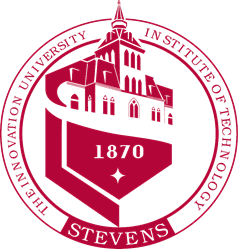
**PROJECT REPORT**

**FOR**

**LOAN PREDICTION USING PYTHON**



**Prepared by:**

Shrey Kshatriya, Hariom Mehta, Tigran Margaryan, Sagar Tanna

1. **Business Understanding:**

* **Identification of Problem:** An average more than thousands of loan applications received by a finance company in a single month. Usually it takes more than 3 to 4 months to review all the application which leads to delaying in roll out the decisions and that directly affects the customer trust and interest towards financial firm. Moreover, only 1/3rd applications from the total the number of loan applications are actually qualified candidates. But, most of the customer more likely to choose the other competitors because of the delay in decision. The amount of data provided by customer and time to analyse and make the decision for loan application is very long procedure which is not optimized and not efficient and time consuming.
* **Business Problem Definition:** Dream Housing Finance company provides all types of home loans. The organization is widely available nationwide, which includes all the urban, semi-urban and rural areas. Customers first apply for a home loan after which the company validates the customer eligibility for a loan. The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, we identify the customers’ segments, those which are eligible for loan amount so that they can specifically target those customers.
* **Motivation:** The main motivation of this is to automate the process of approving the loan. Loan prediction is a very common real-life problem that each retail bank faces in regular basis. If done correctly, it can save a lot of man-hours from a retail bank’s end and serve a greater number of customers in less span of time.
* **Addressing the business problem:** By using various data analytics approach, we can automate the whole decision process which filter out the best matching, well qualified candidates which are more likely to be consider for loan approval. Here are some important details usually provided by the applicant which is critical to determine the loan application decision: Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History. There are some constrains, rules and regulation to focus on which helps Dream Housing finance company to predict which applicant are more likely to get approval. If applicant has less income and the loan value is very high, then it will automatically send a rejection letter. This also helps to predict the applicant which matches all the conditions. This will reduce the overall amount of paperwork and man hours and resources. By applying data analytics in loan prediction company can save more time in processing the application and serve the greater number of customers in less time.

1. **Data Understanding:**

The dataset that has been chosen for this project is Loan Prediction problem Dataset that was chosen from Kaggle website. There were two datasets, which were the test dataset and the training dataset.

Since, the problem for the project was to predict whether to approve the loan of a customer should be approved or not, this dataset was a perfect option. Other datasets that were also considered for this project were, ‘Lending Club Loan Data’, ‘Bank Loan Status Dataset’ and ‘Bank Loan Modelling’.

The dataset that we selected contains data columns such as Loan ID, Gender, Married, Dependents, Education, Self-Employment, Income, Credit History, Loan Amount, Property History. This contains very helpful information as it indicates all the necessary features such as how much the applicant earns, whether they have any dependents or not, how good is their credit history along with other features. Using these features, it becomes easier to predict whether to provide loan to the application. Some threshold has been set which should be cleared by the applicant to get the loan approved or else, the loan would be rejected.

We have 12 independent variables and 1 target variable, i.e. Loan\_Status in the train dataset.

We can see there are three format 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.

