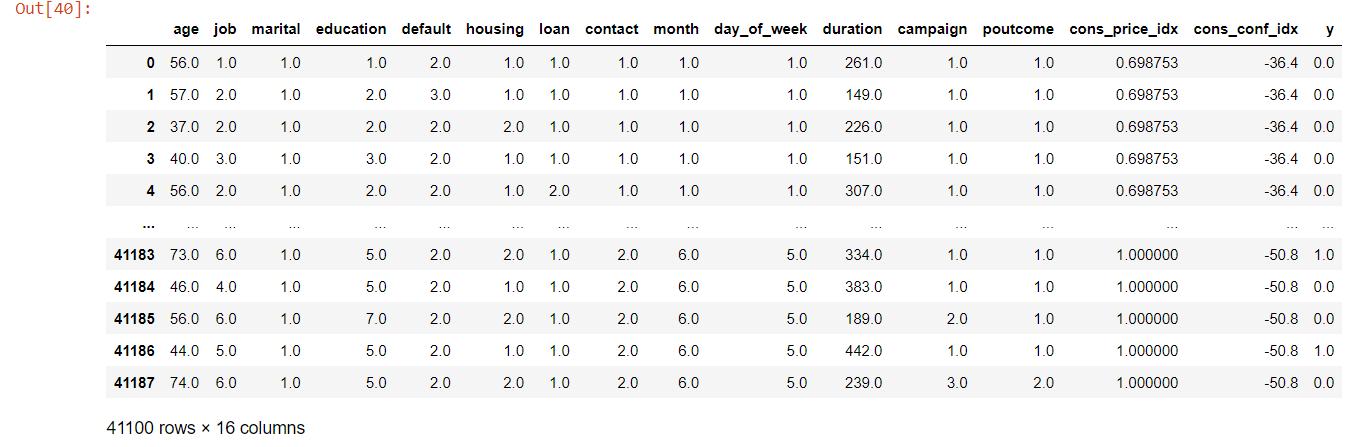
1. **Remove columns that has low variance.**

By removing the columns that has low variance will be more efficient to train the model.

**Graphical user interface

Description automatically generated with medium confidence**

After using the data cleaning methods, the rows and columns are



**Exploratory Data Analysis:**

Data analysis utilizing visual methods is called exploratory data analysis (EDA). With the use of statistical summaries and graphical representations, it is used to identify trends, patterns, or to verify assumptions.

1. **Plot on distribution on Education:**Result: the graph shows that the data has more candidates with University Degree followed by high-school

**Chart, histogram

Description automatically generated**

1. **Catplot:**

cons.price.idx: The Consumer Price Index (CPI) is a measure of the average change over time in the prices paid by urban consumers for a market basket of consumer goods.

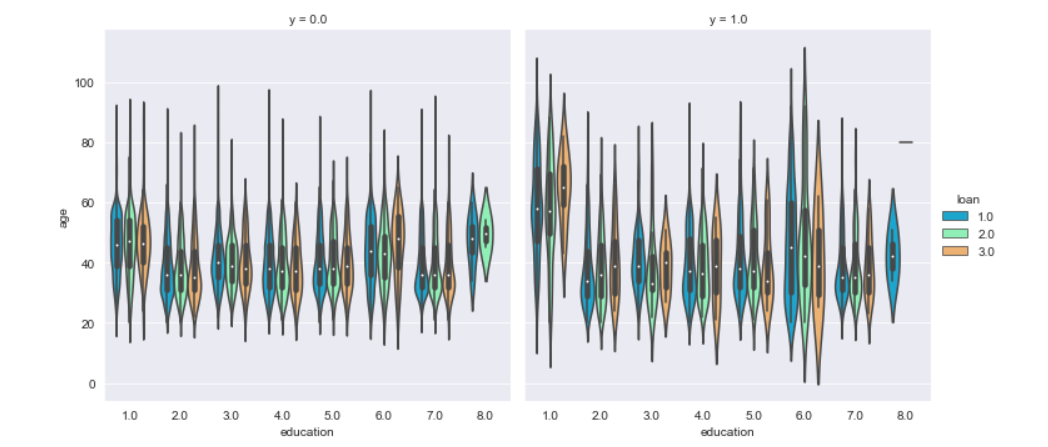
Poutcome: outcome of the previous marketing campaign (categorical:'failure','nonexistent','success')

Plot on how rise in CPI impacts outcome of the phone based direct marketing campaign outcomes

Chart, bar chart

Description automatically generated

Based on Age of an individual and their educational background, the plot shows whether if the contact has a personal loan.



