LOAN ELIGIBILITY PREDICTION USING SUPERVISED LEARNING

(Logistic Regression, Linear Regression, Naïve Bayes, KNN, SVM, Gradient Descent)

Pavithra Shankar Babu

CONTENTS

- Introduction
- Business Question
- Problem Statement & Problem Solution
- Software Used
- Proposed Work
- Methodology

INTRODUCTION

Loans are the core business of banks. The main profit comes directly from the loan's interest. The loan companies grant a loan after an intensive process of verification and validation. However, they still don't have assurance if the applicant can repay the loan with no difficulties. In this Project, a predictive model ha been built to predict if an applicant is eligible for the loan or not.

BUSINESS QUESTION

"What factors most accurately predict whether a loan applicant is likely to repay their loan on time, and how can this information be leveraged to make informed decisions regarding loan approval or rejection?"

PROBLEM STATEMENT

- Intensive time Consumption process of verification and validation.
- Human errors can be introduced during the validation process.
- No cross referencing previous loan records.

PROBLEM SOLUTION

- Our Machine learning model calculates all the parameters given and predicts if the applicant is eligible for loan or not in very less time.
- Time required for verification, and validation reduces significantly.

SOFTWARE USED

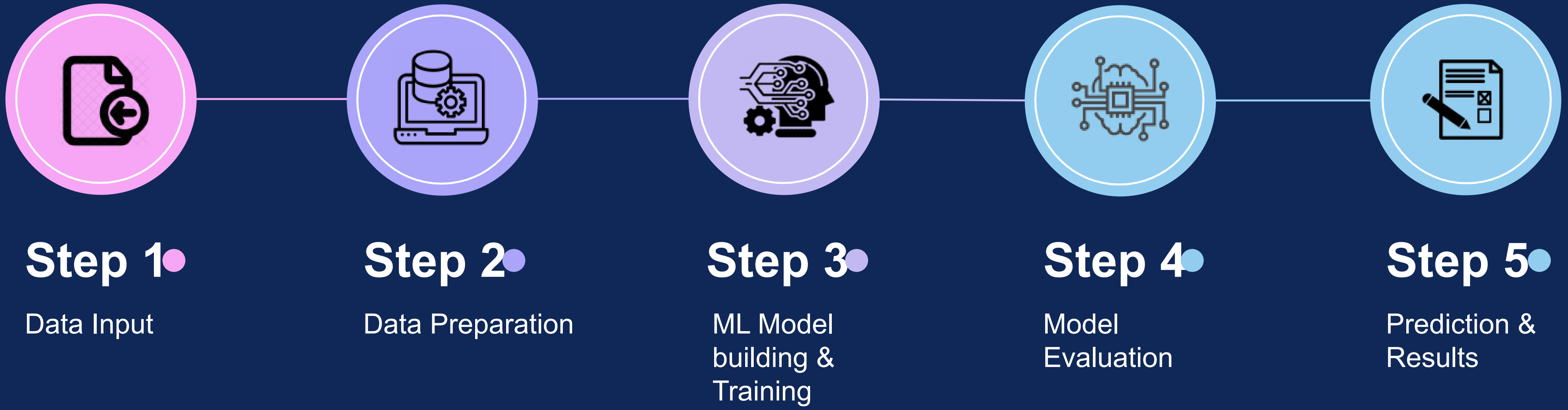
Libraries Used:

- Pandas
- Numpy
- Seaborn
- Matplotlib.pyplot
- Sklearn.preprocessing
- Sklearn.svm
- Sklearn.linear_model.LinearRegression

Language Used:

