Final Project

Abstract:

This project is about building or creating a model to anticipate if a borrower will repay the loan in full using supervised learning classification machine learning algorithms. The models used in this project are Logistic regression, Decision tree, Naïve Bayes, SVM, Random Forest, Neural Networks Classifiers. The project was written using Anaconda environment and Jupyter notebook kernels. Data was downloaded from Kaggle.com.

I. Introduction

The aim of this project is to classify and predict if the borrower will default on the loan using loan_data dataset and supervised learning classification machine learning algorithms. The models were used in this project can help a bank to reduce the default loans and increase total profits. The dimensions of the data are (9578,14).

II. Uses of machine learning in banking and finance

Nowadays, banking and finance benefit from using machine learning. Machine learning can assist in

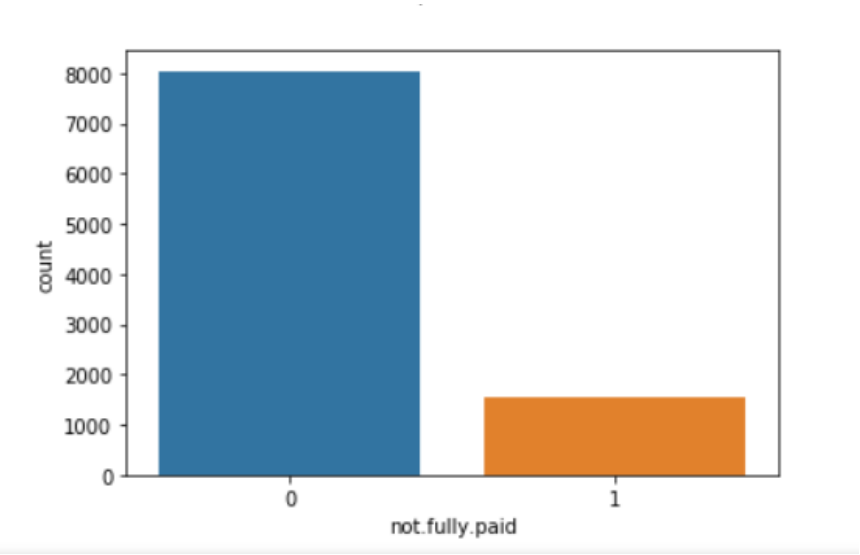
1- **Fraud detection and prevention**: Using Machine learning can help detect fraud in real-time to avoid or prevent loss.

2- **Risk prevention and detection**: Most banks like JPMorgan, Wells Fargo, Bank of America, City Bank and US banks use machine learning for risk prevention and detection.

3- **Credit scoring:** Machine learning -based credit scoring solutions is using an algorithm to predict if a customer will pay back or not. [(1)(2)]

III. Machine learning process.

According to Aurelien Geron's book, there are some important steps in machine learning process. First, look at the big picture. This step is very important. A data scientist should know the business objective and the benefits from the model. In the project problem the objective is to anticipate if a borrower will pay back the loan. Second, collect the data. A data scientist should download the data. Also, he/she will need a number of Python models such as Jupter, Pandas, Matplotlib and Scikit-Learn. A data scientist should perform some useful methods such as info (), describe (). In this project the data was downloaded from Kaggle. The project was written using Anaconda environment and Jupyter notebook kernels. Third, Visualize the data to obtain insight. A data scientist should visualize the data to gain insight. Some useful charts are scatterplot, pair plot, scatter matrix, and count plot. In this project I created a count plot for each column. (3)



