Credit Card Default Prediction

Divakar Kumar
Data Science Trainees,
AlmaBetter, Bangalore

Ashish Gupta
Data Science Trainees,
AlmaBetter, Bangalore

Abstract: Predicting potential credit default accounts ahead of time is testing. Conventional factual strategies normally can't deal with a lot of information and the unique idea of misrepresentation and people. To handle this issue, late exploration has zeroed in on counterfeit and computational knowledge-based approaches. In this work, we introduce and approve a heuristic way to deal with mine potential default accounts ahead of time where a hazard likelihood is precomputed from all past information and the danger likelihood for late exchanges are figured as soon, they occur. Besides our heuristic methodology, we additionally apply an as of late proposed AI approach that has not been applied already on our designated dataset. Accordingly, we view as that these applied methodologies beat existing best in class approaches.

Keywords: *Default, bankruptcy EDA*

1.PROBLEM STATEMENT

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Description of columns from dataset documentation:

There are 25 variables:

• ID: ID of each client.

• LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit.

• SEX: Gender (1=male, 2=female)

• EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown).

• MARRIAGE: Marital status (1=married, 2=single, 3=others).

• AGE: Age in years.

• PAY_0: Repayment status in September 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)

• PAY_2: Repayment status in August 2005 (scale same as above)

PAY_3: Repayment status in July 2005 (scale same as above)

• PAY_4: Repayment status in June 2005 (scale same as above)

• PAY_5: Repayment status in May 2005 (scale same as above)

• PAY_6: Repayment status in April 2005 (scale same as above)

• BILL_AMT1: Amount of bill statement in September 2005 (NT dollar)

• BILL_AMT2: Amount of bill statement in August 2005 (NT dollar)

• BILL_AMT3: Amount of bill statement in July 2005 (NT dollar)

• BILL_AMT4: Amount of bill statement in June 2005 (NT dollar)

• BILL_AMT5: Amount of bill statement in May 2005 (NT dollar)

• BILL_AMT6: Amount of bill statement in April 2005 (NT dollar)

• PAY_AMT1: Amount of previous payment in September 2005 (NT dollar)

• PAY_AMT2: Amount of previous payment in August 2005 (NT dollar)

• PAY_AMT3: Amount of previous payment in July 2005 (NT dollar)

• PAY_AMT4: Amount of previous payment in June 2005 (NT dollar)

• PAY_AMT5: Amount of previous payment in May 2005 (NT dollar)

• PAY_AMT6: Amount of previous payment in April 2005 (NT dollar)

• default payment next month: Default payment (1=yes, 0=no)

2. INTRODUCTION

In general, we can refer to a customer's inability to pay, or their default on a payment, or personal bankruptcy, all as potential issues of non-payment. However, each of these scenarios is a result of different circumstances. Sometimes it is due to a sudden change in a person's income source due to job loss, health issues, or an inability to work. Sometimes it is a deliberate, for instance, when the customer knows that he/she is not solvent enough to use a credit card anymore, but still uses it until the card is stopped by the bank. In the latter case, it is a type of fraud, which is very difficult to predict, and a big issue to creditors.

To address this issue, credit card companies try to predict potential default, or assess the risk probability, on a payment in advance. From the creditor's side, the earlier the potential default accounts are detected the lower the losses. For this reason, an effective approach for predicting a potential default account in advance is crucial for the creditors if they want to take preventive actions. In addition, they could also investigate and help the customer by providing necessary suggestions to avoid bankruptcy and minimize the loss.

Analyzing millions of transactions and making a prediction based on that is time consuming, resource intensive, and some time error prone due to the dynamic variables (e.g., balance limit, income, credit score, economic conditions, etc.). Thus, there is a need for optimal approaches that can deal with the above constraints. In our previous work, we proposed an approach that precomputes all previous data (offline data) and calculates a score. Subsequently, it waits for a new transaction (online data) to occur and calculate another score as soon as the transaction occurs. Finally, all scores are combined to make a decision. We used the term OLAP data for offline data and OLTP data for online data in our previous work. The main limitations of the previous work were the use of a synthetic dataset and a lack of validation of the proposed model using a publicly available, realworld dataset. Online Analytical Processing (OLAP) systems typically use archived historical data over several years from a data warehouse to gather business intelligence for decisionmaking and forecasting. On the other hand, Online Transaction Processing (OLTP) systems, only analyze records within a short window of recent activities - enough to successfully meet the requirement of current transactions within a reasonable response time.

