APPROACH TO SOLUTION

CREDIT CARD DEFAULT

PROBLEM STATEMENT

To predict whether an account has a risk of default or not, by analyzing various details of the account holder in given dataset. A label of 1 signifies that the account has a definite risk of default, whereas a label of 0 denotes low/no risk accounts.

MODELS TRIED AND USED FOR THIS PROCESS:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Artificial neural network
4. XGboost Algorithm
5. Logistic regression
6. Linear regression
7. SVM

By studying the models and trying them for this models, the conclusions are written and I have chosen **Random Forest Classifier** model for this problem.

RANDOM FOREST OVER DECISION TREE:

Random forests consist of multiple single trees each based on a random sample of the training data. They are typically more accurate than single decision trees. Hence random forest is better to use over decision tree.

RANDOM FORESR OVER NEURAL NETWORK:

Neural network takes more time to train and require GPU to run, whereas Random forest doesn't require GPU to run and takes less time. The neural network will simply decimate the interpretability of your features to the point where it becomes meaningless for the sake of performance. Due to unequal 0's and 1's in Label, the neural network deviated more towards zero, and doesn't give a good performance in this case. So better to choose random forest over neural network.

RANDOM FOREST OVER XGBOOST:

The model tuning in Random Forest is much easier than in case of XGBoost. Model XGBoost is better for small, big models but it doesn't work well on sparse data and dispersed data. In this case, accuracy also is too low when compared to random forest. so Random forest is preferred over XGBoost.

RANDOM FOREST OVER SVM:

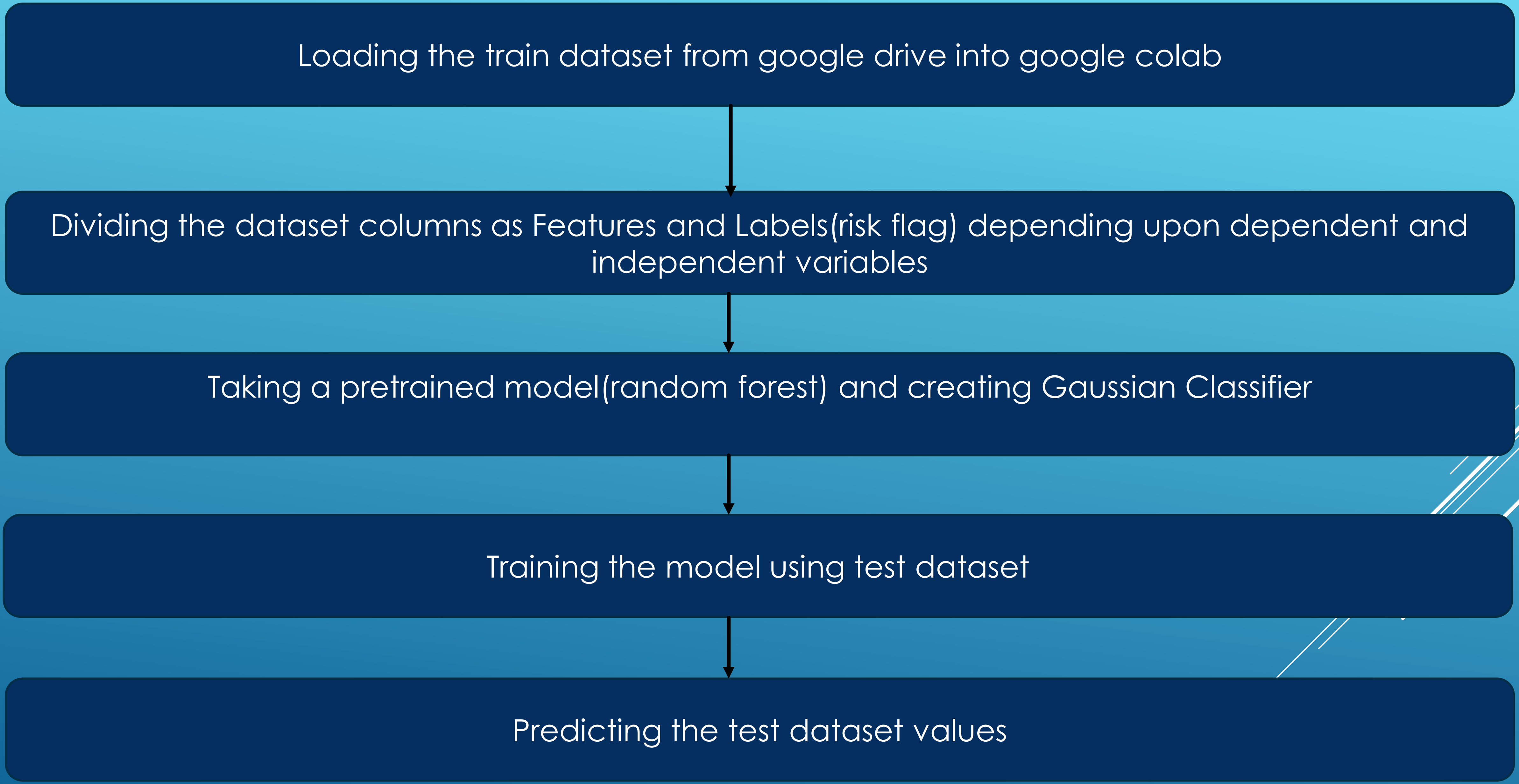
SVM algorithm is not suitable for large datasets. It is taking more time to train the model when compared to random forest. In this, it gives prediction which are close to its given values, so most probably this model is giving zero's as the train dataset contains maximum of zero's. so this model also decreases accuracy in this case, so random forest is chosen when compared to SVM.

