**Loan Application Status Prediction using Machine Learning**

**Problem Statement:**

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 is able to repay the loan or not. Banks want to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others.

I’ll build a predictive model to predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset.

**Data Analysis:**

Source: <https://github.com/dsrscientist/DSData/blob/master/loan_prediction.csv>

The data set have 12 independent variables and 1 target variable, i.e. Loan\_Status in the training dataset We have 614 rows and 13 columns in the train dataset. The column names are as follows:

Independent Variables:

Loan\_ID

Gender

Married

Dependents

Education

Self\_Employed

ApplicantIncome

CoapplicantIncome

Loan\_Amount

Loan\_Amount\_Term

Credit History

Property\_Area

Dependent Variable (Target Variable):

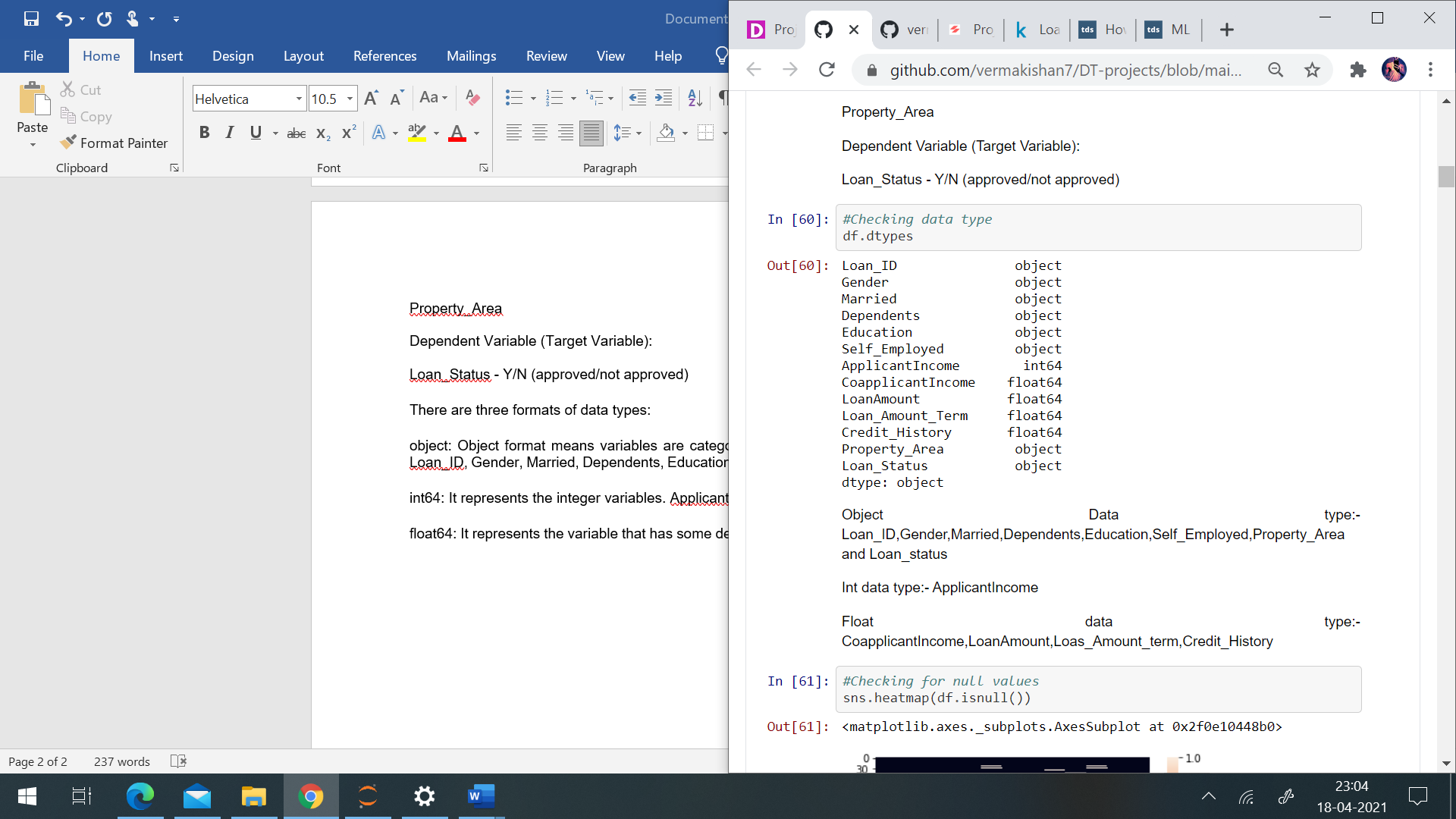
Loan\_Status - Y/N (approved/not approved)

There are three formats of data types:

object: Object format means variables are categorical. Categorical variables in our dataset are Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Status.

int64: It represents the integer variables. ApplicantIncome is of this format.

float64: It represents the variable that has some decimal values involved. They are also numerical