Currently, a variety of Machine Learning approaches are used to detect fraud and predict payment defaults. Some of the more common techniques include K Nearest Neighbor, Support Vector Machine, Random Forest, Artificial Immune System, Meta-Learning, Ontology Graph, Genetic Algorithms, and Ensemble approaches. However, a potential approach that has not been used frequently in this area is Extremely Random Trees, or Extremely Randomized Trees (ET). This approach came about in 2006 and is a tree-based ensemble method for supervised classification and regression problems. In Extremely Random Trees (ET) randomness goes further than the randomness in Random Forest. In Random Forest, the splitting attribute is determined by some criteria where the attribute is the best to split on that level, whereas in ET the splitting attribute is also chosen in an extremely random manner in terms of both variable index and splitting value. In the extreme case, this algorithm randomly picks a single attribute and cut point at each node, which leads to a totally randomized trees whose structures are independent of the target variables values in the learning sample. Moreover, in ET, the whole training set is applied to train the tree instead of using bagging to produce the training set as in Random Forest. As a result, ET gives a better result than Random Forest for a particular set of problems. Besides accuracy, the main strength of the ET algorithm is its computational efficiency and robustness. While ET does reduce the variance at the expense of an increase in bias, we will use this algorithm as the foundation for our proposed approach.

3. STEPS INVOLVED:

- **Exploratory Data Analysis**

  After importing library need to load the dataset the Pre-Processing was performed on Colab Notebook. The CSV file was loaded using pd.read_csv () function.

- **Null values Treatment**

Our dataset not containing any null value. And checking the null data is checked using isnull() function of python. Additionally, which sum() function was invoked to attain the index numbers of the missing values in the dataset. The output depicted that there was no missing values in our dataset.

Using Visual we can able see the how the visual looking when Null value not present

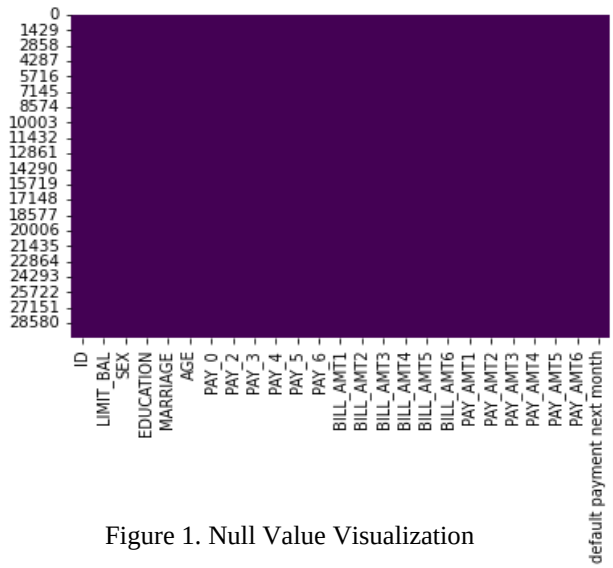


Figure 1. Null Value Visualization

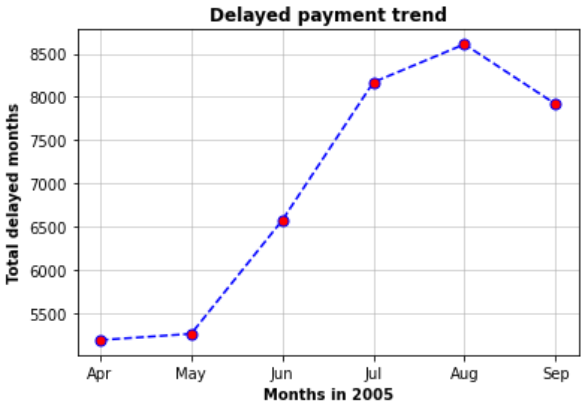


Figure 2. Delayed Payment Trend

Initially, from each record (customer) in the Taiwan dataset, we created 5 online transactions of type "pay" (payment) from PAY_AMT1 to PAY_AMT5. There was a huge jump from May,2005 (PAY_5) to July 2005 (PAY_3) when delayed payment increased significantly, then it peaked at August, 2005 (PAY_2), things started to get better in September, 2005 (PAY_1).