Forth, Prepare the data for machine learning algorithm. This step includes data cleaning from missing values. Moreover, converting text categorical attributes to numbers. In this

project there were no missing value. There was one categorical variable(purpose), I converted it to numbers using the get_dummies method from Pandas.

```
#check for null values
loans.isnull().any()
```

```
credit.policy        False
purpose              False
int.rate             False
installment          False
log.annual.inc       False
dti                  False
fico                 False
days.with.cr.line    False
revol.bal            False
revol.util           False
inq.last.6mths       False
delinq.2yrs          False
pub.rec              False
not.fully.paid       False
dtype: bool
```

```
#show the final data info
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 19 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   credit.policy              9578 non-null   int64
 1   int.rate                   9578 non-null   float64
 2   installment                9578 non-null   float64
 3   log.annual.inc             9578 non-null   float64
 4   dti                        9578 non-null   float64
 5   fico                       9578 non-null   int64
 6   days.with.cr.line          9578 non-null   float64
 7   revol.bal                  9578 non-null   int64
 8   revol.util                 9578 non-null   float64
 9   inq.last.6mths             9578 non-null   int64
 10  delinq.2yrs                9578 non-null   int64
 11  pub.rec                    9578 non-null   int64
 12  not.fully.paid             9578 non-null   int64
 13  purpose_credit_card        9578 non-null   uint8
 14  purpose_debt_consolidation 9578 non-null   uint8
 15  purpose_educational        9578 non-null   uint8
 16  purpose_home_improvement   9578 non-null   uint8
 17  purpose_major_purchase     9578 non-null   uint8
 18  purpose_small_business     9578 non-null   uint8
dtypes: float64(6), int64(7), uint8(6)
memory usage: 1.0 MB
```

Fifth, in this step, a data scientist should evaluate his/her system on the test set. Sixth, Present the solution. Seventh, launch, observe, and maintain the system In this step, a data scientist should write a code to check the system's live performance. [3]

IV. Machine learning algorithms.

- **Decision tress** is a supervised machine algorithm which has root and leaf nods. Every node is a feature. Every leaf is a result. Decision tree can solve regression and classification problems. Decision tree strengths are requiring less work for data preparation during pre-processing, no need for normalization of data, low sensitivity to missing data, and easier interpretation. Decision tree weaknesses are massive change in the decision tree could occur as a result of minor change of data, needing long time to train the model, and high cost because of complexity and time. [4][5]

- **Logistic regression** is supervised classification algorithm used to find the probability of event success and event failure. There are two types of logistic regression binary and multi-linear functions. Logistic regression strengths are low complexity in implementation, efficiency in training, high speed at classifying unidentified records, and working well in linearly separable dataset. Logistic regression weakness is having overfitting if the number of features is greater than the number of observations. [6][7]

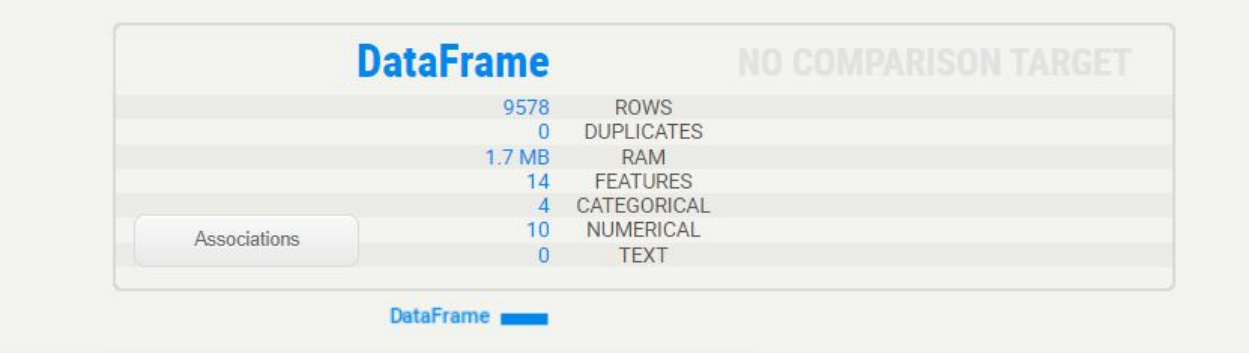
- **Naïve Bayes** is a machine learning algorithm that use Bayes Theorem. It assumes that existence of one specific feature in a class doesn't affect the existence of another one. Naïve Bayes strengths are high speed, better performance, needing less training data, and solving multi-class prediction problems. Naïve Bayes weakness is assuming that all features or predictors are independent. This limits the applicability of this algorithm. [8]