There are null values in the following columns: -

Gender- 13

Married- 03

Dependents- 15

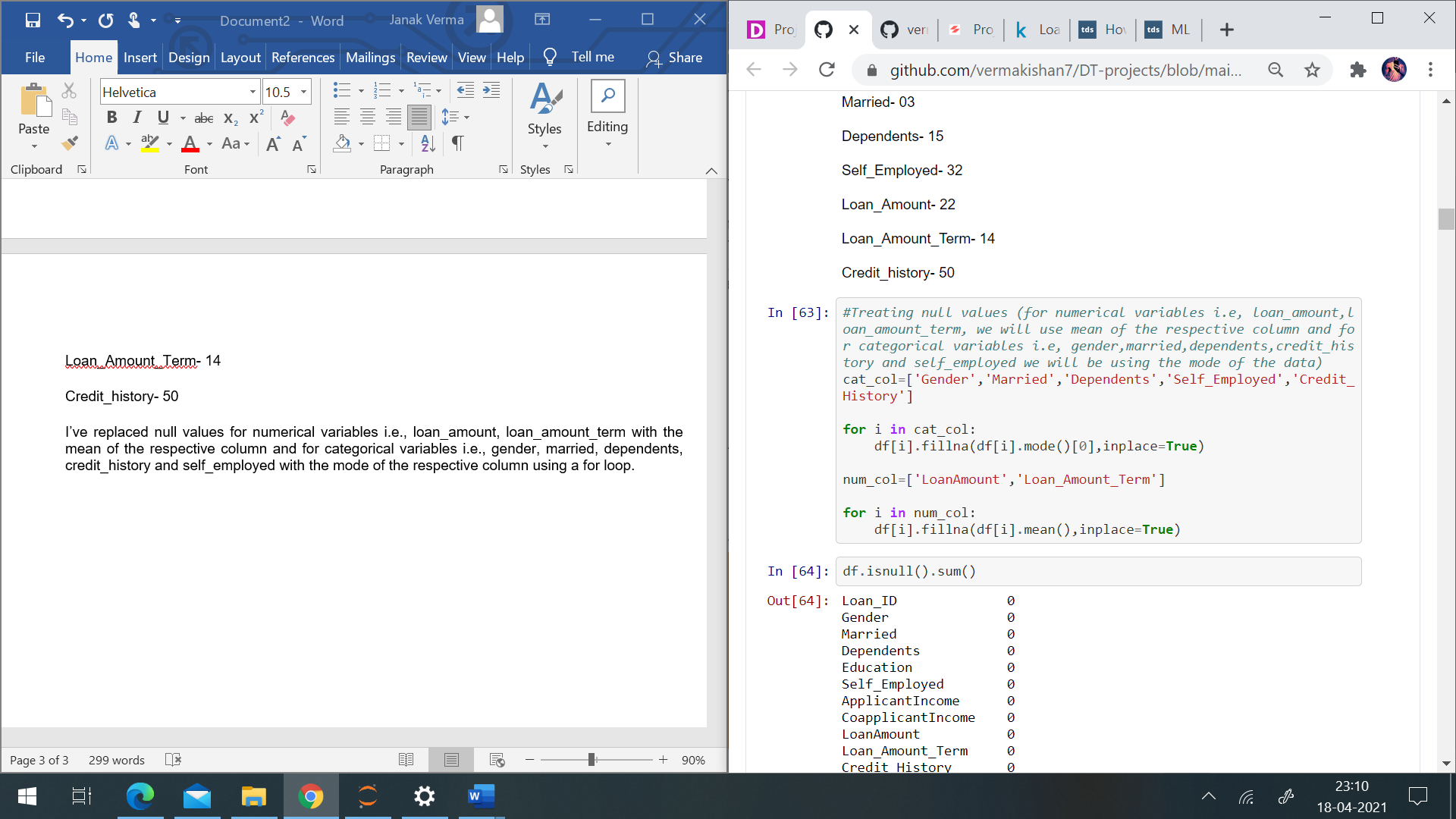
Self\_Employed- 32

Loan\_Amount- 22

Loan\_Amount\_Term- 14

Credit\_history- 50

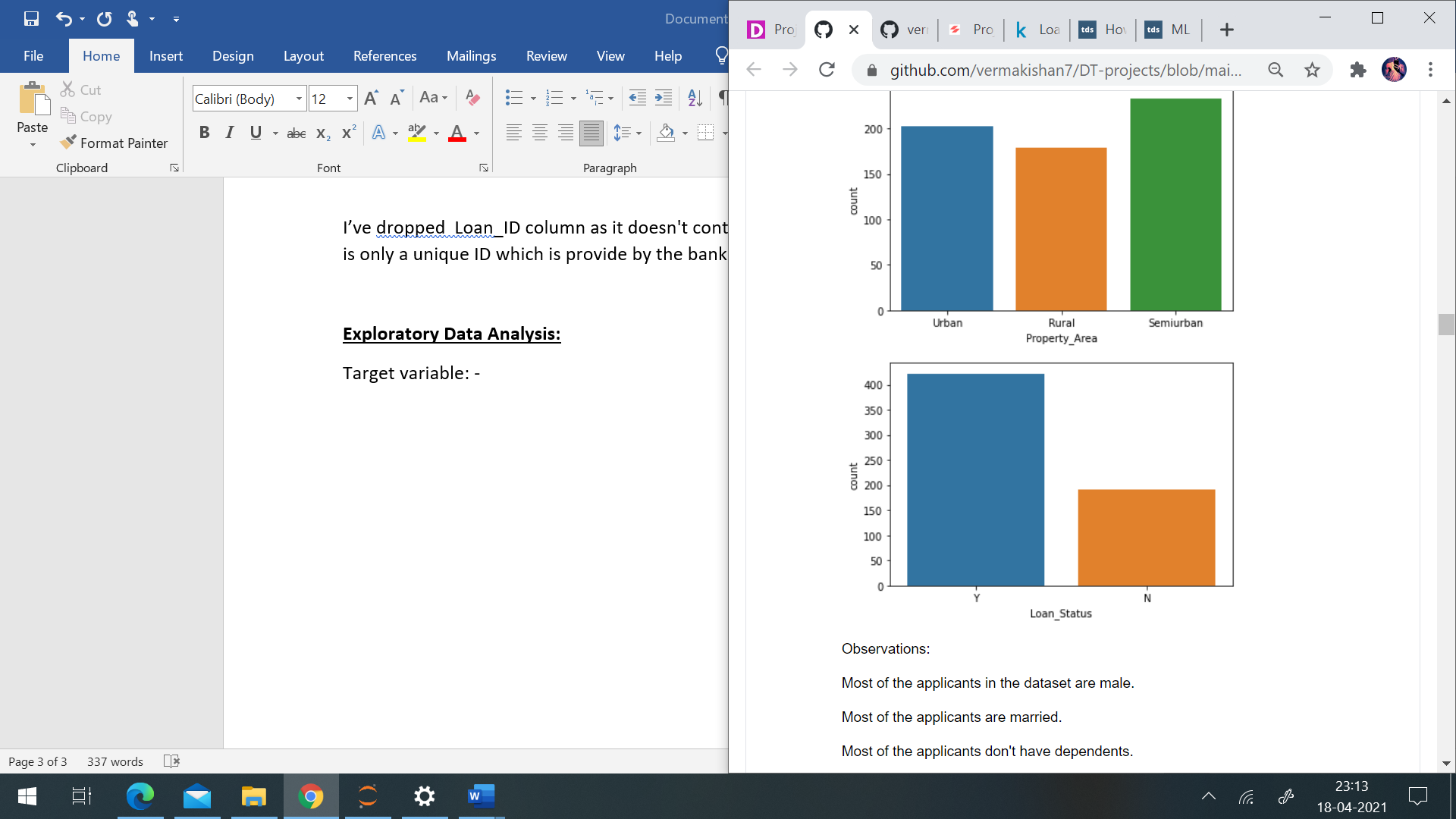
I’ve replaced null values for numerical variables i.e., loan\_amount, loan\_amount\_term with the mean of the respective column and for categorical variables i.e., gender, married, dependents, credit\_history and self\_employed with the mode of the respective column using a for loop.