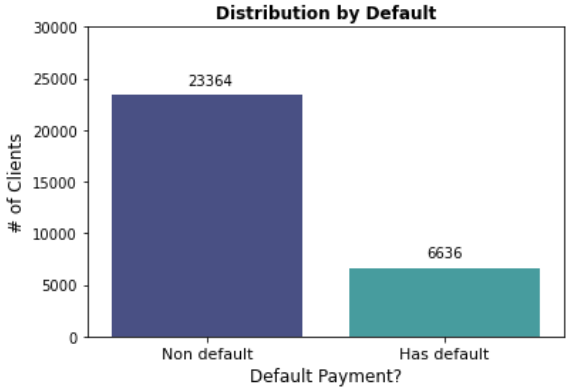


Figure 3. Distribution by Default

From the above bar chat, we can observe that all the almost types of distribution Non-default number is high and Has default Number is less.

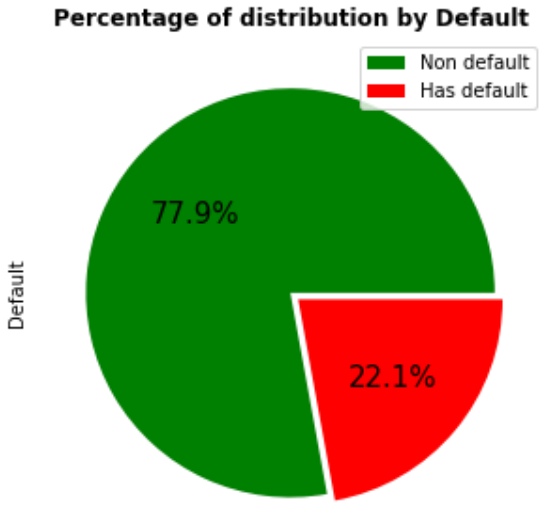


Figure 4. % Of distribution by default

From the above Pai chart, we can see that Non default is 77.9% and Has default is 22.1%.

And more point want add Here we can see that the data is imbalanced with 77.88% Non-default vs. 22.12% Has default.

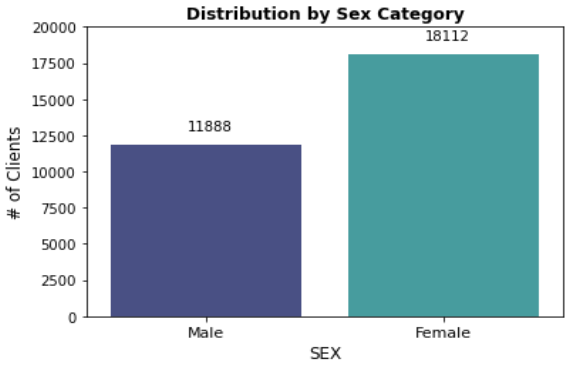


Figure 5. Male vs Female

From the above Bar chart, we can see that Number of female clients is more compared to Male clients.

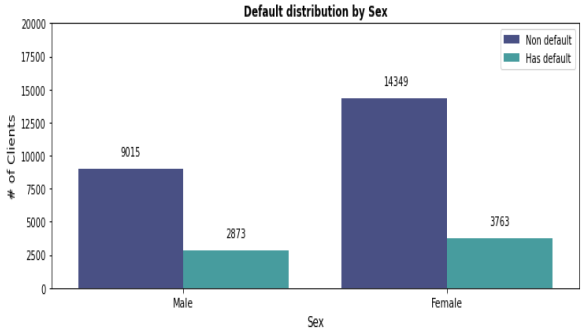


Figure 6. Default Distribution by Sex

From the above Bar chat, we can see that Number of Male Non default is 9015, Has default is 2873.And Number of Female Non default is 14349, Has default is 3763.

So Female client Has default number is more.