RANDOM FOREST OVER LINEAR REGRESSION:

Random forest supports non linearity, where LR supports only linear solutions. In general cases, random forest will be having better average accuracy. For categorical independent variables, random forest are better than linear regression.

RANDOM FOREST OVER LOGISTIC REGRESSION:

Non-linear problems can't be done by logistic regression. It is bound to discrete number set. The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. Also in this case, accuracy is more Random forest. It may lead to overfitting sometimes, so on average Random forest is better than Logistic regression.



Loading the train dataset from google drive into google colab

Dividing the dataset columns as Features and Labels(risk flag) depending upon dependent and independent variables

Taking a pretrained model(random forest) and creating Gaussian Classifier

Training the model using test dataset

Predicting the test dataset values

Training the Train dataset

using Random Forest

RANDOM FOREST

Random forest are an ensemble learning method that operates by constructing a multiple decision trees at training time. It is a supervised learning algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

It works in four steps:

1. Select random samples from a given dataset.

2. Construct a decision tree for each sample and get a prediction result from each decision tree.

3. Perform a vote for each predicted result.

4. Select the prediction result with the most votes as the final prediction.

ADVANTAGES OF USING RANDOM FOREST:

1. Random forests is considered as a highly accurate method because of the number of decision trees participating in the process.
2. It does not suffer from the overfitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.
3. Random forests can also handle missing values.
4. You can get the relative feature importance, which helps in selecting the most contributing features for the classifier.

PREPROCESSING GIVEN DATASET:

Our given train dataset contains columns – “Id”, “income”, “age”, “experience”, “married”, “house ownership”, “car ownership”, “profession”, “state”, “city”, “current house years”, “current job years”, “risk flag”.

Now we are going to divide it into Features and Labels.
We include “Id”, “income”, “age”, “experience”, “married”, “house ownership”, “car ownership”, “profession”, “state”, “current house years”, “current job years” in Features.
We include “risk flag” in Labels.
“city” is not chosen as Feature because I thought it may be noise for data and misleading results.

As the model takes numerical as input, so we are using replace() and change “profession”, “state”, “married”, “house ownership”, “car ownership” strings into integer values starting from 0,1,2,.. And so on.

Similarly we are going to do the same with test dataset also.

TRAINING AND PREDICTING VALUES:

From the changed dataset, we are going to train and predict the values for given test dataset using Random Forest Classifier and prepare csv file containing Id and risk flag values of test dataset and check F1 score for that in Kaggle.

THANK YOU