1. **Violinplot:**

The following graph show cases the impact on output ‘Y’ based on the call duration on each campaign phone call.

Result: Looks like short duration calls have less chances of subscribing to the bank contract.

Chart

Description automatically generated

This consumer confidence indicator provides an indication of future developments of households' consumption and saving, based upon answers regarding their expected financial situation, their sentiment about the general economic situation, unemployment and capability of savings.

The score is relative to 1985, so anything above 100 means consumers feel more optimistic about the economy than they did in 1985 and anything below 100 means they feel less confident than in 1985.

The following graph shows how marital status impacts the customer sentiment about the economic systems.

['married', 'single', 'divorced', 'unknown'], [1,2,3,4]

df['cons.conf.idx'].unique() ranges from [-35 to -50]

Chart

Description automatically generated

1. **Joinplot:**

The following graph show cases the call duration on each campaign phone call based on individuals of different age groups is proportionate to their interest to know more details about the campaign to subscribe to a term deposit.

Result: Age group between 30 to 40 tend to spare more time on call understanding more details.

Chart, scatter chart

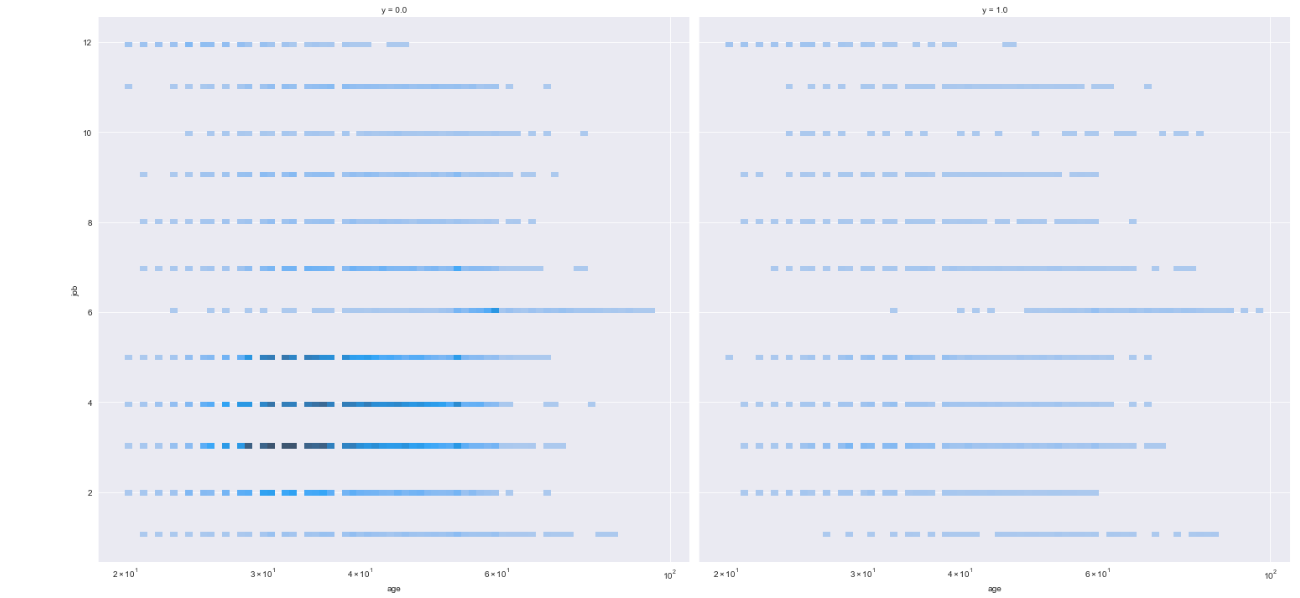
Description automatically generated

1. **Displot:**

The following graph shows the subsription status of the candidates according to the type of job at a specific age.