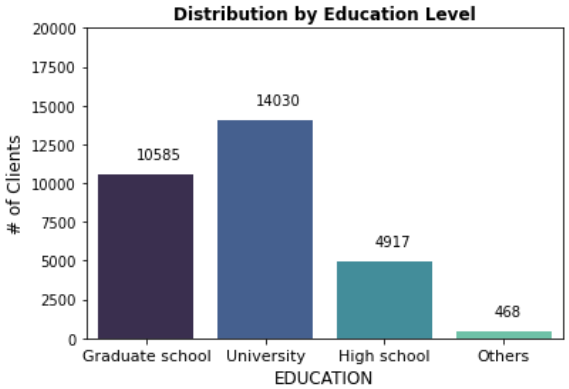


Figure 7. Distribution by Education Level

From the above Bar chart, we can see that those who are studied in university they are holding more number of credit card and Next Graduate school education level are holding card.

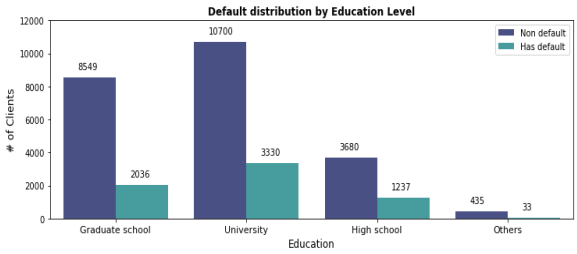


Figure 8. Default Distribution by Education Level

From the above Bar chart, we can see that high number of universities studied client found Has default and next as usual Graduate school university studied client found Has default.

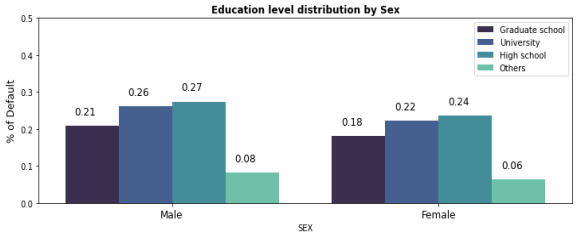


Figure 9. % of default client

From the above Bar chart, we can see that percentage of default client and it will divide with their sex with there are education level.

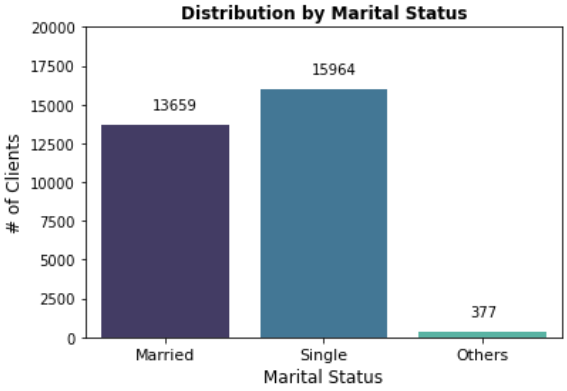


Figure 10. Distribution by Material Status

From the above Bar chart, we can see that high number of Single clients are holding credit card. And second highest is Married clients.

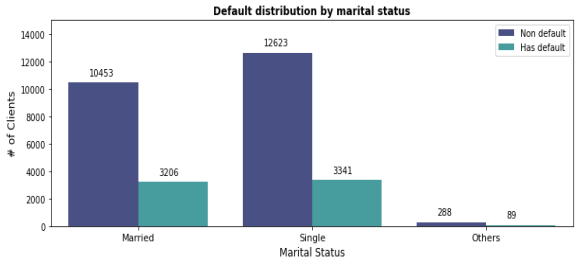


Figure 11. Default distribution by material status

From the above Bar chart, we can see that the more no of has default client are single and its number 3341 and Married client also holding credit card and It's has default number is 3206.

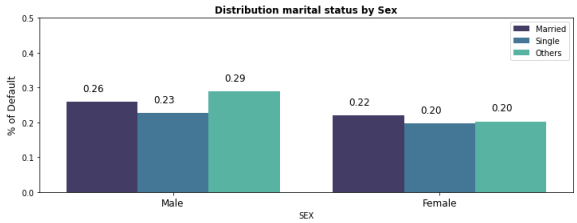


Figure 12. Distribution Marital Status by Sex

From the above Bar chart, we can see that percentage of default clients Distribution Marital Status by Sex.