- **SVM classifier** is a machine learning algorithm. The purpose of SVM's is to discover a hyperplane that obviously classifies the data points. SVM strengths are having L2 regularization feature, working well with non-linear data, and solving both classification and regression problems. Support vector weaknesses are requiring a lot of memory, needing long training time on large datasets, and difficulty to understand [9][10]
- **Random Forest**: is a machine learning algorithm which is based on the ensemble learning technique(bagging). Random Forest strengths are solving both classification and regression problems, working good with categorical and continuous variables, and handling missing values and outliers automatically. Random Forest weaknesses are requiring more computational power and resources and requiring longer time to train data. [11]
- **Neural Network:** also known as artificial neural network or simulated neural network. It is an attempt to match biological neural network by adding hidden layer between the input and output layers. Neural Network strengths are performing multiple tasks without impacting the system

performance, accepting any number of inputs to produce the output, learning by itself, and sorting the input in its own network that can prevent loss of data. Neural network weakness is requiring much more data than other machine learning algorithms. [12][13]

V. Data preparation

First, I applied some useful functions such as Sweetviz, Autoviz, Dtale. Using Sweetviz was helpful in checking for duplicate values. There were zero duplicate values. I read the dataset CVS file using Pandas library (read_csv function). I checked for null using isnull() method. There were no null or missing values. I used the fillna() method to fill the missing values. Sweetvis picture below is showing that there are four categorical variables. There are ten numerical variables.



VI. Modeling

I divided the data columns into two variables X, y. X has all the columns except not.fully.paid. y has only the column that we need to predict (not.fully.paid). I split the data into train data and test data using the train_split() method.

```
#split the data to train and test data
from sklearn.model_selection import train_test_split
#we should predict the not.fully.paid column
X = final_data.drop('not.fully.paid', axis=1)
y = final_data['not.fully.paid']

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
```

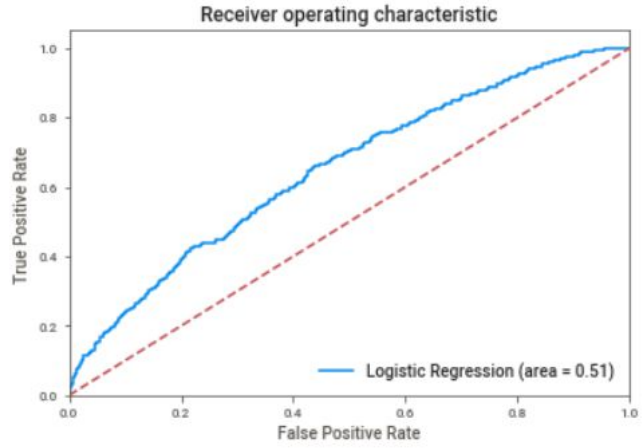
After training the model on the prepared data. We found the following:

Logistic Regression:

```
[[1989    3]
 [ 397    6]]
              precision    recall  f1-score   support

           0       0.83      1.00      0.91      1992
           1       0.67      0.01      0.03       403

    accuracy                           0.83      2395
   macro avg       0.75      0.51      0.47      2395
weighted avg       0.81      0.83      0.76      2395
```




```
AUC: 0.649
```

The model accuracy was 0.83. The area under the curve (AUC) was 0.649. FN=397, FP=3.