I’ve dropped Loan\_ID column as it doesn't contribute towards prediction of loan approval. It is only a unique ID which is provide by the bank to the customer to identify the customers.

**Exploratory Data Analysis:**

Target variable: -



The loan of 422(around 69%) people out of 614 were approved.

Now, let's visualize each variable separately. Different types of variables are Categorical, ordinal, and numerical.

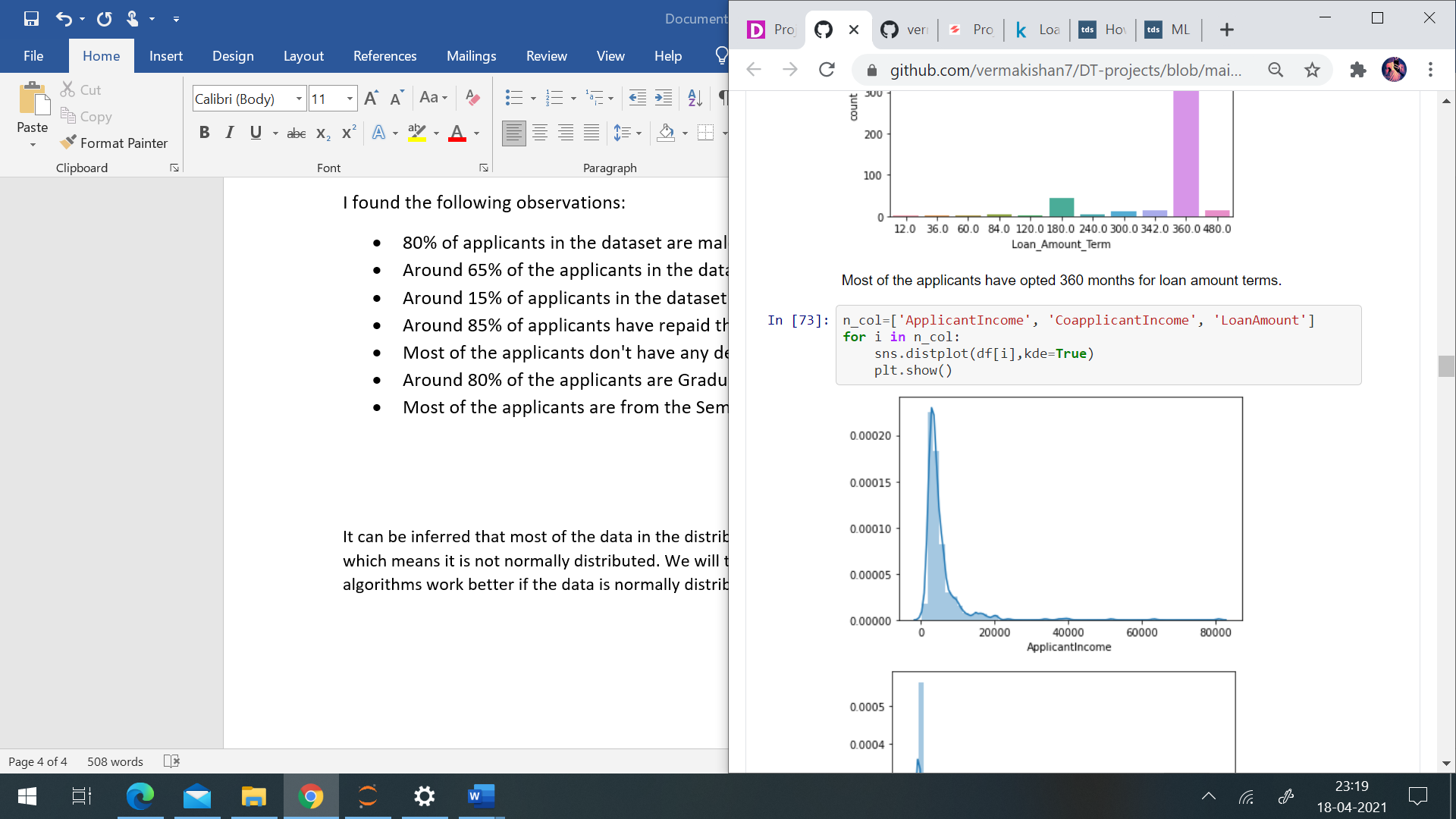
Categorical features: These features have categories (Gender, Married, Self\_Employed, Credit\_History, Loan\_Status)