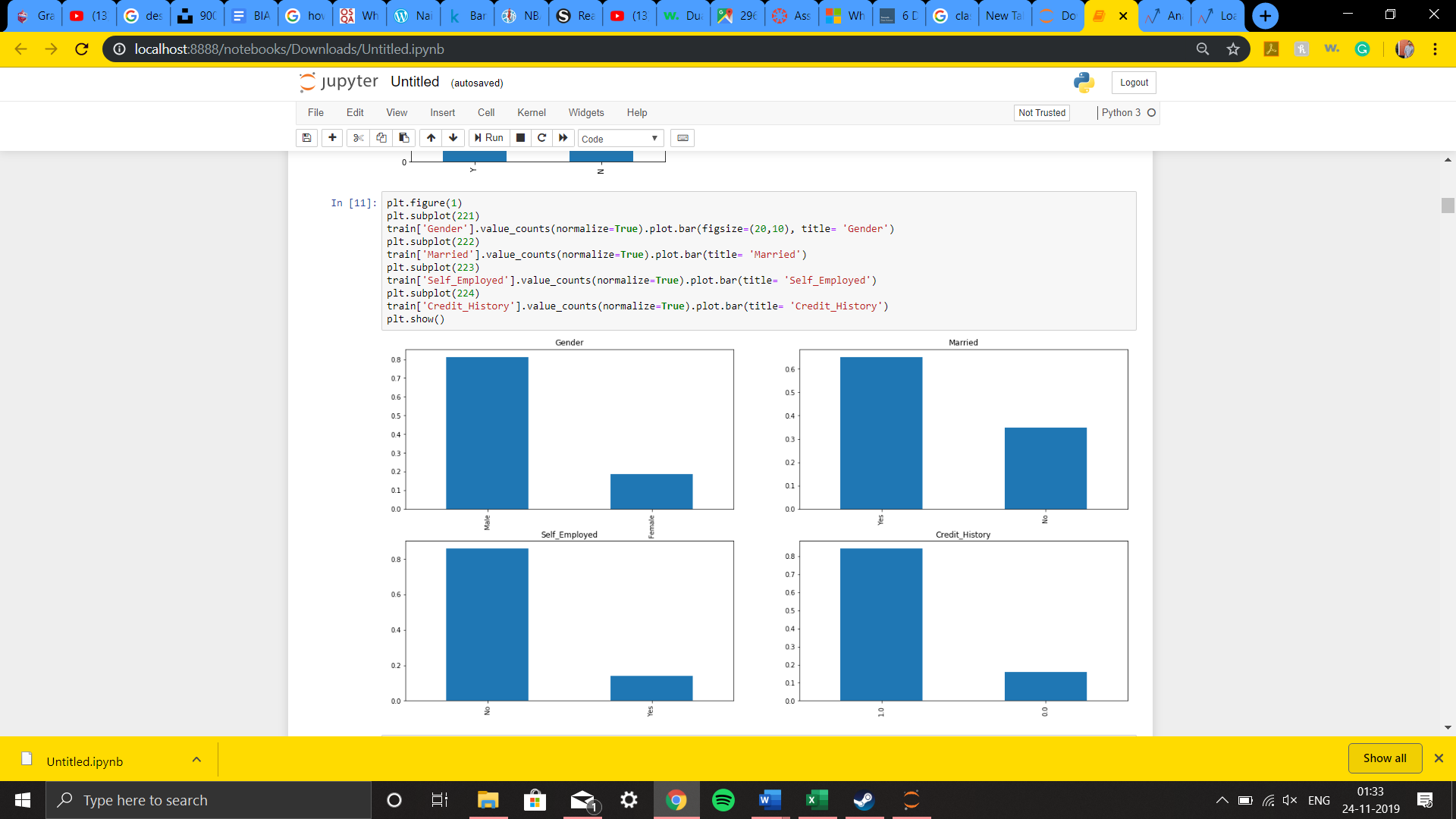
**float64:** It represents the variable which have some decimal values involved. They are also numerical variables. Numerical variables in our dataset are: CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, and Credit\_History.

The data has been read into a Panda data frame and then the Panda data frame was converted into a numpy array. There are also several important columns in the dataset such as the ‘Married’ column. The married column indicates whether the applicant might have a dependent. Then, the applicant’s income also plays an important role as it would give an understanding into if the applicant will be able to pay back the loan amount. Another important column is the credit history column. This column shows the how the applicant has managed their finances in the past and whether they have a good reputation or not. This is one of the most important columns as even if the applicant crosses all the thresholds, but has a bad credit score, even then the loan application can be rejected.