JOB: ['housemaid', 'services', 'admin.', 'blue-collar', 'technician', 'retired', 'management', 'unemployed', 'self-employed', 'unknown', 'entrepreneur', 'student'], [1,2,3,4,5,6,7,8,9,10,11,12]

Result: It looks like 'admin.', 'blue-collar', 'technician' job type candidates didn't show interest to subscribe and age group 40 to 60 being an entrepreneur, self-employed, retired opted IN.

****

1. **HeatMap:**

The following diagram shows the correlation of attributes in every possible combination with the remaining attributes.

**Chart, scatter chart

Description automatically generated**

1. **Subplot:**

Job Labels:

(['housemaid', 'services', 'admin.', 'blue-collar', 'technician','retired', 'management', 'unemployed', 'self-employed', 'unknown','entrepreneur', 'student'], [1,2,3,4,5,6,7,8,9,10,11,12])

Result: Majority of the data collected in the dataset are between age group 20 to 45 working as 'admin.', 'blue-collar', 'technician' or 'retired'

**Chart, histogram

Description automatically generated**

1. **Autocorrelation plot:**

In the below graph it is compared between duration, cons\_price\_idx & y, by this autocorrelation we can see the randomness of the data, where cons\_price\_idx has the more randomness than duration & y

Chart, line chart

Description automatically generated

1. **Boxplot:**

From the below graph we can say that how many number of cutomers took loan in which month

**Chart, box and whisker chart

Description automatically generated**

1. **Pivot plot:**

The pivot plot is shown between the age vs job where we can say what age group of people are having what type of job

Chart

Description automatically generated

**Project Phase 2**

**Models :**

1. SVM
2. Naive Bayes
3. Decision Tree calssifier
4. Random Forest
5. Logistic Regression

**Additional Model :**

1. K-Means

**SVM :**

There are a variety of different hyperplanes that might be used to split the two classes of data points. Finding a plane with the greatest margin—that is, the greatest separation between data points from both classes—is our goal. Maximizing the margin distance adds some support, increasing the confidence with which future data points can be categorized.

The hyperplane is essentially a line if there are just two input features. The hyperplane turns into a two-dimensional plane if there are three input features.

Since we have multi-dimensional features we used rbf.

Accuracy:

1. SVM: 90.3%

**Precision vs Recall graph for Naïve Bayes:**

**Chart, line chart

Description automatically generated**

**Confusion Matrix for SVM:**

Chart

Description automatically generated

**Naïve Bayes:**

Comparing SVM with naive bayes:

Accuracy:

1. SVM: 90.3%
2. Naive: 86.1%

Both are classifiers where Naive bayes is probabilistic in nature wherein SVM is geometric categorizing the data points maximizing the differences between them.

Naive assumes feature values are independent on other feature values, however SVM tries to build relations between the features.

For the dataset that's consider in the project, SVM has a sight more accuracy than Naive base. One of the assumptions is that more than one feature has correlation with the other features which means there is dependency among the features. This boosts SVM to classify data points more accurately. Naive based works on independent dependencies and features like duration, poutcome, education and contact etc., but these are quite low compared to internal dependencies among the features.

One advantage of Naive Bayes over SVM is training cost. Naive runs in 30 seconds but SVM takes almost 150 seconds running on GPU.

In conclusion, we can say that due to less correlation of features on output field and high inter-feature dependencies SVM outperformed Naive bayes.

**Precision vs Recall graph for Naïve Bayes:**

**Chart, line chart

Description automatically generated**

**Confusion Matrix for Naïve Bayes:**

**Chart

Description automatically generated**

**Decision Tree Classifier:**

(From the above two models) Now that we understood the colinearity is more than the correlation on output field and the dataset has categorical data in the features like job, education, marital status and campaign etc, we thought DecisionTrees can be helpful considering their functionality on independent variables, categorial data and multicolinearity. However, SVM outperformed Decision Trees by one percent. This might be because DecisionTrees are sensitive to noises in the data.