- Python

PROPOSED WORK



Step 1

Data Input

Step 2

Data Preparation

Step 3

ML Model building & Training

Step 4

Model Evaluation

Step 5

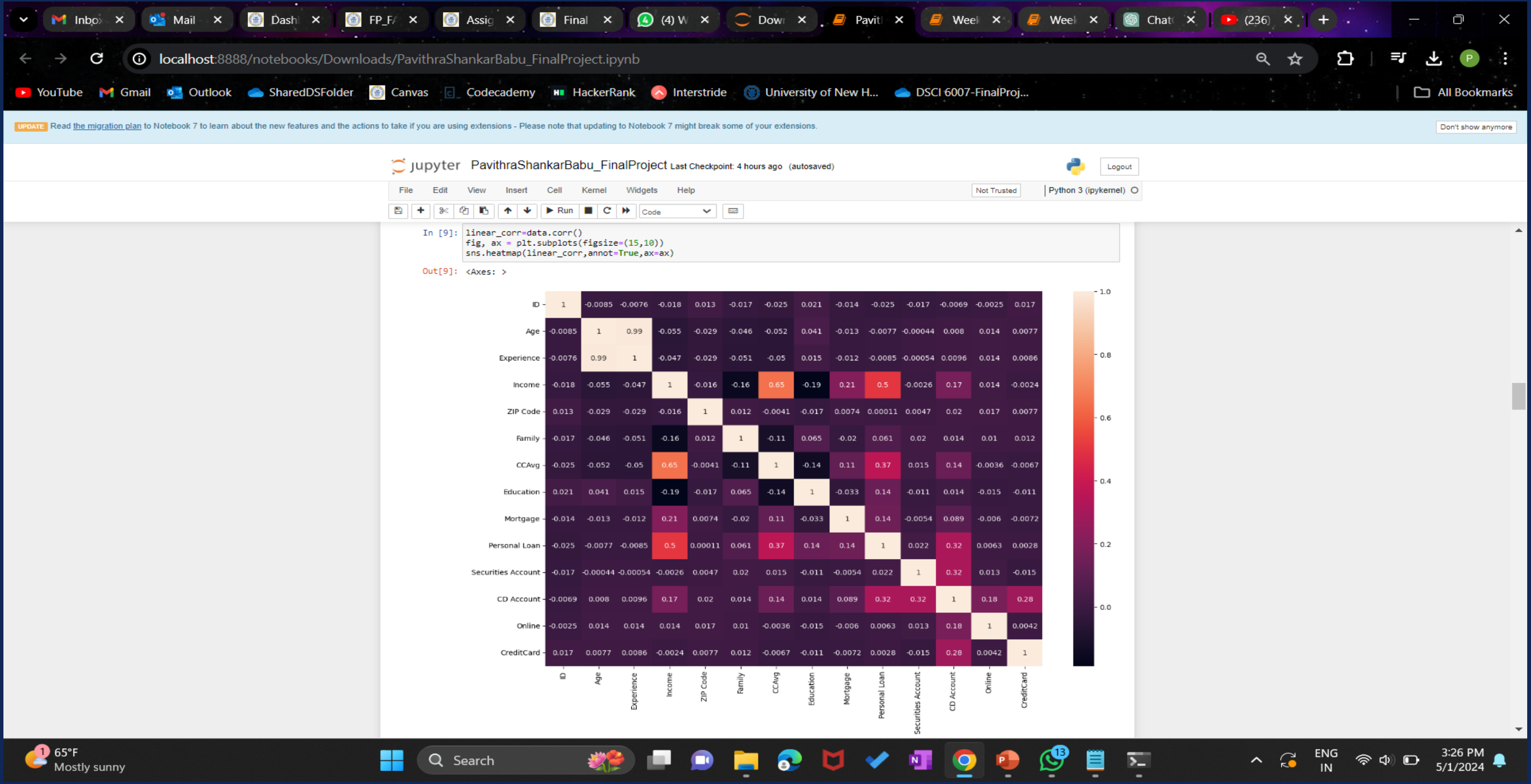
Prediction & Results

METHODOLOGY

- Data Collection: A single dataset containing historical loan application data, from Kaggle is taken including features such as applicant demographics, financial information, credit history, and loan outcome (eligible or not eligible).
- Analyzing Data: Explore the data to understand its structure, distributions, and relationships between variables. Determine which features are relevant for predicting loan eligibility and identify any patterns or trends in the data.
- Data Cleaning: Clean the data by handling missing values, outliers, and inconsistencies. Ensure that all features are in the appropriate format for analysis and preprocessing, such as numerical encoding for categorical variables.