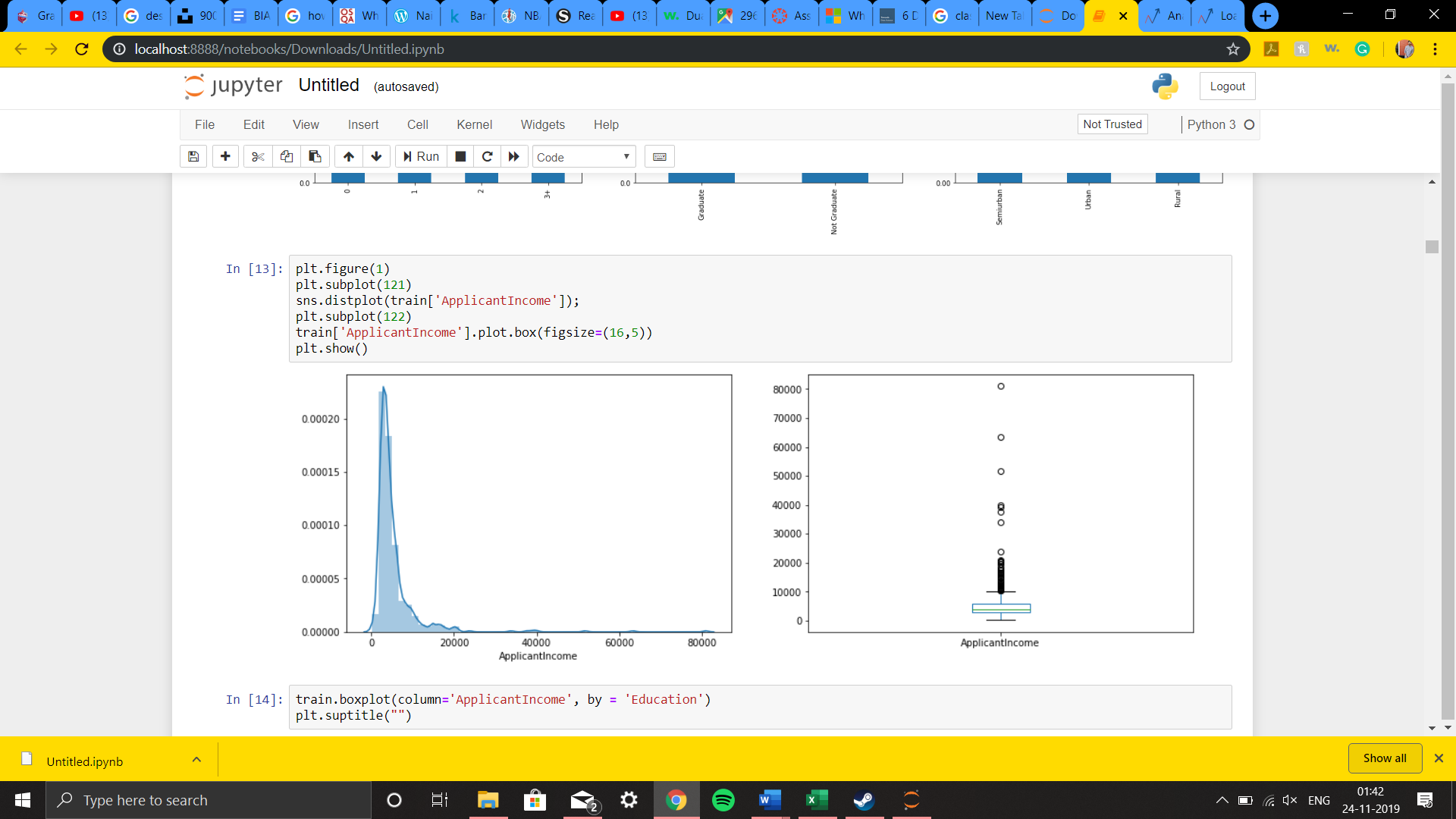
There have been many anomalies in the data which have been cleaned. In some places where the column had missing values, but was not an important column, then it was removed. In other places where the column was important, code imputation was performed.

To understand the data better, we plotted graphs which give us better understanding of the dataset and the attributes.

