CS4055 – Data Mining and Data Warehousing

Group2

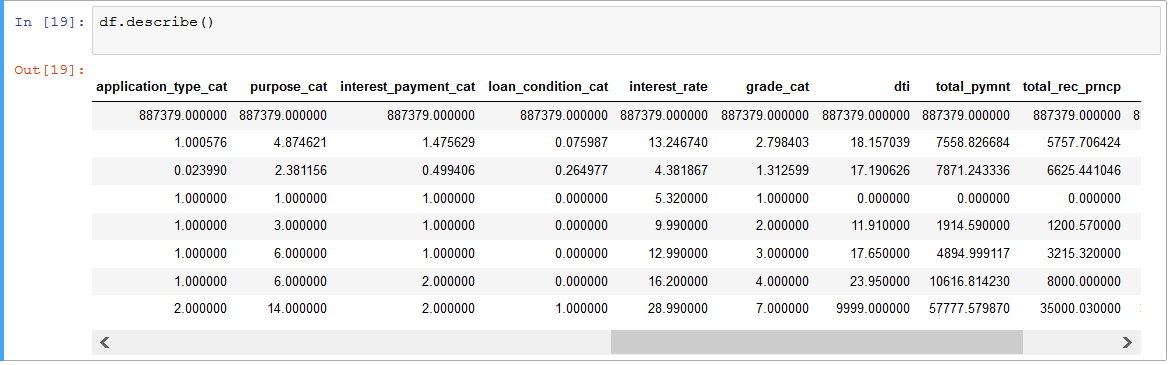
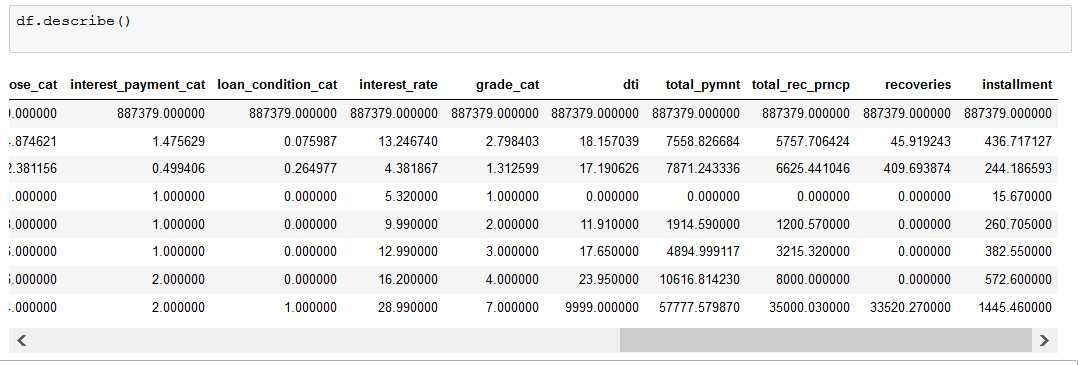
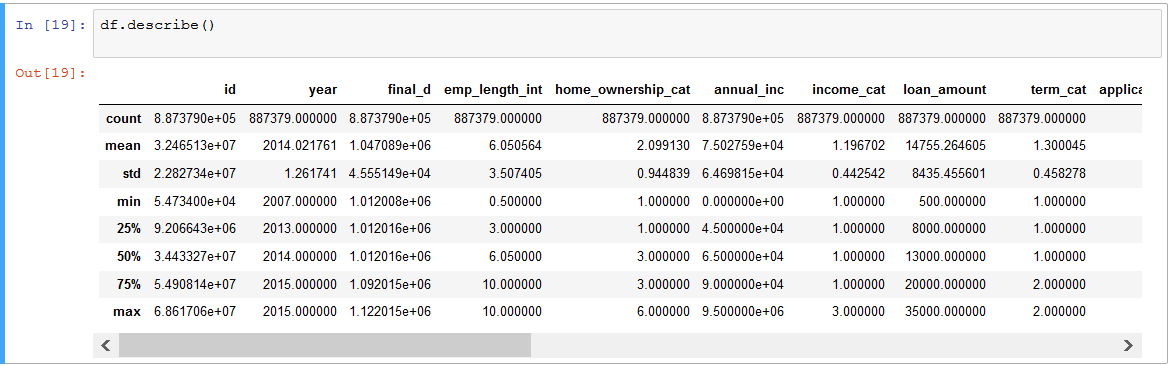
15169235 – David Kearney