Ordinal features: Variables in categorical features having some order involved (Dependents, Education, Property\_Area)

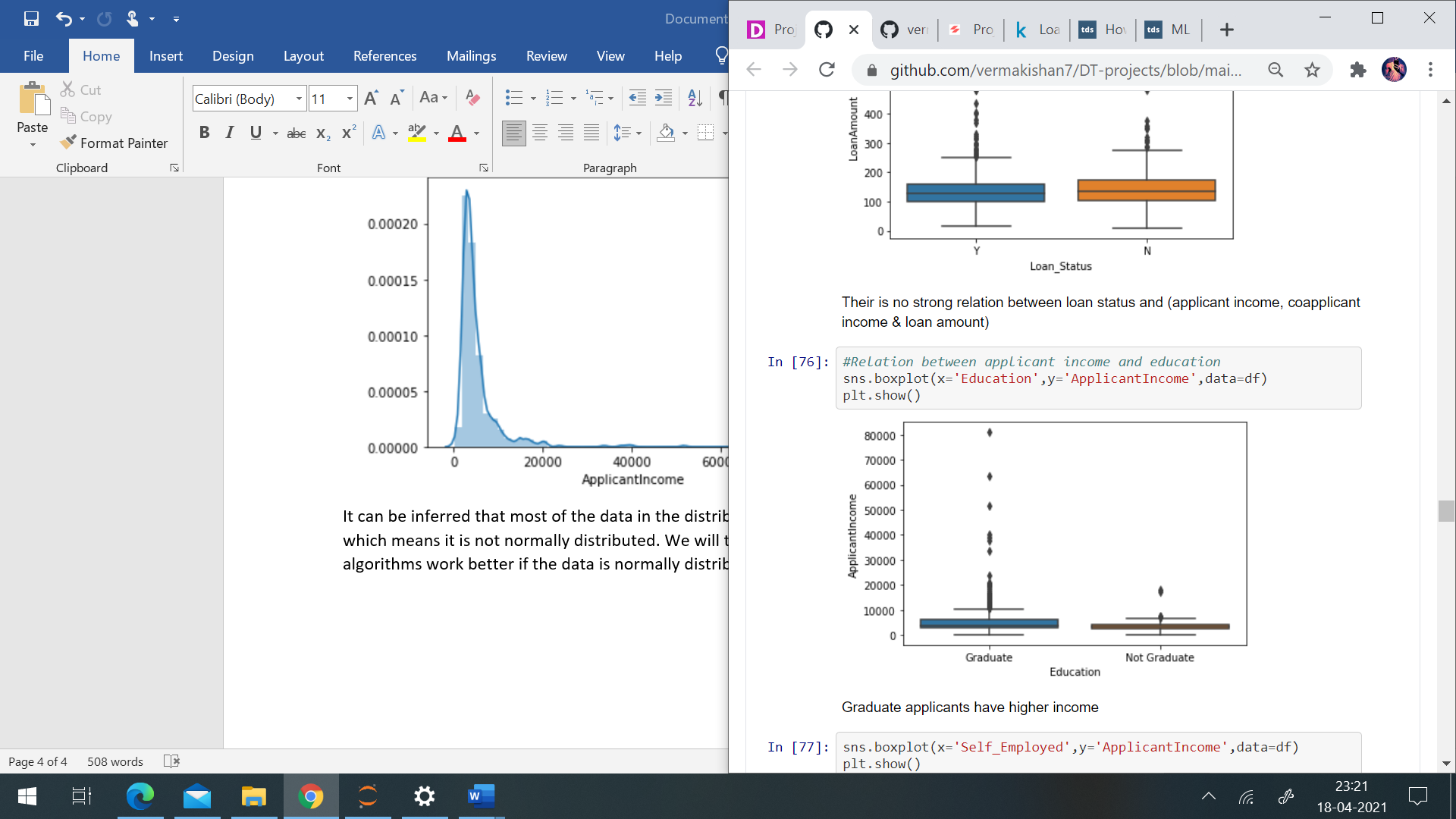
Numerical features: These features have numerical values (ApplicantIncome, Co-applicantIncome, LoanAmount, Loan\_Amount\_Term)

I found the following observations:

* 80% of applicants in the dataset are male.
* Around 65% of the applicants in the dataset are married.
* Around 15% of applicants in the dataset are self-employed.
* Around 85% of applicants have repaid their doubts.
* Most of the applicants don't have any dependents.
* Around 80% of the applicants are Graduate.
* Most of the applicants are from the Semiurban area.