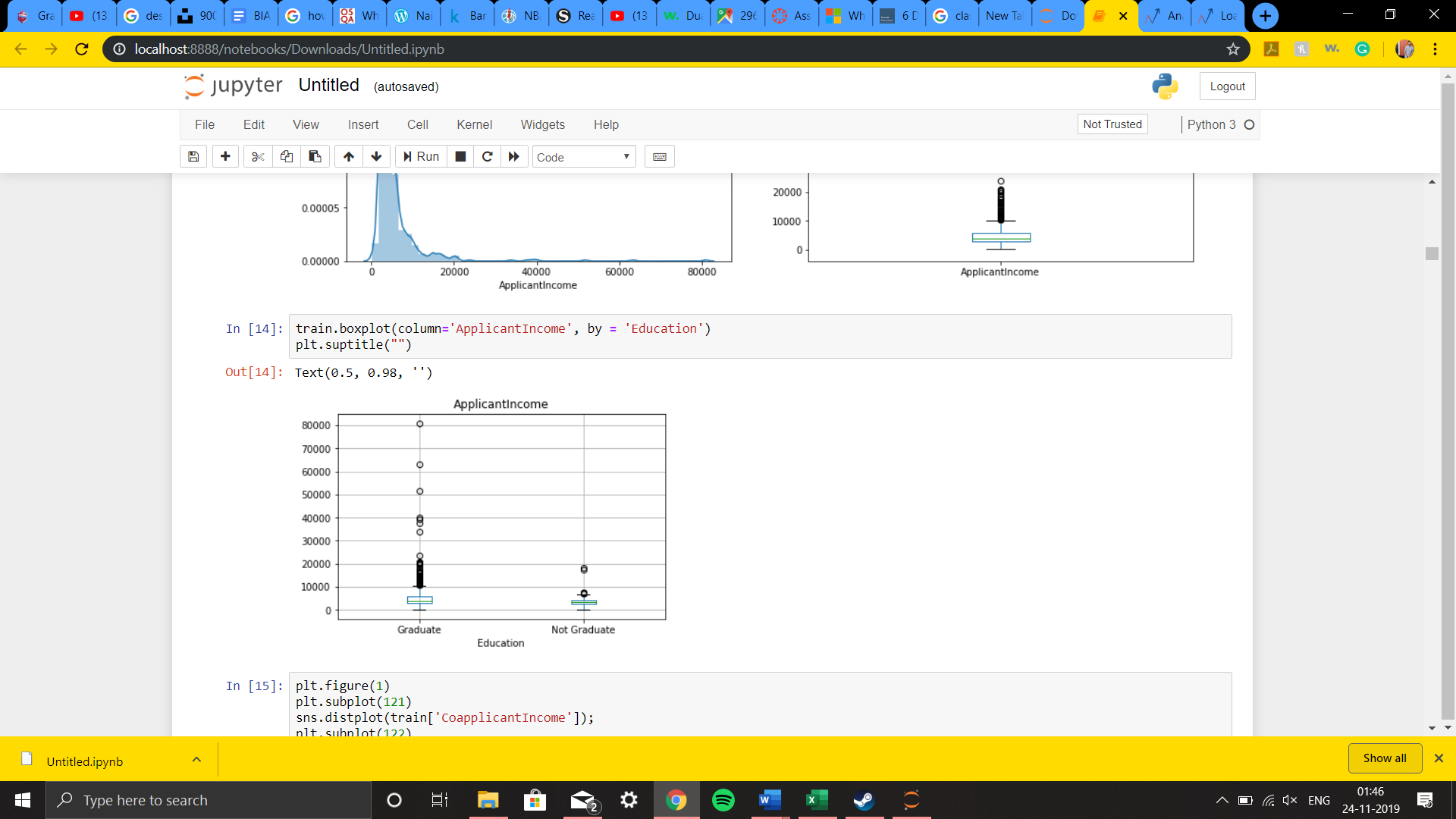
Plotting graphs, scatterplots and heatmaps help us to understand dependencies and correlation between the data in the dataset. This can be used to predict whether or not the loan to an applicant should be granted or not.