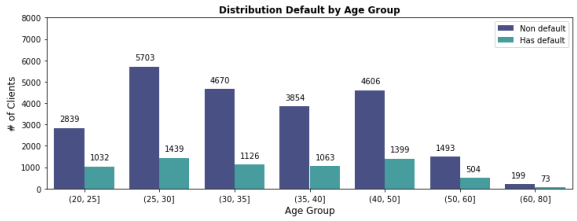


Figure 13. Distribution default by Age Group

From the above Bar chart, we can notice highest number of has default client age between 25 to 30.

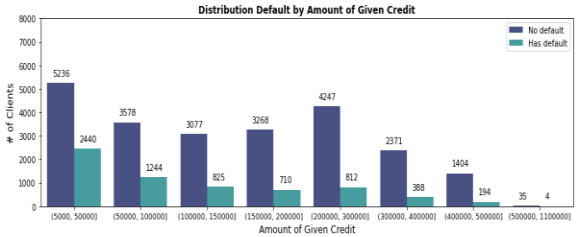


Figure 14. Distribution default by Amount of Given Credit

From the above Bar chart, we can see that those who having credit limit between 5000 to 50000 in this limit range of clients more number of Has default.

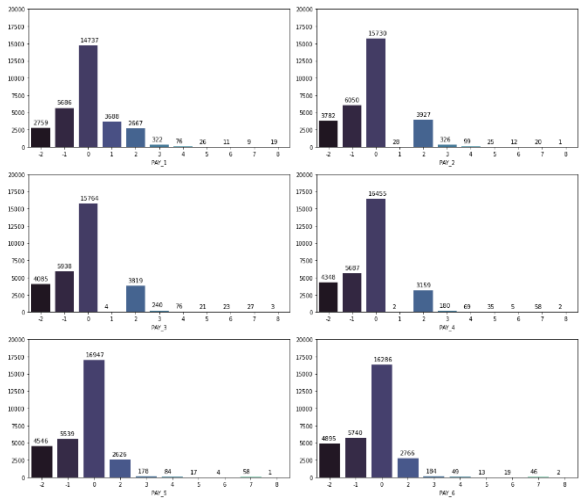


Figure 15. Repayment status for each month

From the above Bar chart, we can observe that repayment status of April, May, June, July, August and September 2005.

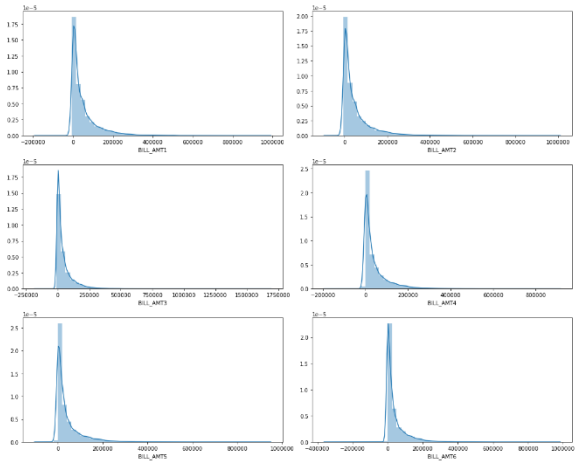


Figure 16. Amount of Bill Statement Each Month

From the above Bar chart, we can observe that BILL_AMT1 to BILL_AMT5. Since BILL_AMT is the sum of all individual bills or transactions, we divided this BILL_AMT into individual transactions by following the data distribution of a real credit card transactions datas

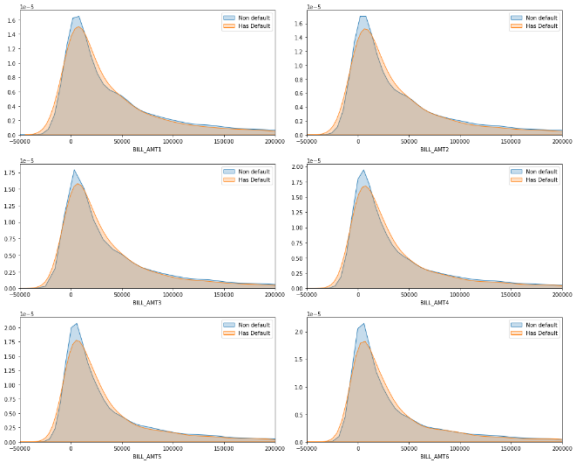


Figure 18. Amount of Bill Statement Each Month default distribution

From the above plot, we can observe that default distribution.