- Model Building: Implement various supervised learning algorithms for loan eligibility prediction, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Regression, Linear Regression, Naive Bayes, and Gradient Descent. Train each model on the training data and tune hyperparameters as needed.
- Evaluating Performance Metrics of Models: Evaluate the performance of each model using appropriate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Compare the performance of different algorithms to determine which one works best for predicting loan eligibility in this specific context. Tune the selected model further if necessary to improve its performance.

RESULT

Correlation Matrix:



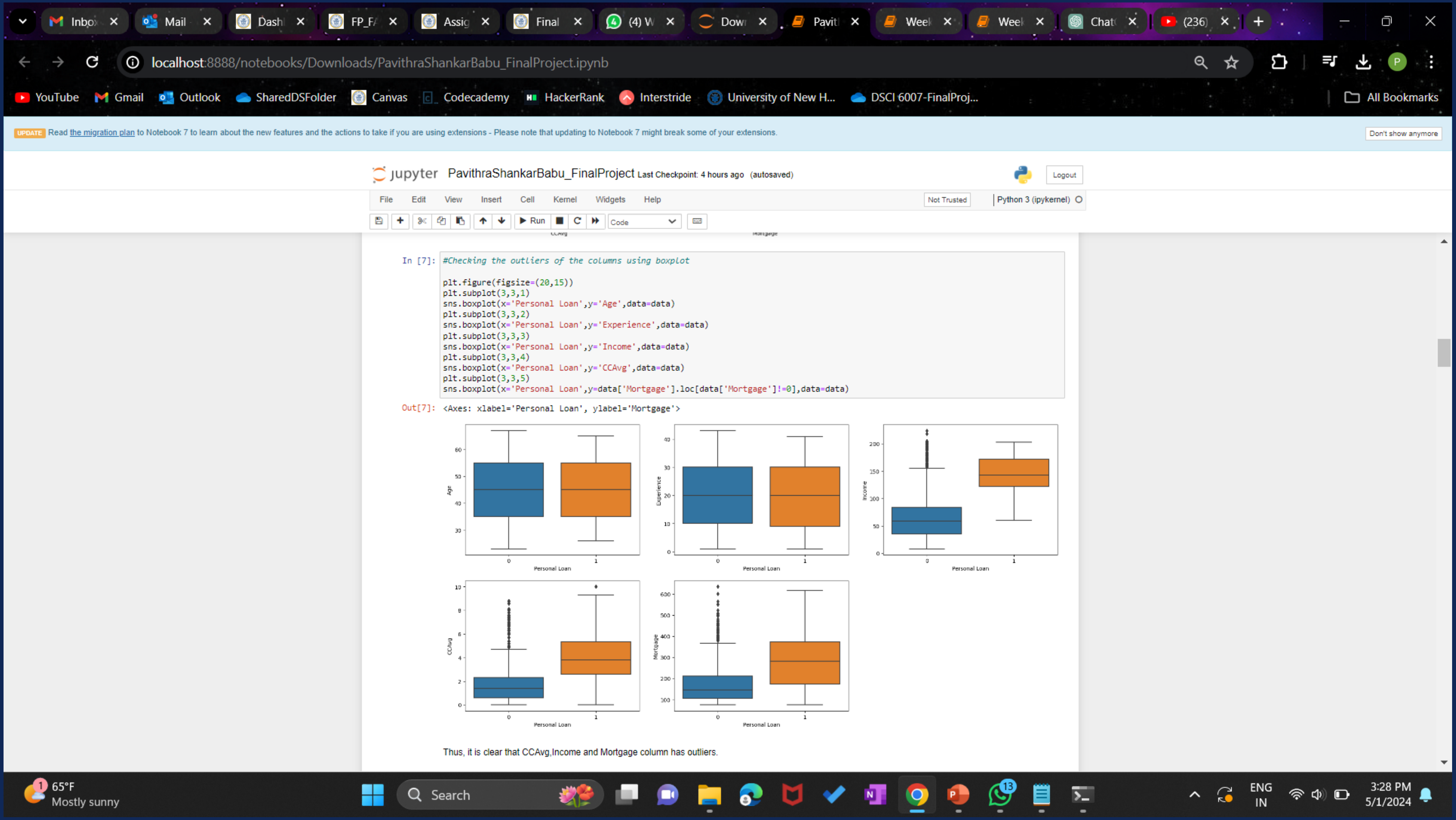
Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