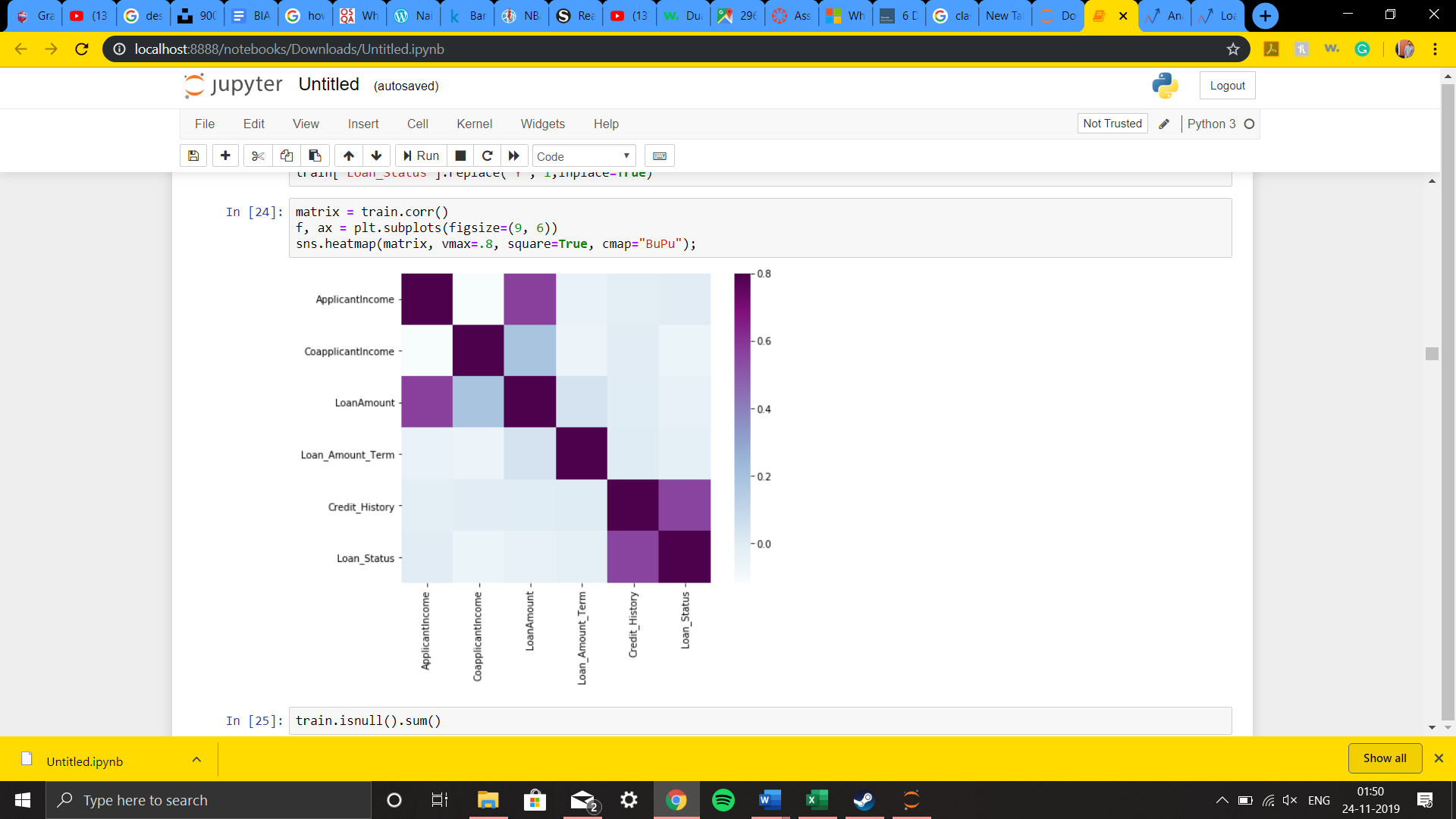
This graph shows the columns against the attributes of the column. Such as the number of Male and Females in Gender, the number of people who have 1.0 credit history and those who have 0.0 credit history.



In the above plot, we plot the applicant income in a distributed plot and then the applicant income is shown in a BoxPlot.



The above graph is a BoxPlot that compares the number of applicants that are Graduates and those who are Not Graduates.



The above graph is a heatmap, that shows the correlation between the attributes of the dataset. This is important as once the correlation is known, this can be used to predict the data. This helps us to understand the dataset even better and model it in a way, in which it can be predicted correctly.

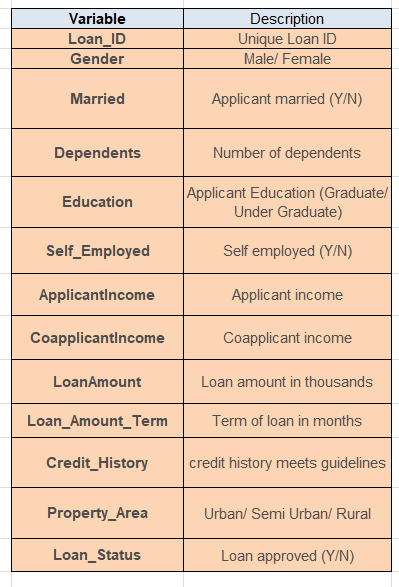
1. **Data Preparation:**

Data preparation is the process of cleaning and transforming the raw data prior to processing and analysis. It is an important step prior to processing and it often involves reformatting or normalizing the raw data, making corrections to data and the combining of data sets to enrich data. It is often a very lengthy process, but it is essential as a prerequisite to put data in context in order to turn it into insights and eliminate bias resulting from poor data quality.