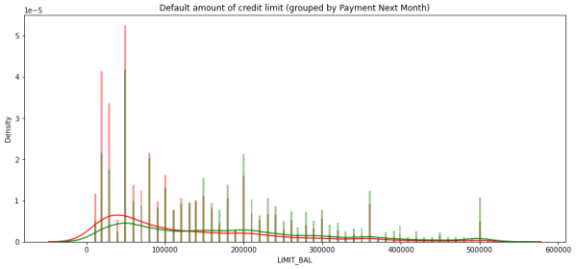


Figure 19. Default amount of credit limit (Group by payment next Month)

From the above Bar chart, we can observe that Most of defaults are for credit limits 0-1, 00,000. Larger defaults numbers are for the amounts of 50,000, 20,000 and 30,000.

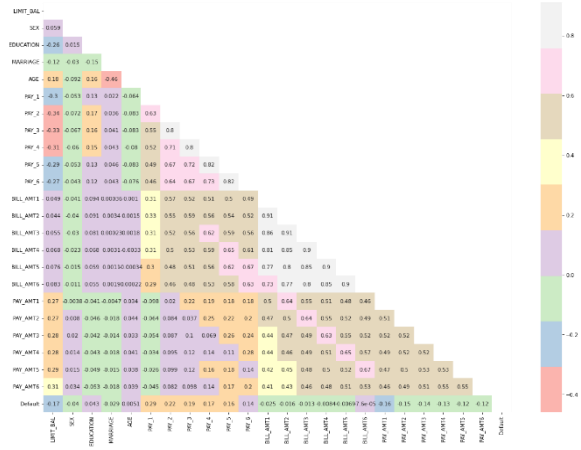


Figure 20. Correlation matrix of the dataset variables

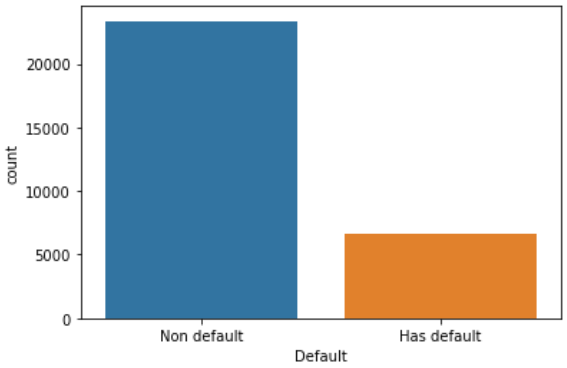


Figure 21. Non – default Vs Has Default

Here we can see that the data is imbalanced with 77.88% non-default vs. 22.12% Has default.

Fitting different models

For modelling we tried various classification algorithms like:

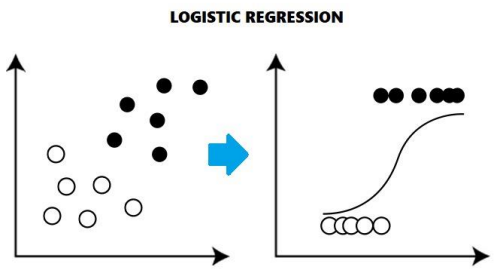
1. **Logistic Regression**
2. **Stochastic Gradient Descent**
3. **Random Forest classifier**
4. **K-Nearest Neighbors (KNN)**
5. **Supporting Vector Machine (SVM)**

Algorithms:

We used Different Model for Performance Checking purpose:

1. Logistic Regression:

Logistic Regression is a mathematical model used in statistics to estimate (guess) the probability of an event occurring using some previous data. Logistic Regression works with binary data, where either the event happens (1) or the event does not happen (0).