PavithraShankarBabu_FinalProject Last Checkpoint: 4 hours ago (autosaved)

```
In [9]: linear_corr=data.corr()
        fig, ax = plt.subplots(figsize=(15,10))
        sns.heatmap(linear_corr,annot=True,ax=ax)

Out[9]: <Axes: >
```

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
ID	1	-0.0085	-0.0076	-0.018	0.013	-0.017	-0.025	0.021	-0.014	-0.025	-0.017	-0.0069	-0.0025	0.017
Age	-0.0085	1	0.99	-0.055	-0.029	-0.046	-0.052	0.041	-0.013	-0.0077	-0.00044	0.008	0.014	0.0077
Experience	-0.0076	0.99	1	-0.047	-0.029	-0.051	-0.05	0.015	-0.012	-0.0085	-0.00054	0.0096	0.014	0.0086
Income	-0.018	-0.055	-0.047	1	-0.016	-0.16	0.65	-0.19	0.21	0.5	-0.0026	0.17	0.014	-0.0024
ZIP Code	0.013	-0.029	-0.029	-0.016	1	0.012	-0.0041	-0.017	0.0074	0.00011	0.0047	0.02	0.017	0.0077
Family	-0.017	-0.046	-0.051	-0.16	0.012	1	-0.11	0.065	-0.02	0.061	0.02	0.014	0.01	0.012
CCAvg	-0.025	-0.052	-0.05	0.65	-0.0041	-0.11	1	-0.14	0.11	0.37	0.015	0.14	-0.0036	-0.0067
Education	0.021	0.041	0.015	-0.19	-0.017	0.065	-0.14	1	-0.033	0.14	-0.011	0.014	-0.015	-0.011
Mortgage	-0.014	-0.013	-0.012	0.21	0.0074	-0.02	0.11	-0.033	1	0.14	-0.0054	0.089	-0.006	-0.0072
Personal Loan	-0.025	-0.0077	-0.0085	0.5	0.00011	0.061	0.37	0.14	0.14	1	0.022	0.32	0.0063	0.0028
Securities Account	-0.017	-0.00044	-0.00054	-0.0026	0.0047	0.02	0.015	-0.011	-0.0054	0.022	1	0.32	0.013	-0.015
CD Account	-0.0069	0.008	0.0096	0.17	0.02	0.014	0.14	0.014	0.089	0.32	0.32	1	0.18	0.28
Online	-0.0025	0.014	0.014	0.014	0.017	0.01	-0.0036	-0.015	-0.006	0.0063	0.013	0.18	1	0.0042
CreditCard	0.017	0.0077	0.0086	-0.0024	0.0077	0.012	-0.0067	-0.011	-0.0072	0.0028	-0.015	0.28	0.0042	1

Outliers:



UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

PavithraShankarBabu_FinalProject Last Checkpoint: 4 hours ago (autosaved)

In [7]:

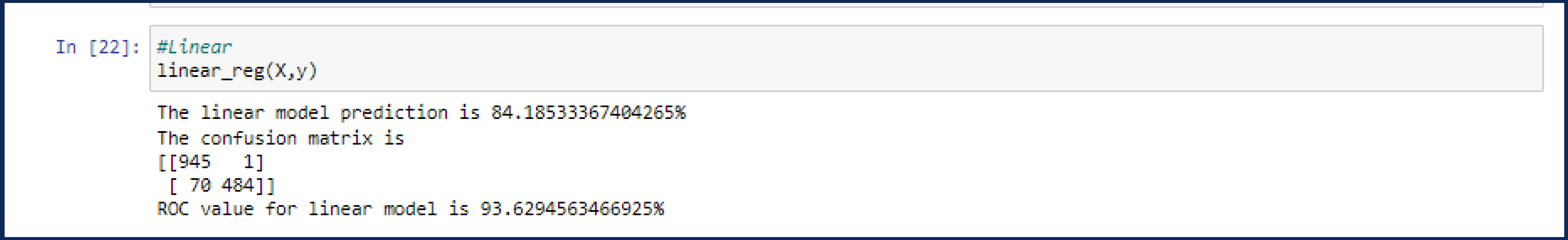
```
#Checking the outliers of the columns using boxplot

plt.figure(figsize=(20,15))
plt.subplot(3,3,1)
sns.boxplot(x='Personal Loan',y='Age',data=data)
plt.subplot(3,3,2)
sns.boxplot(x='Personal Loan',y='Experience',data=data)
plt.subplot(3,3,3)
sns.boxplot(x='Personal Loan',y='Income',data=data)
plt.subplot(3,3,4)
sns.boxplot(x='Personal Loan',y='CCAvg',data=data)
plt.subplot(3,3,5)
sns.boxplot(x='Personal Loan',y=data['Mortgage'].loc[data['Mortgage']!=0],data=data)
```

Out[7]:

```
<Axes: xlabel='Personal Loan', ylabel='Mortgage'>
```

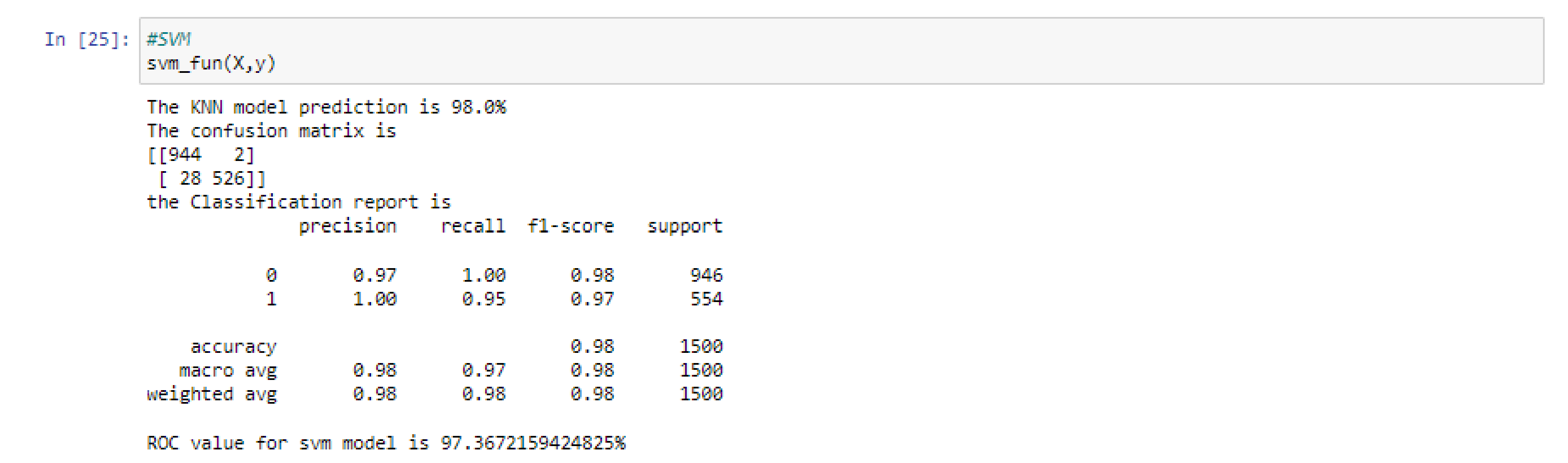
Thus, it is clear that CCAvg,Income and Mortgage column has outliers.