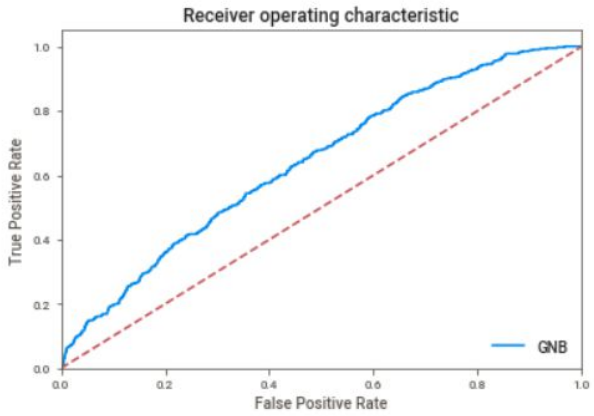
3& 397 are the number of errors.

Naive Bayes:

```
Accuracy on training set: 0.8244
Accuracy on test set: 0.8192
[[1921   71]
 [ 362   41]]
              precision    recall  f1-score   support

           0       0.84      0.96      0.90      1992
           1       0.37      0.10      0.16       403

    accuracy                           0.82      2395
   macro avg       0.60      0.53      0.53      2395
weighted avg       0.76      0.82      0.77      2395
```




```
AUC: 0.639
```

The model accuracy was 0.82. The area under the curve (AUC) was 0.639. FN=362, FP=71.

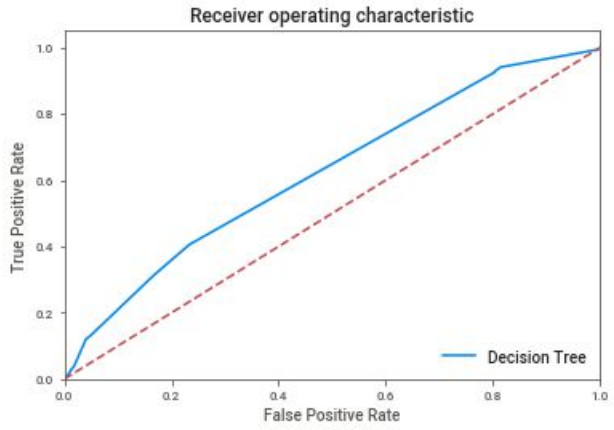
71& 362 are the number of errors.

Decision Tree:

```
Accuracy on training set: 0.8451
Accuracy on test set: 0.8251
Feature importances:
[0.42539312 0.18295795 0.09172618 0.         0.04357156 0.03305299
 0.01024392 0.         0.03912724 0.12618069 0.         0.
 0.         0.         0.         0.         0.         0.04774636]
[[1963   29]
 [ 390   13]]
              precision    recall  f1-score   support

           0       0.83      0.99      0.90      1992
           1       0.31      0.03      0.06       403

    accuracy                           0.83      2395
   macro avg       0.57      0.51      0.48      2395
weighted avg       0.75      0.83      0.76      2395
```




```
AUC: 0.623
```

The model accuracy was 0.83. The area under the curve (AUC) was 0.623. FN=390, FP=29.

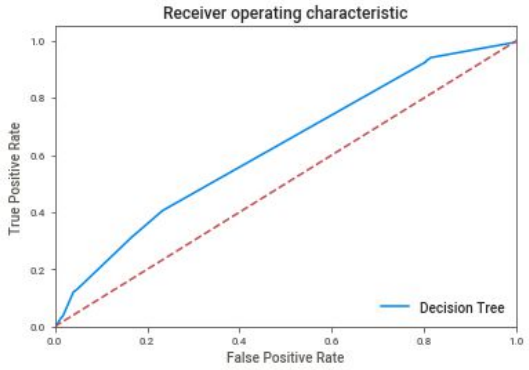
29 & 390 are the number of errors.