Good data preparation allows for efficient analysis, limits errors and inaccuracies that can occur to the data during processing and makes all the processed data more accessible to users.

Following are the steps involved in the preparation of data:

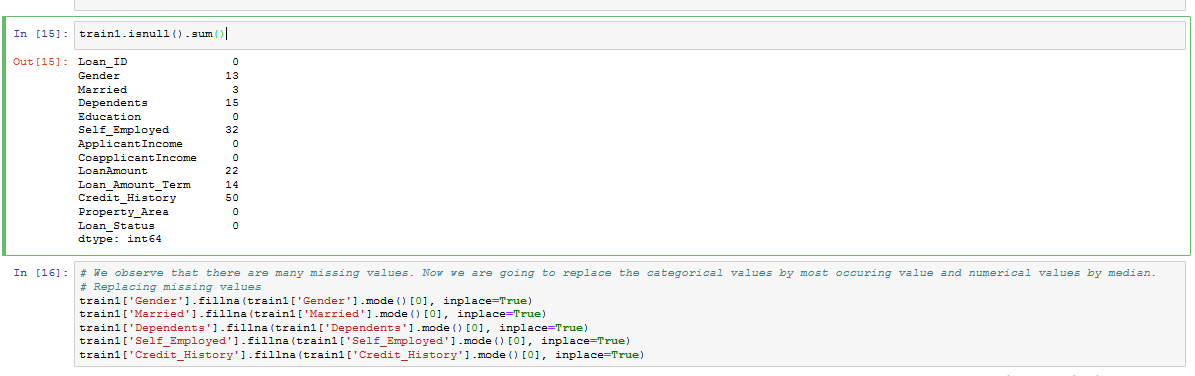
* **Gathering data:** The data preparation process begins with finding the right data. Gathering the data involved getting together and finalizing not only the topic we wanted to work on but also the type of problem we were interested to work on for example, if we wanted to do a problem related to regression or solve a classification problem. After the first few initial meetings, we came to a consensus of doing a loan prediction problem and looked for existing datasets available online and gathered datasets related to our project idea.
* **Discover and access the data:** After collecting the data, it is important to discover each dataset. It is about getting to know the data and understanding what all needs to be done before the data becomes useful in a particular context. After finalizing on the dataset, we have a look at the variables the datasets consist of. There are 2 data sets; one is training data and the other one, the test data. There are altogether 13 columns in our data set. Of them Loan\_Status is the response variable and rest all are the variables /factors that decide the approval of the loan or not. After understanding the data columns, we summarize all the 13 columns in a table for better understanding.



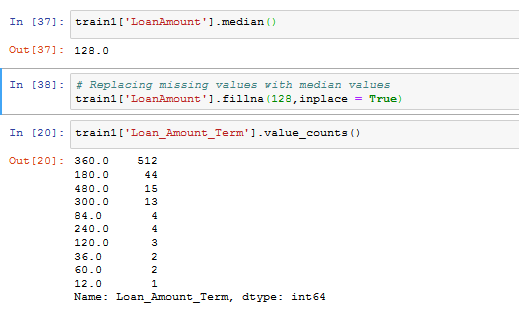
* **Cleansing data:** Cleaning up the data is traditionally the most time-consuming part of the data preparation process, but it is equally crucial to remove faulty data and filling in the gaps. Important tasks here include:

1. Removing extraneous data and outliers.
2. Imputing or filling in the missing values.
3. Conforming the data to a standardized pattern that is easy for computation.

For cleaning part, we first check if there exist any missing values. For that we use the code snippet isnull().

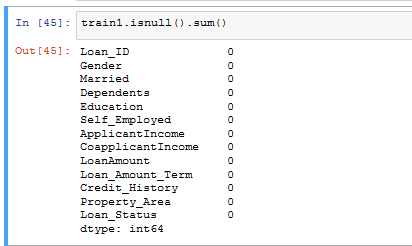


Based on the above screenshot, we can see that there are 13 missing values in Gender, 3 in Married, 15 in Dependents, 32 in Self\_Employed, 22 in Loan Amount, 14 in Loan\_Amount\_Term and 50 in Credit History. Except the Loan Amount and Loan\_Amount\_Term everything else which is missing is of type categorical. Hence we can replace the missing values by mode of that particular column. Replacing the categorical variable by mode makes some sense. For example if we take the above case, 398 are married, 213 are not married and 3 are missing. So as married people are higher in number we are considering the missing values as married. This may be right or wrong. But the probability of them being married is higher. In the case of continuous variables, replacing the missing values by the median makes more sense than replacing by mean because it is largely affected by outliers.