Linear Regression:



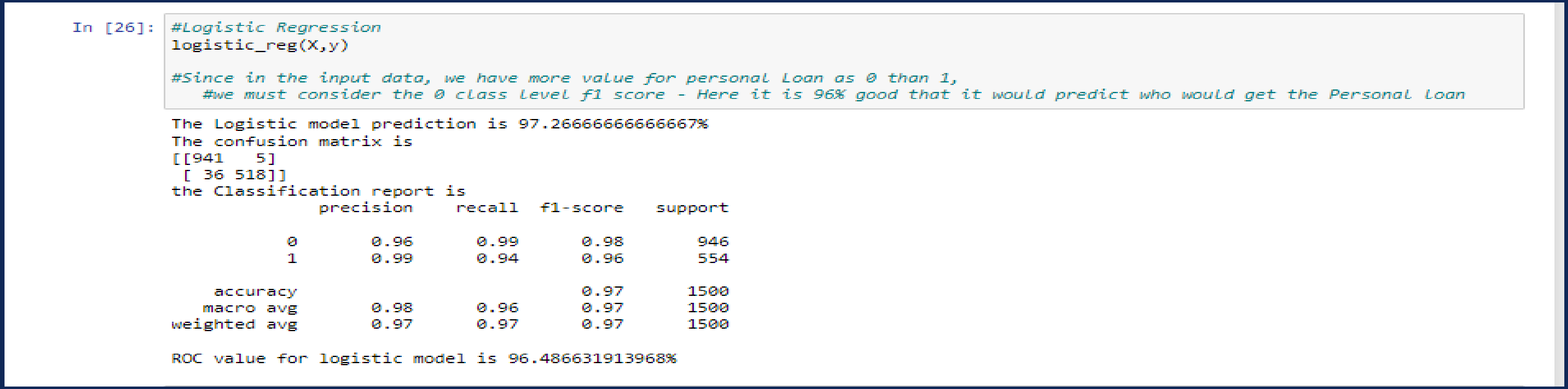
```
In [22]: #Linear
         linear_reg(X,y)

         The linear model prediction is 84.18533367404265%
         The confusion matrix is
         [[945   1]
          [ 70 484]]
         ROC value for linear model is 93.6294563466925%
```

SVM:



```
In [25]: #SVM
         svm_fun(X,y)

         The KNN model prediction is 98.0%
         The confusion matrix is
         [[944   2]
          [ 28 526]]
         the Classification report is
                       precision    recall  f1-score   support

                    0       0.97      1.00      0.98       946
                    1       1.00      0.95      0.97       554

             accuracy                           0.98      1500
            macro avg       0.98      0.97      0.98      1500
         weighted avg       0.98      0.98      0.98      1500

         ROC value for svm model is 97.3672159424825%
```