Random Forest:

```
Accuracy on training set: 0.843
Accuracy on test set: 0.832
Feature importances:
[0.24376173 0.14489197 0.07097211 0.03636258 0.01984341 0.12463417
 0.03073428 0.03141087 0.06212264 0.16567115 0.00292941 0.0047209
 0.00419389 0.00148119 0.00297475 0.00080847 0.00159369 0.05089279]
[[1992    0]
 [ 403    0]]
              precision    recall  f1-score   support

           0       0.83      1.00      0.91      1992
           1       0.00      0.00      0.00       403

    accuracy                           0.83      2395
   macro avg       0.42      0.50      0.45      2395
weighted avg       0.69      0.83      0.76      2395
```




```
AUC: 0.623
```

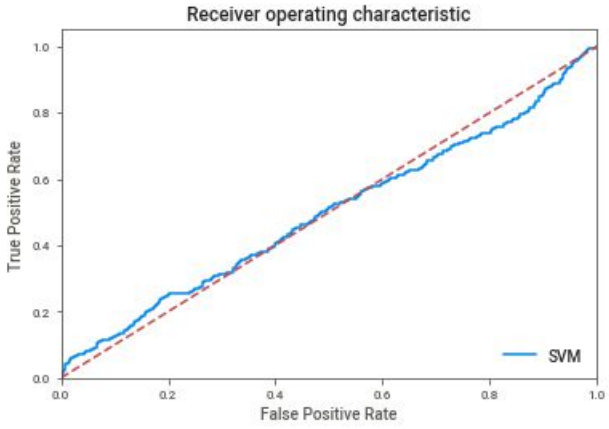
The model accuracy was 0.83. The area under the curve (AUC) was 0.623. FN=403, FP=0. 0 & 403 are the number of errors.

Support Vector Machine:

```
Accuracy on training set: 0.84
Accuracy on test set: 0.83
[[1992    0]
 [ 403    0]]
              precision    recall  f1-score   support

           0       0.83      1.00      0.91      1992
           1       0.00      0.00      0.00       403

    accuracy                           0.83      2395
   macro avg       0.42      0.50      0.45      2395
weighted avg       0.69      0.83      0.76      2395
```




```
AUC: 0.496
```

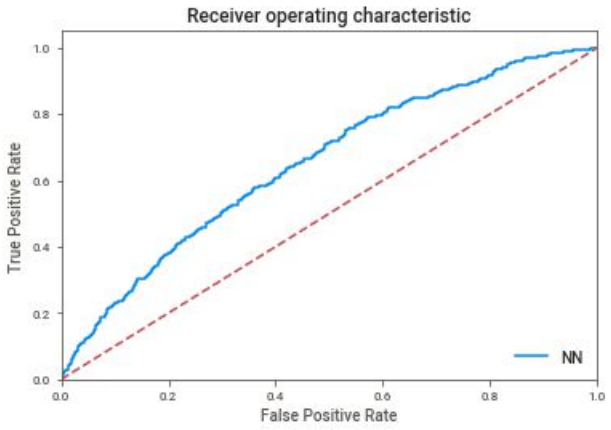
The model accuracy was 0.83. The area under the curve (AUC) was 0.496. FN=403, FP=0. 0 & 403 are the number of errors

Neural Networks:

```
Accuracy on training set: 0.863
Accuracy on test set: 0.825
[[1954   38]
 [ 382   21]]
              precision    recall  f1-score   support

           0       0.84      0.98      0.90      1992
           1       0.36      0.05      0.09       403

    accuracy                           0.82      2395
   macro avg       0.60      0.52      0.50      2395
weighted avg       0.76      0.82      0.77      2395
```




```
AUC: 0.651
```

The model accuracy was 0.82. The area under the curve (AUC) was 0.651. FN=382,

FP=38. 382 & 38 are the number of errors

Classifier	Accuracy	AUC	FP	FN	TP	TN	TP+TN	FP+FN
Logistic Regression	0.83	0.649	3	397	1989	6	1995	400
Naïve Bayes	0.82	0.639	71	362	1921	41	1962	433
Decision Tree	0.83	0.623	29	390	1963	13	1976	419
Random Forest	0.83	0.623	0	403	1992	0	1992	403
SVM	0.83	0.496	0	403	1992	0	1992	403
Neural Network	0.82	0.651	38	382	1954	21	1975	420