In the above code, missing values of Loan-Amount is replaced by 128 which is nothing but the median.

* **Transforming and Enriching of data:** Transforming data is the process of updating the format or the value entries to extract a well-defined outcome or making the data more easily understood by a wider audience. After performing all the data cleaning procedures, we check if there are any more missing values by using the following code snippet train1().isnull().sum()



Enriching the data refers to adding and connecting the data with other related information or creating new variables out of the current ones in order to provide deeper insights. In our dataset, we have two columns named applicant income and co-applicant income. It may be the case that total income might have a great impact on Loan Status. It may or may not work. Also, It may be the case that EMI would have a greater impact on Loan Status as it combines Loan Amount and Loan Amount Term. So, we introduce these new variables by extending the already available data columns that may help to make our model better.

* **Storing data:** Once prepared, the data can be stored or channelled into for example a data frame using pandas and numpy libraries to be able to store and perform analysis and extract meaningful insights from it and provide results that can help solve our problem.

1. **Modelling:**

Our project is a classification problem where we must predict whether a loan would be approved or not. In a classification problem, we must predict discrete values, based on a given set of independent variables.

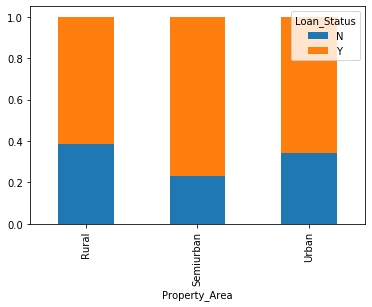
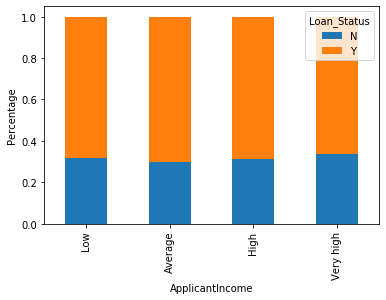
After looking at the problem statement, we moved into the hypothesis generation and listed out all the possible factors that can affect the outcome.

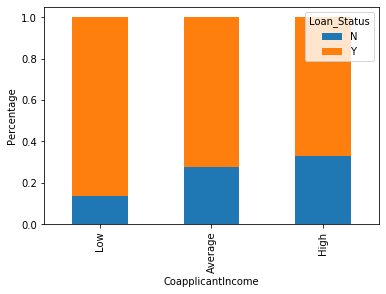
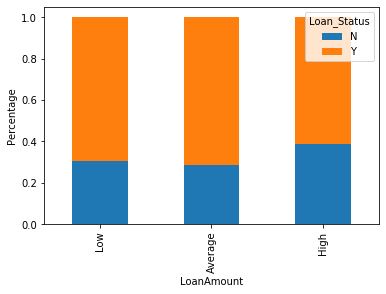
Below are some of the factors which might affect the Loan Approval (dependent variable or this loan prediction problem):

* Salary: Applicants with high incomes should have more chances of loan approval.
* Previous history: Applicants who have repaid their previous debts should have higher chances of loan approval.
* Loan amount: Loan approval should also depend on the loan amount. If the loan amount is less, the chances of loan approval should be high.
* Loan term: Loan for less time period and less amount should have higher chances of approval.
* EMI: Lesser the amount to be paid monthly to repay the loan, the higher the chances of loan approval.
* Property location: If the applicant is in an urban location, his/her application has a higher chance to get approved.
* Education: Graduates have a higher chance of approval.
* Dependents: If the applicant is earning, it increases the chance of getting approved, otherwise, the more dependents there are without earnings, the less the chance of approval.

After plotting the graphs and understanding which components are affecting the approval, we have found out that applicant income, co-applicant income, loan amount, loan amount term, property area, self-employment, as well marital status do not have significant effect on the loan status.