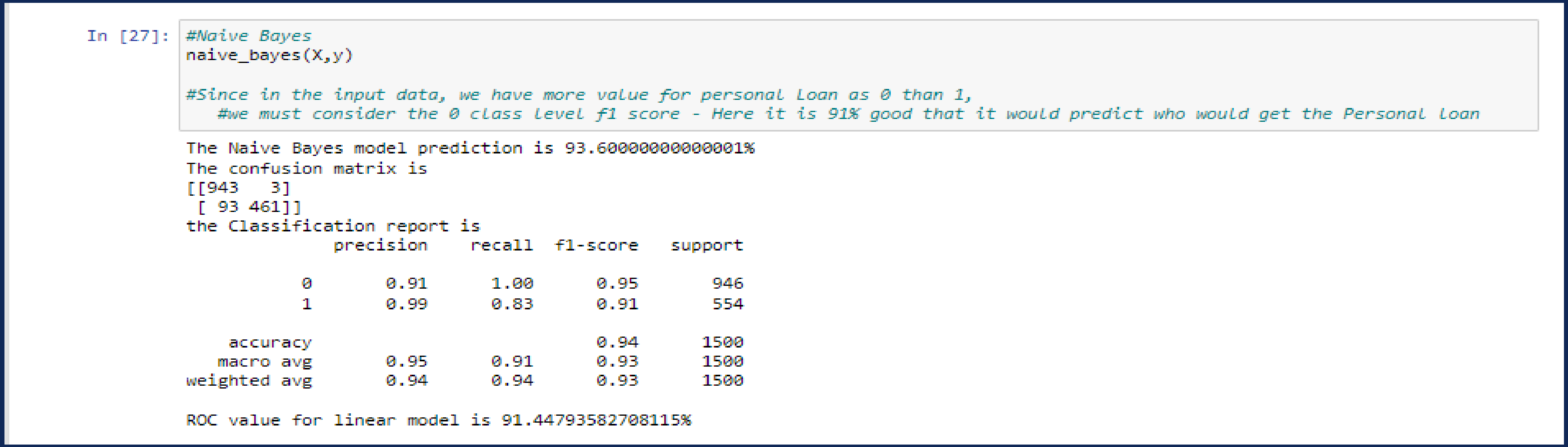
Logistic Regression:



```
In [26]: #Logistic Regression
         logistic_reg(X,y)

         #Since in the input data, we have more value for personal Loan as 0 than 1,
            #we must consider the 0 class level f1 score - Here it is 96% good that it would predict who would get the Personal Loan
```

```
The Logistic model prediction is 97.26666666666667%
The confusion matrix is
[[941   5]
 [ 36 518]]
the Classification report is
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       946
           1       0.99      0.94      0.96       554

    accuracy                           0.97      1500
   macro avg       0.98      0.96      0.97      1500
weighted avg       0.97      0.97      0.97      1500

ROC value for logistic model is 96.4866319139968%
```