Comparing accuracies:

1. SVM: 90.3%
2. Naive: 86.1%
3. Decision tree: 89.39%

**Precision vs Recall graph for Decision Tree Classifier:**

**Chart, line chart

Description automatically generated**

**Confusion Matrix for Decision Tree Classifier:**

**Chart

Description automatically generated**

**Random Forest:**

Since decision trees are graphs that illustrates all possible outcomes and are prone to noises. To avoid this we can use random forests which outputs only a set of Decision Trees that work according to the output. We build a model on Random Forest algorithm which gave an accuracy of 2% more than decision tree.

This is so far highest accuracy we have achieved and unfortunately, we can't visualize Random Forest trees.

Comparing accuracies:

1. SVM: 90.3%
2. Naive: 86.1%
3. Decision tree: 89.39%
4. Random Forest: 90.9%

**Precision vs Recall graph for Random Forest:**

Chart, line chart

Description automatically generated

**Confusion Matrix for Random Forest:**

Chart

Description automatically generated

**Logistic Regression:**

Logistic regression is a classification model which is dedicated to binomial data predicting binary classes. This computes probability of occurrence of an event using logarithmic computations which gave an accuracy of 90.54%.

Comparing accuracies:

1. SVM: 90.3%
2. Naïve Bayes: 86.1%
3. Decision tree: 89.39%
4. Random Forest: 90.9%
5. Logistic Regression: 90.57%

**Precision vs Recall graph for Logistic Regression:**

**Chart, line chart

Description automatically generated**

**Confusion Matrix for Logistic Regression:**

**Chart

Description automatically generated**

**Precision vs Recall graphs:**

The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

In the below graph we can clearly see that precision and recall scores are equally high for Random Forest among all. Which leads to more accuracy since predicted true positive and true negatives are high.

Chart

Description automatically generated

**Additional Model:**

**K-Means:**

We classified the data into two and using the elbow technique we understood that number of clusters = 2 is ideal which is correct since we have binominal data.

**Chart, line chart

Description automatically generated**

Accuracy for the K-Means is 86.7%

**Performing balancing:**

We can clearly observe that the records with 'NO' label ratio is high compared to 'Yes' labeled  records. Hence, we need to balance data in such a way that our data has equal number of records with  'Yes' and 'No' labels.

We'll observe precesion score and recall as well as confusion matrix in the following cells on the balanced data.

Earlier, there are 36495 records with 'No' and 4605 records with 'Yes'. After balancing we have  equal distribution. 4605 records in each

Accuracy for the models after performing

1. SVM: 85.1%
2. Naïve Bayes: 77.92%
3. Decision tree Classifier: 72.92%
4. Random Forest: 86.86%
5. Logistic Regression: 79.3%

**Confusion Matrix for after balancing the data:**

**SVM:**

**A picture containing diagram

Description automatically generated**

**Naïve Bayes:**

A picture containing timeline

Description automatically generated

**Decision Tree Classifier:**

**A picture containing graphical user interface

Description automatically generated**

**Random Forest:**

**A picture containing graphical user interface

Description automatically generated**

**Logistic Regression:**

**A picture containing diagram

Description automatically generated**

**Precision vs Recall graphs after balancing the data:\**

**Chart

Description automatically generated**

**Project Phase 3**

**Selecting the Model:**

From the models we trained for the dataset, Random Forest Classifier is the best model with highest accuracy of more than 86% compared to other models we trained.

**b. How we have specifically used the models from Phase2?**

We have created a web application to the bank organizations where the employees can provide the inputs of all the campaign details (ex: call duration, age, education, income etc) in web form and are passed on to the above random forest model. We used Flask interface that integrates user interface .html with the model. Random Forest classifier predicts the output, scaling the input values from the interactive webpage provided by bank employee to predict customer’s chances of opting in for a campaign.