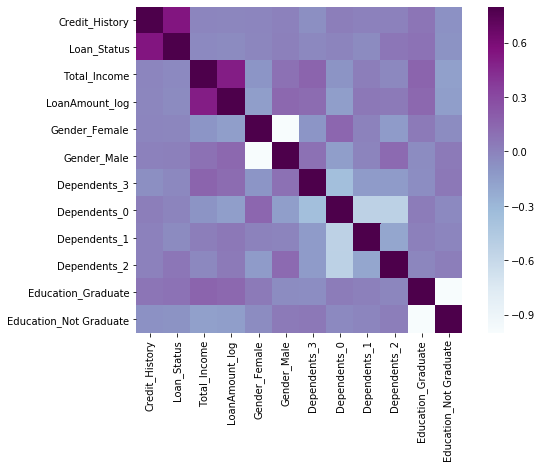
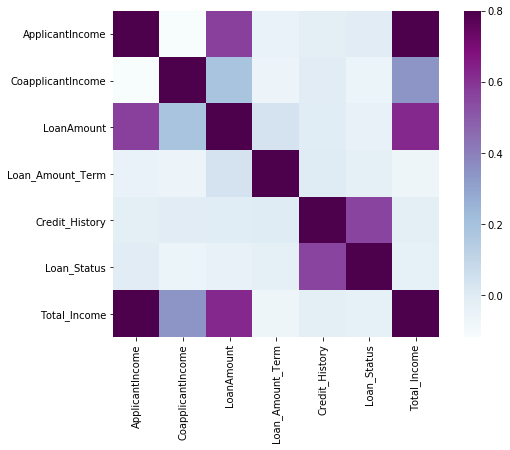
 

Eventually, we have decided to drop the unnecessary variables. We did the same changes to the test dataset which we did for the training dataset.

Then, as we have two columns named applicant income and co-applicant income, it may be the case that total income might have a great impact on Loan Status. So, we decided to use some common sense to find a new variable that can impact the outcome. Hence we created a new variable:

Total Income = Applicant Income + Co-applicant Income

This is how our matrix looks like after we dropped those columns and created the new variable.



According to the second heatmap, we have relatively better picture in terms of those variables which affect the loan decision making process the most.

Since we used to scikit-learn (sklearn) for making different models, which requires the target variable in a separate dataset, we dropped our target variable from the train dataset and saved it in another dataset.

Next, we made dummy variables for categorical variables, which make them a lot easier to quantify and compare. After doing so, we trained the model on the training dataset and made predictions for the test dataset.

After that, we need to validate these predictions, as well. To do so, we divided our train dataset into two parts: train and validation. We trained the model on this train part and using that make predictions for the validation part. In this way, we validated our predictions as we have the true predictions for the validation part (which we do not have for the test dataset). We used the train\_test\_split function from sklearn to divide our train dataset.

Having the possible scenarios in mind, we made our first model in order to predict the target variable. We started with Logistic Regression for predicting the binary outcome.

But first we needed to estimate the profit and loss amounts. After some brainstorming, we decided that “true not re-payed” loss should be 70, false not re-payed loss should be 0, false re-payed loss should be 70, and true re-payed profit should be 100.

Since the dataset has been already divided into the training and validation part, we imported Logistic Regression and accuracy\_score from sklearn and fitted the logistic regression model.

Having the logistic regression running successfully, we moved over to more complex models like Random Forest, Neural network, Support vector machine with linear kernel and (radial basis function kernel), k-nearest neighbour, Naive Bayes', Decision Tree classifier, and Random Forest classifiers.

After plotting the ROC and AUC curves, it becomes clear that logistic regression is doing relatively better job at classifying the loans. We have got an accuracy score of around 82%, however there is still a lot of room to enhance accuracy which we will try to figure out still.

Below are the accuracy scores of other models:

* The accuracy of Naive Bayes: 81
* The accuracy of Support vector machine with rbf: 68
* The accuracy of k-nearest neighbor: 65
* The accuracy of Decision tree: 79
* The accuracy of Random forest: 78

1. **Evaluation:**

Discuss how the results of your analytics are/should be evaluated. How should a business case be developed to project expected improvement? ROI? If this is impossible/very difficult, explain why and identify any viable alternatives.

1. **Deployment:**

Discuss how the result of the data analytics will be deployed.

Discuss any issues the rm should be aware of regarding deployment.

Are there important ethical considerations?

Identify the risks associated with your proposed plan and how would you mitigate them.