2. Stochastic Gradient Descent:

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

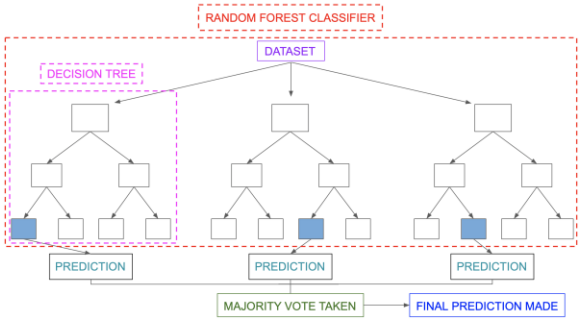
SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features. Strictly speaking, SGD is merely an optimization technique and does not correspond to a specific family of machine learning models. It is only a *way* to train a model. Often, an instance of **SGD Classifier** or **SGD Regressor** will have an equivalent estimator in the scikit-learn API, potentially using a different optimization technique. For example, using SGD Classifier(loss='log') results in logistic regression, i.e. a model equivalent to **Logistic Regression** which is fitted via SGD instead of being fitted by one of the other solvers in **Logistic Regression**. Similarly, SGDRegressor(loss='squared_error', penalty='l2') and **Ridge** solve the same optimization problem, via different means.

3. Random Forest classifier

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.



4. K-Nearest Neighbors:

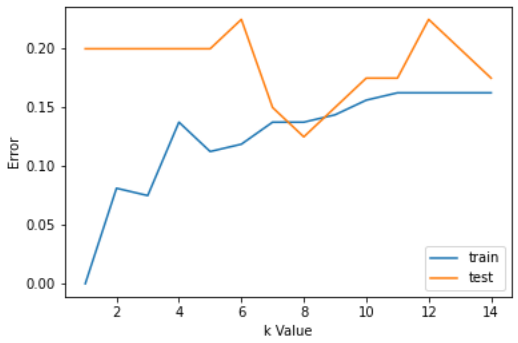
K Nearest Neighbor algorithm falls under the Supervised Learning category and is used for classification (most commonly) and regression. It is a versatile algorithm also used for imputing missing values and resampling datasets. As the name (K Nearest Neighbor) suggests it considers K Nearest Neighbors (Data points) to predict the class or continuous value for the new Datapoint.

The algorithm's learning is:

1. Instance-based learning: Here we do not learn weights from training data to predict output (as in model-based algorithms) but use entire training instances to predict output for unseen data.

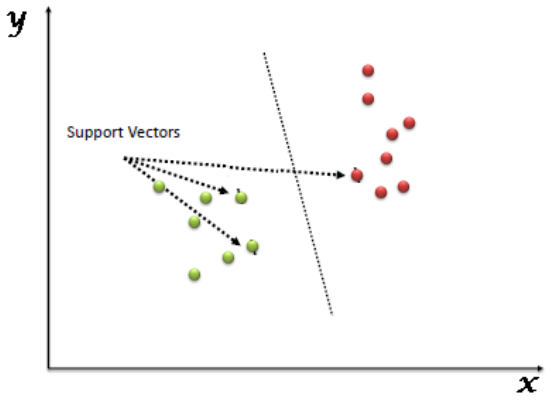
2. Lazy Learning: Model is not learned using training data prior and the learning process is postponed to a time when prediction is requested on the new instance.

3. Non Parametric: In KNN, there is no predefined form of the mapping function.



5. Supporting Vector Machine (SVM):

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line).

4.Conclusion:

That's it! We reached the end of our exercise.

Starting with loading the data so far, we have done EDA, null values treatment, In the EDA I was trying to show the best way the visuals and answering all questions need to show.

1) Using a Logistic Regression classifier, we can predict with 60.33% accuracy, whether a customer is likely to default next month.
2) Using a Stochastic Gradient Descent classifier, we can predict with 53.07% accuracy, whether a customer is likely to default next month.
3) Using a Random Forest classifier, we can predict with 81.68% accuracy, whether a customer is likely to default next month.
4) Using a K-Nearest Neighbour classifier, we can predict with 77.83% accuracy, whether a customer is likely to default next month.
5) Using a Support Vector Machine classifier, we can predict with 94.43% accuracy, whether a customer is likely to default next month.

The strongest predictors of default are the PAY_X (ie the repayment status in previous months), the LIMIT_BAL & the PAY_AMTX (amount paid in previous months).
We found that using Random Forest classifier and Support Vector Machine classifier are better.

Demographics: we see that being Female, more educated, Single and between 30-40years old means a customer is more likely to make payments on time.

References-

1. Pandas library [https://pandas.pydata.org/]
2. GeeksforGeeks
3. Python Graph Gallery [https://www.python-graph-gallery.com/43-use-categorical-variable-to-color-scatterplot-seaborn]
4. Seaborn [https://seaborn.pydata.org/]
5. Analytics Vidhya