**c. For recommendations related to your problem statement based on your analysis.**

*what can users learn from your product? how does it help them solve problems related to your problem statement?*

Users can estimate the chances of a customer’s interest to opt-in to a bank campaign. And this helps to improve bank organizations to conduct campaigns in a more effective and efficient ways understanding the correlation of features on the result. Also, if a customer was predicted to opt-out from a campaign, there can be an extra effort put make him/her subscribe.

*other ideas for how to extend your project, or other avenues that could be explored related to the problem?*

There can be many improvements that can be done. For example, while customer’s are applying for a loan their possibility of sanctioning a loan should be displayed at end-user’s perspective implementing a regression model. This helps bank marketing in conducting campaigns.

**Webpage:**

For the web application we used Flask framework to connect our model and HTML page.

Webpage Contains:

1. Input fields
2. Predicted value
3. Visualization graph.

**Visualization:**

User get to visualize a graph where the ‘age vs duration’ is plotted and can compare the graph the actual data and can give the others inputs to get the better result.

**Input:**

In the webpage we have input fields like age, married, duration, education and so on. Where all the fields are mandatory to be given in the input to get the output value.

Graphical user interface, text

Description automatically generated

**Output:**

When the user click on the predict button it will redirect to the other html page and get the to know whether a person can optin for the campaign or not and also can see the visualization where point between age vs duration plot compared to the data.

Chart, scatter chart

Description automatically generated

**Conclusion:**

RandomForest helped us understand how given features associated to a customer impact his/her perspective of subscribing to a banking campaign/policy. Here, we displayed our prediction (yes/no) and plotted two graphs on age over duration (features which has high correlation on output) to understand where our point lies compared to all other data points in the dataset to improve. If in the picture we can see person with age 56 spent less than 200 seconds had chances of opting out, looking at the jointplot on the right we can see if duration spent was more or near to 1000 seconds the chances of opting in are high, employee can work on it.

**Contributions:**

|  |  |  |
| --- | --- | --- |
| **Parts** | **sudheera** | **vkonkima** |
| Project Phase 1 | 50% | 50% |
| Project Phase 2 | 50% | 50% |
| Project Phase 3 | 50% | 50% |

**References:**

1. [**https://www.kaggle.com/datasets/ruthgn/bank-marketing-data-set**](https://www.kaggle.com/datasets/ruthgn/bank-marketing-data-set)
2. [**https://www.digitalocean.com/community/tutorials/exploratory-data-analysis-python**](https://www.digitalocean.com/community/tutorials/exploratory-data-analysis-python)
3. [**https://realpython.com/python-data-cleaning-numpy-pandas/**](https://realpython.com/python-data-cleaning-numpy-pandas/)
4. [**https://seaborn.pydata.org/generated/seaborn.boxplot.html**](https://seaborn.pydata.org/generated/seaborn.boxplot.html)
5. [**https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm**](https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm)
6. **kmeans:**[**https://realpython.com/k-means-clustering-python/**](https://realpython.com/k-means-clustering-python/)
7. **Decisiontrees:**[**http://scikit-learn.org/stable/modules/tree.html**](http://scikit-learn.org/stable/modules/tree.html)
8. **Precision and Recall:**[**https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_precision\_recall.html**](https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html)
9. **SVM:**[**https://www.kaggle.com/code/prashant111/svm-classifier-tutorial/notebook**](https://www.kaggle.com/code/prashant111/svm-classifier-tutorial/notebook)
10. **Naive:**[**https://ieeexplore.ieee.org/abstract/document/5360338?casa\_token=yfdCz2kU1j0AAAAA:xY32jlPmP7jBsiZeHem25F9r8GBz5UjW0VOaVoRQA3iA9yw0Y617LlqCMOHfcLqG34lp8ESDJQ**](https://ieeexplore.ieee.org/abstract/document/5360338?casa_token=yfdCz2kU1j0AAAAA:xY32jlPmP7jBsiZeHem25F9r8GBz5UjW0VOaVoRQA3iA9yw0Y617LlqCMOHfcLqG34lp8ESDJQ)
11. **RandomForest:**[**https://www.tandfonline.com/doi/full/10.1080/21642583.2014.956265**](https://www.tandfonline.com/doi/full/10.1080/21642583.2014.956265)