AUC score indicates how good our model is at differentiating between classes. The higher AUC, the better the model. From the table above we can see that the algorithms have comparable performance because they have almost the same accuracy value and AUC value expect SVM which has low AUC. In this problem, TP means the model predicted that borrower will pay the loan and in fact he/she paid it. TN means that the model predicted the borrower will not pay the loan and in fact he/she did not pay the loan. FP (Type 1 error) means the model predicted that the borrower will pay but in fact the borrower did not pay. FN (type 2 error) The model predicted that the borrower will not pay the loan but in fact the borrower paid the loan. We should maximize the TP+TN in order to maximize the profit. And, Minimize the FN+FP to minimize the loss. The best performance model for generalizing unseen data in order was Logistic regression (it has high accuracy and AUC and TP+TN and low FN+FP).

VII. Conclusion

There are many ways to optimize these models. First, adding more data. The more data the better and more accurate the models. Second, applying multiple algorithms. We need to try all suited models and check the performance to find the best model with the highest accuracy. Third, treating missing and outliers' value. The accuracy of a model is usually decreased by missing values and outliers. Forth, using other types or models such as deep learning algorithm. Fifth, using feature engineering. We can remove features that are not beneficial and create new features that better represent the underlying problem of the model. [14]

References:

1- Horacio, & Here, P. (2020, April 08). Is machine learniing in banking sector the most trending thing now? Retrieved February 28, 2021, from https://www.fintechnews.org/is-machine-learning-in-banking-sector/

2- Editor. (2019, December 16). AI and machine learning in Finance: Use cases in banking, Insurance, investment, and CX. Retrieved February 28, 2021, from https://www.altexsoft.com/blog/datascience/machine-learning-use-cases-in-finance/

3- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo: O'Reilly.

4- Guide to decision Tree Algorithm: Applications, pros & cons & Example. (2020, December 08). Retrieved February 10, 2021, from https://www.upgrad.com/blog/guide-to-decision-tree-algorithm/

5- K, D. (2020, December 26). Top 5 advantages and disadvantages of decision tree algorithm. Retrieved February 10, 2021, from https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a

6- Pant, A. (2019, January 22). Introduction to logistic regression. Retrieved February 10, 2021, from https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

7- Advantages and disadvantages of logistic regression. (2020, September 02). Retrieved February 10, 2021, from https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

8- Naive Bayes explained: FUNCTION, advantages & DISADVANTAGES, applications in 2021. (2021, January 11). Retrieved February 10, 2021, from https://www.upgrad.com/blog/naive-bayes-explained/

9- Kumar, N. (n.d.). Advantages and disadvantages of SVM (support Vector machine) in machine learning. Retrieved February 10, 2021, from http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-svm.html

10- Gandhi, R. (2018, July 05). Support vector machine - introduction to machine learning algorithms. Retrieved February 10, 2021, from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

11- Kumar, N. (n.d.). Advantages and disadvantages of random forest algorithm in machine learning. Retrieved March 04, 2021, from http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html

12- -, M., By, -, Thomas, M., Thomas, M., & Thomas, I. (2020, November 22). Neural networks: Advantages and applications. Retrieved March 04, 2021, from https://www.marktechpost.com/2019/04/18/introduction-to-neural-networks-advantages-and-applications/

13- Donges, N. (n.d.). 4 reasons why deep learning and neural networks aren't always the right choice. Retrieved March 04, 2021, from https://builtin.com/data-science/disadvantages-neural-networks

14- Sunil RayI am a Business Analytics and Intelligence professional with deep experience in the Indian Insurance industry. I have worked for various multi-national Insurance

companies in last 7 years. (2020, June 26). How to increase accuracy of machine learning model. Retrieved February 10, 2021, from https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/