It can be inferred that most of the data in the distribution of applicant income are towards the left which means it is not normally distributed. We will try to make it normal in later sections as algorithms work better if the data is normally distributed.



It can be seen that there are a higher number of graduates with very high incomes, which are appearing to be outliers.

After visualizing loan status with independent variables, the following observations were found:

* More males have taken loan than females.
* Approved loans are higher for married applicants.
* Approved loans for applicants having no dependents are high.
* Loans of graduate applicants are likely to be approved than non graduate applicants.
* Applicants which are not self-employed i.e., which are salaried applicants are likely to get their loan approved.
* Majority of loans are taken with 360 months of loan amount terms.
* Applicants having credit history (1) are likely to get their loan approved.
* Applicants from semi urban area have higher loan approval rate.

It can be seen that the proportion of approved loans is higher for Low and Average Loan Amount as compared to that of High Loan Amount.

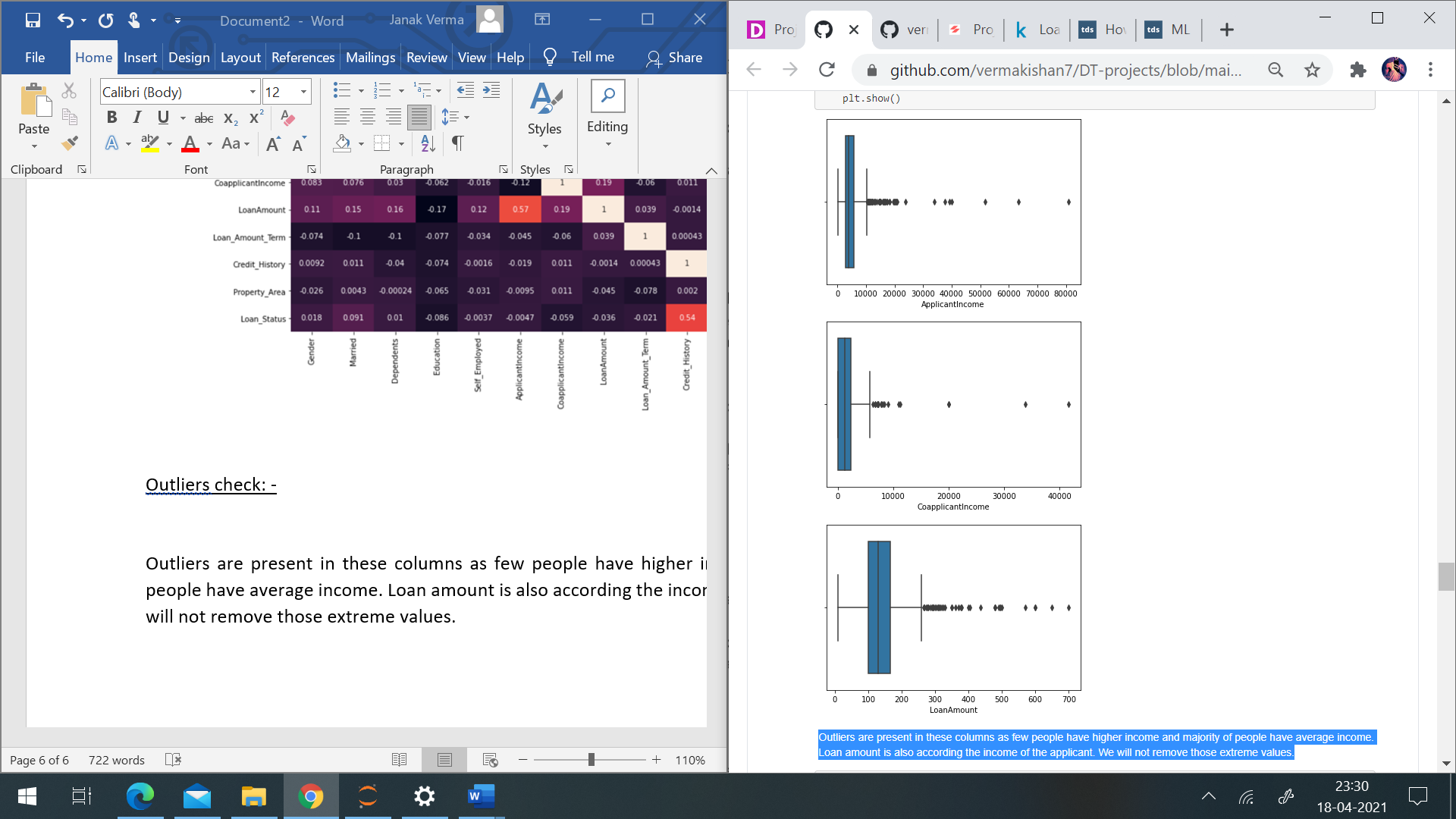
Correlation between variables: -

I’ve used a heatmap to visualize the correlation. Applicant income and loan amount are positively correlated. Loan status and credit history are positively correlated.