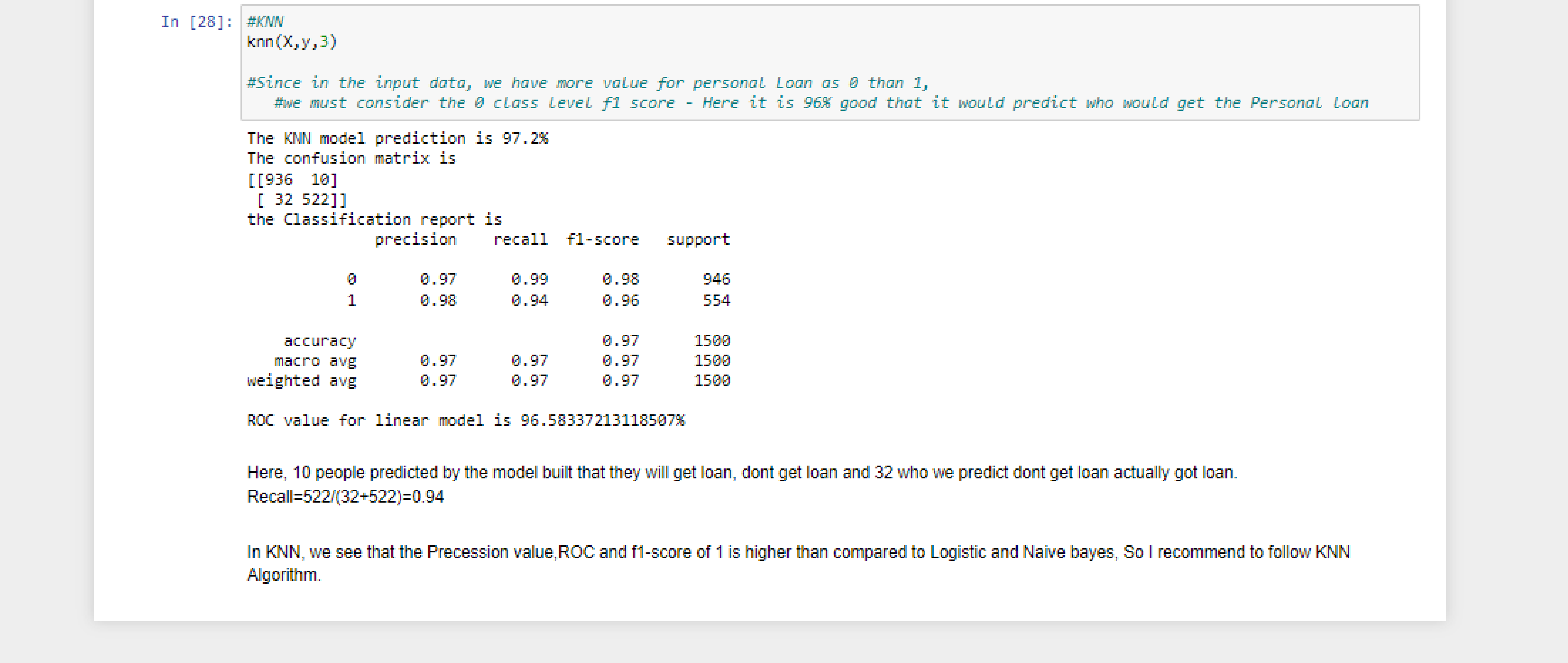
Naïve Bayes:



```
In [27]: #Naive Bayes
         naive_bayes(X,y)

         #Since in the input data, we have more value for personal Loan as 0 than 1,
            #we must consider the 0 class level f1 score - Here it is 91% good that it would predict who would get the Personal Loan
```

```
The Naive Bayes model prediction is 93.60000000000001%
The confusion matrix is
[[943   3]
 [ 93 461]]
the Classification report is
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       946
           1       0.99      0.83      0.91       554

    accuracy                           0.94      1500
   macro avg       0.95      0.91      0.93      1500
weighted avg       0.94      0.94      0.93      1500

ROC value for linear model is 91.44793582708115%
```

KNN:



In [28]:

```
#KNN
knn(X,y,3)

#Since in the input data, we have more value for personal Loan as 0 than 1,
   #we must consider the 0 class level f1 score - Here it is 96% good that it would predict who would get the Personal Loan
```

```
The KNN model prediction is 97.2%
The confusion matrix is
[[936  10]
 [ 32 522]]
the Classification report is
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       946
           1       0.98      0.94      0.96       554

    accuracy                           0.97      1500
   macro avg       0.97      0.97      0.97      1500
weighted avg       0.97      0.97      0.97      1500

ROC value for linear model is 96.58337213118507%
```

Here, 10 people predicted by the model built that they will get loan, dont get loan and 32 who we predict dont get loan actually got loan.
Recall=522/(32+522)=0.94

In KNN, we see that the Precession value,ROC and f1-score of 1 is higher than compared to Logistic and Naive bayes, So I recommend to follow KNN Algorithm.

SUMMARY

Estimating the probability of default for approved loans: This can help the bank assess the risk associated with approving a loan and make informed decisions about loan terms and interest rates.

Predicting the loan amount: Given a set of applicant characteristics, the model can predict the amount of loan that is likely to be approved, providing valuable information to both the bank and the applicant. By developing this accurate predictive model, financial institutions can streamline their loan approval processes, minimize risk, and improve customer satisfaction.

THANK YOU

Pavithra Shankar Babu

pshan4@unh.newhaven.edu