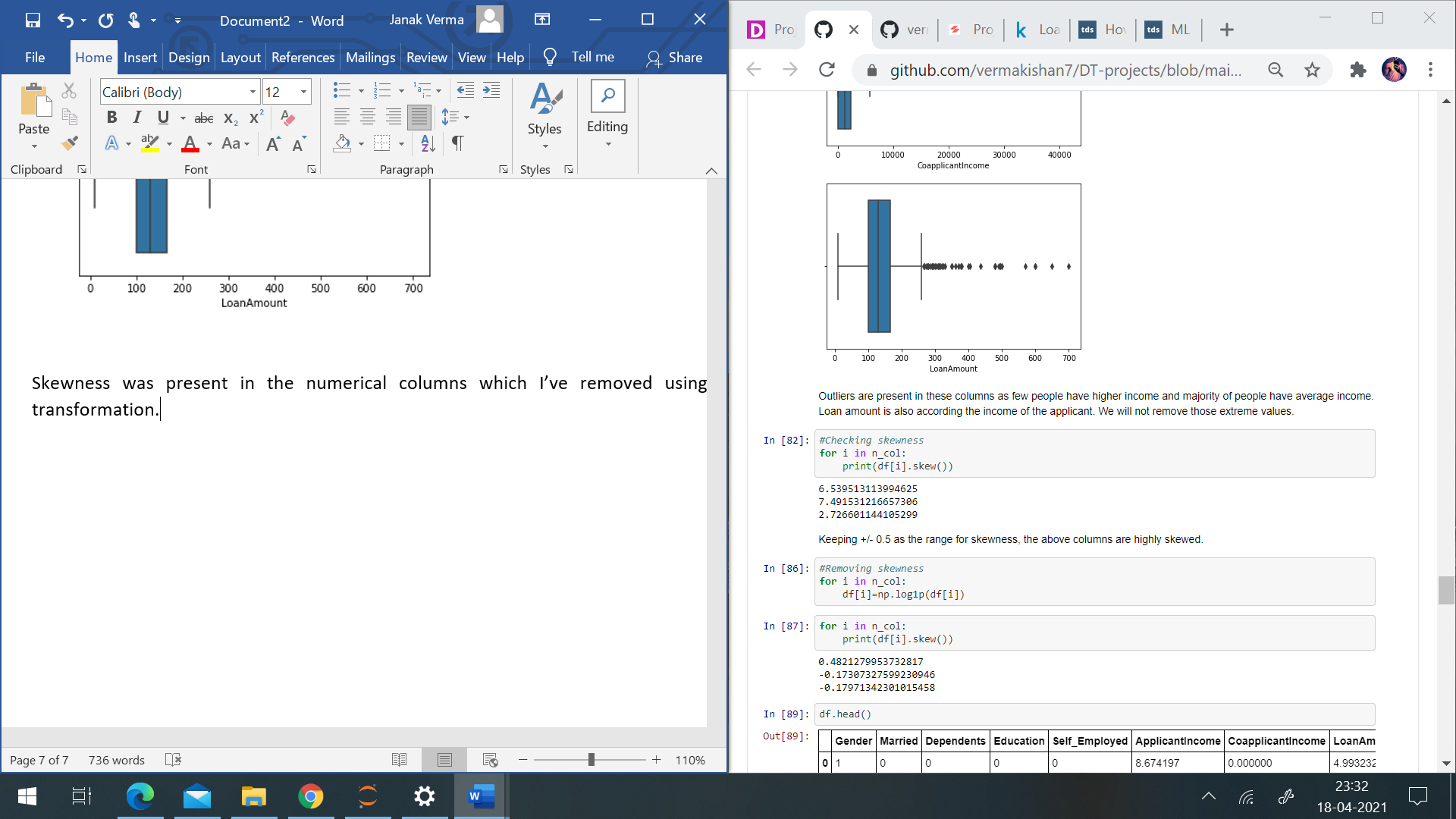


Outliers check: -

Outliers are present in these columns as few people have higher income and majority of people have average income. Loan amount is also according the income of the applicant. We will not remove those extreme values.



Skewness was present in the numerical columns which I’ve removed using the log transformation.



The data describe() function have given the following observations:

* The average applicant income is 5403 and maximum is 81000.
* Average coapplicant income is 1621 and highest is 41667.
* The loan amount ranges from 9 to 700.
* The loan amount term ranges from 12 to 480 months. And majority of applicants have opted for 360 months as 25%,50% and 75% all have 360.
* Average credit history is 0.85 it means that 85% of applicants have credit history equals to 1.

After EDA I’ve converted the object type columns to numerical using label encoder so that I can train my machine learning models.

**Building Machine Learning Model:**

Before building the ML model, first separate the independent variable and dependent variable as x and y. I’ve started by finding the best random state for the dataset, I’ll be doing the train test split at the random state on which highest accuracy is obtained. I’ve obtained an accuracy of 86% on random state 115. Now I’ve done train test split at random state 8. I’ve used the following classification models:

LogisticRegression

RandomForestClassifier

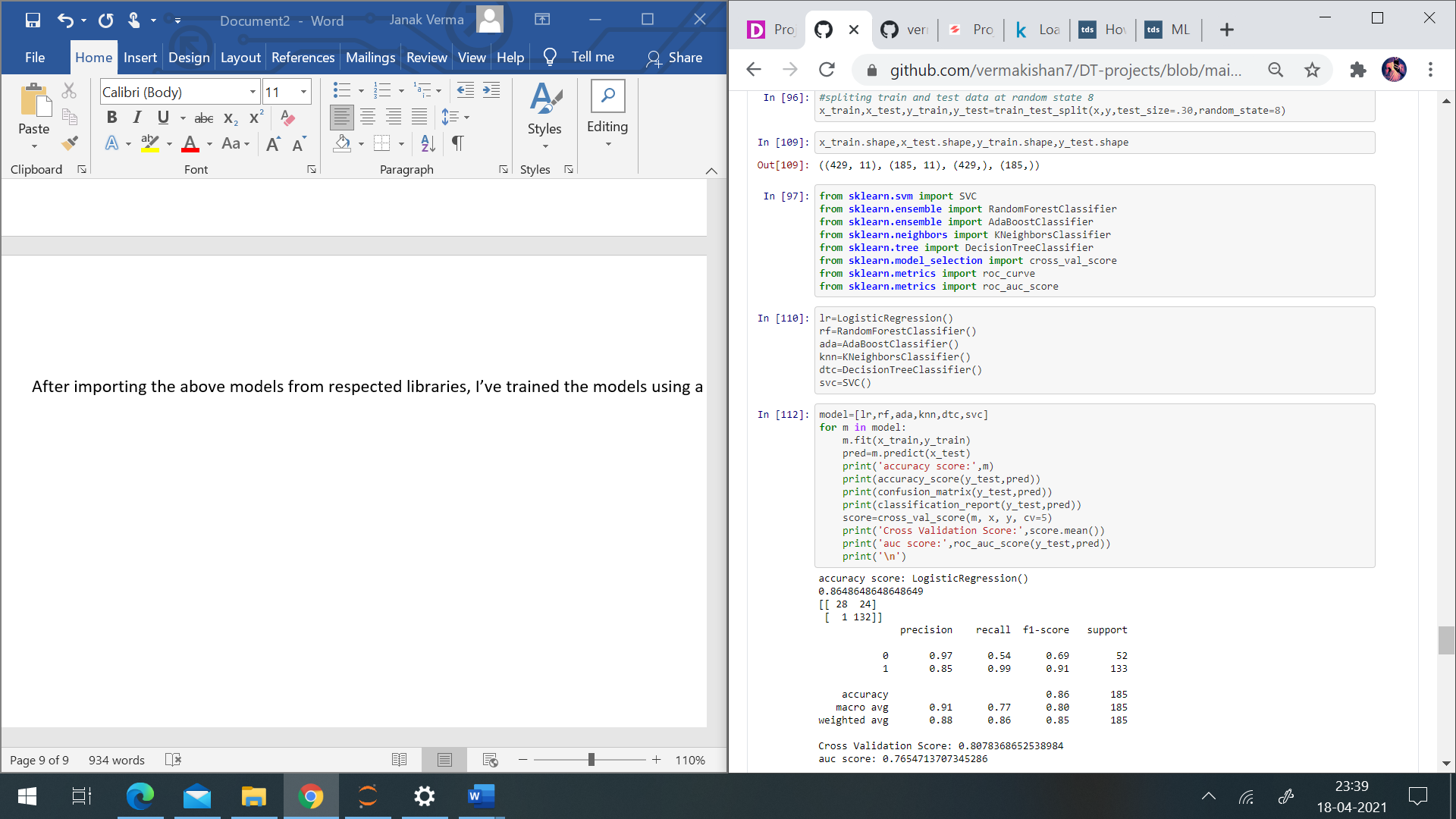
AdaBoostClassifier

KNeighborsClassifier

DecisionTreeClassifier

SVC

After importing the above models from respected libraries, I’ve trained the models using a for loop.



The result of the above loop is :

accuracy score: LogisticRegression()

0.8648648648648649

[[ 28 24]

[ 1 132]]

precision recall f1-score support

0 0.97 0.54 0.69 52

1 0.85 0.99 0.91 133

accuracy 0.86 185

macro avg 0.91 0.77 0.80 185

weighted avg 0.88 0.86 0.85 185

Cross Validation Score: 0.8078368652538984

auc score: 0.7654713707345286

accuracy score: RandomForestClassifier()

0.8324324324324325

[[ 30 22]

[ 9 124]]

precision recall f1-score support

0 0.77 0.58 0.66 52

1 0.85 0.93 0.89 133

accuracy 0.83 185

macro avg 0.81 0.75 0.77 185

weighted avg 0.83 0.83 0.82 185

Cross Validation Score: 0.7866453418632547

auc score: 0.754626951995373

accuracy score: AdaBoostClassifier()

0.8324324324324325

[[ 31 21]

[ 10 123]]

precision recall f1-score support

0 0.76 0.60 0.67 52

1 0.85 0.92 0.89 133

accuracy 0.83 185

macro avg 0.81 0.76 0.78 185

weighted avg 0.83 0.83 0.83 185

Cross Validation Score: 0.7801012928162068

auc score: 0.7604829381145171

accuracy score: KNeighborsClassifier()

0.7783783783783784

[[ 21 31]

[ 10 123]]

precision recall f1-score support

0 0.68 0.40 0.51 52

1 0.80 0.92 0.86 133

accuracy 0.78 185

macro avg 0.74 0.66 0.68 185

weighted avg 0.76 0.78 0.76 185

Cross Validation Score: 0.7100626416100226

auc score: 0.6643290919606709

accuracy score: DecisionTreeClassifier()

0.7189189189189189

[[ 28 24]

[ 28 105]]

precision recall f1-score support

0 0.50 0.54 0.52 52

1 0.81 0.79 0.80 133

accuracy 0.72 185

macro avg 0.66 0.66 0.66 185

weighted avg 0.73 0.72 0.72 185

Cross Validation Score: 0.7182460349193656

auc score: 0.6639676113360324

accuracy score: SVC()

0.7189189189189189

[[ 0 52]

[ 0 133]]

precision recall f1-score support

0 0.00 0.00 0.00 52

1 0.72 1.00 0.84 133

accuracy 0.72 185

macro avg 0.36 0.50 0.42 185

weighted avg 0.52 0.72 0.60 185

Cross Validation Score: 0.6872984139677463

auc score: 0.5

I’ve got the best results with logistic regression with accuracy score of 86%, cross validation score of 80% and auc roc score of 76%. F1 score for 0 is 69% and for 1 is 91%. Then I’ve performed hyper parameter tuning for the best model i.e., Logistic regression.

The results after hyper parameter tunning are:

The best parameters were: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}

accuracy score:

0.8648648648648649

[[ 28 24]

[ 1 132]]

precision recall f1-score support

0 0.97 0.54 0.69 52

1 0.85 0.99 0.91 133

accuracy 0.86 185

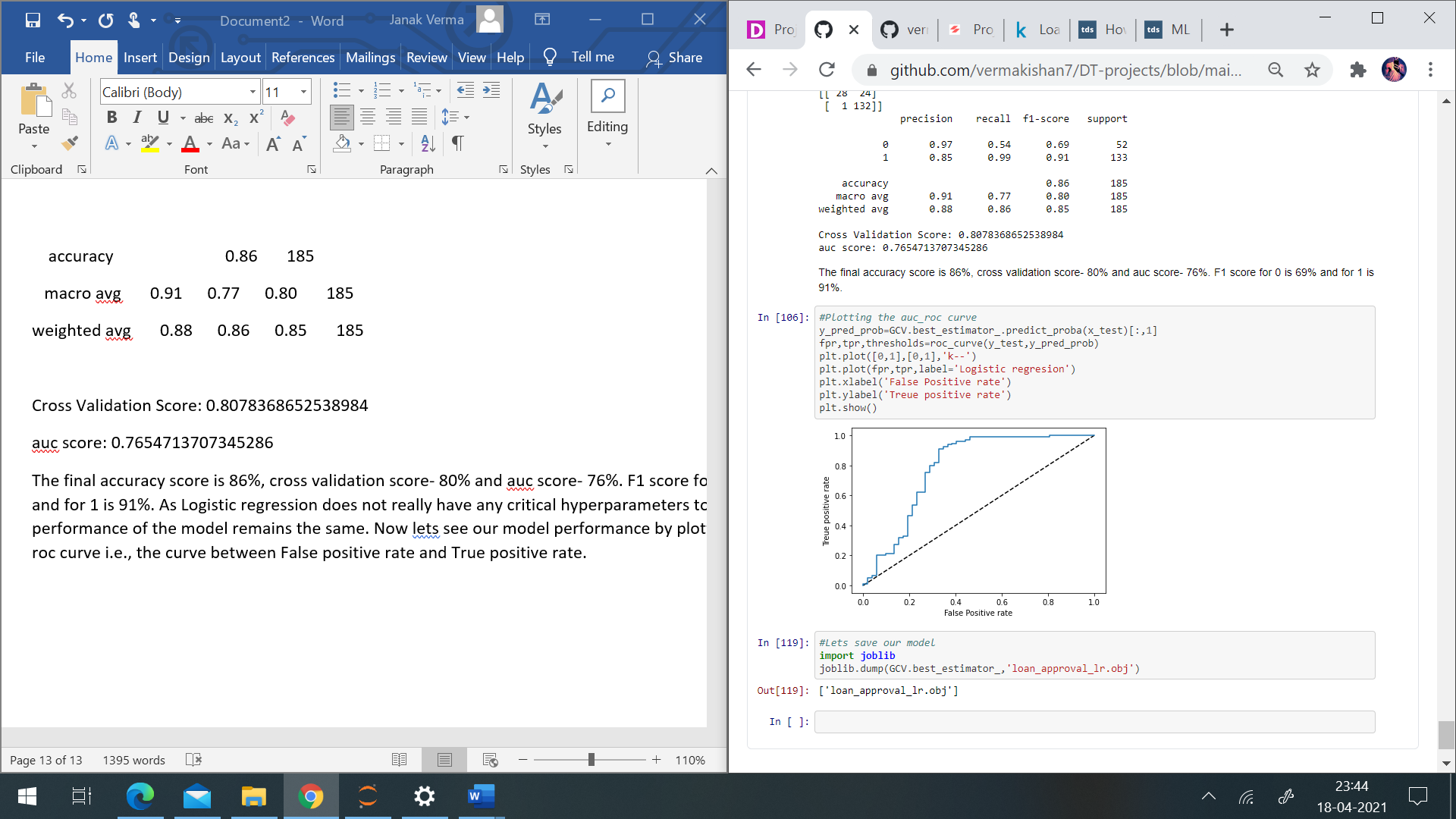
macro avg 0.91 0.77 0.80 185

weighted avg 0.88 0.86 0.85 185

Cross Validation Score: 0.8078368652538984

auc score: 0.7654713707345286

The final accuracy score is 86%, cross validation score- 80% and auc score- 76%. F1 score for 0 is 69% and for 1 is 91%. As Logistic regression does not really have any critical hyperparameters to tune, the performance of the model remains the same. Now lets see our model performance by plotting auc roc curve i.e., the curve between False positive rate and True positive rate.



**Conclusion:**

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. Logistic regression is an estimation of Logit function. The logit function is simply a log of odds in favour of the event. I have got an accuracy of 86 % which means our model have identified 86% of the loan status correctly. I got an auc roc value of 0.76 which is quite good for a model. This model will help banks to approve or reject the loan applications of the customers.