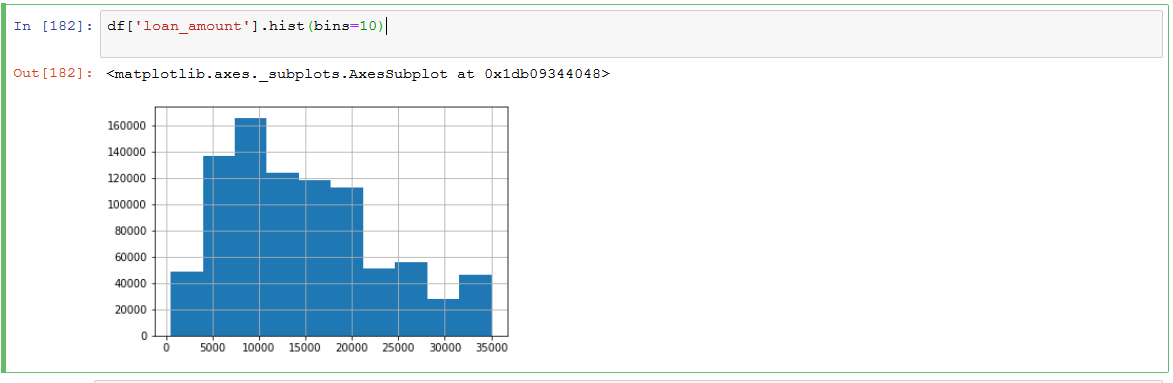
15170756 – Michael Keegan

12130583 – Gavin Randles

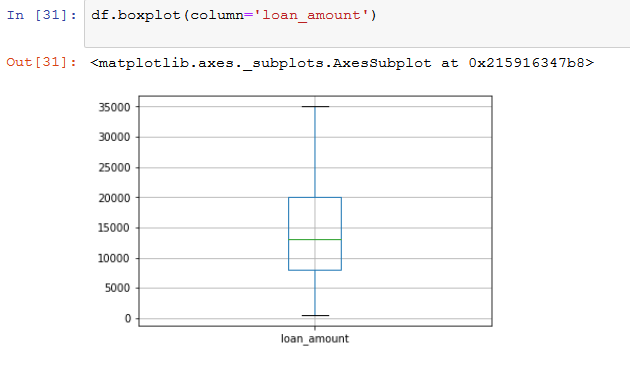
15145271 – Sean Wright

Section A -Sean Wright

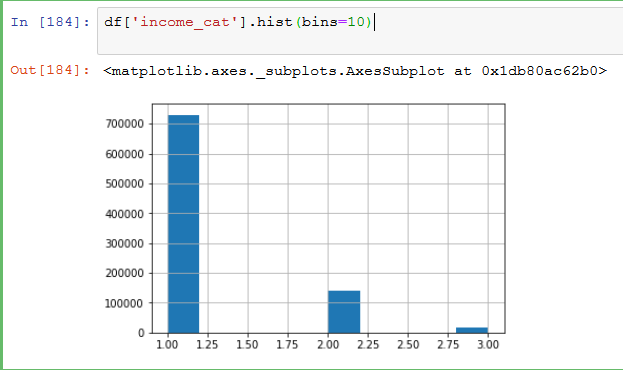
I used the df.describe() function to describe the data set were using this shows count, mean, standard deviation (std), min, quartiles and max in its output, the output of this can be seen below.



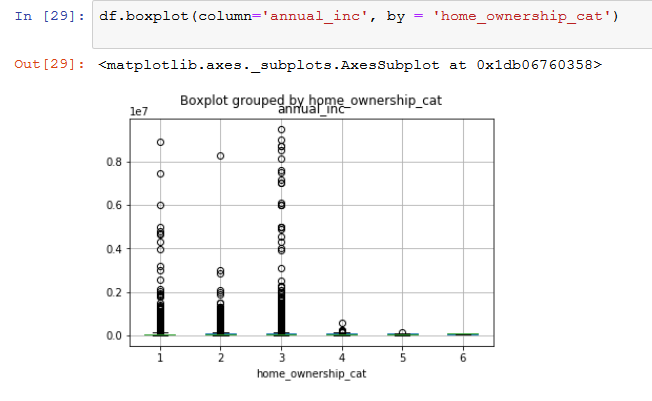
Shows the most frequency of the loan amounts in this data set we can see here that most loans are around the 10000.



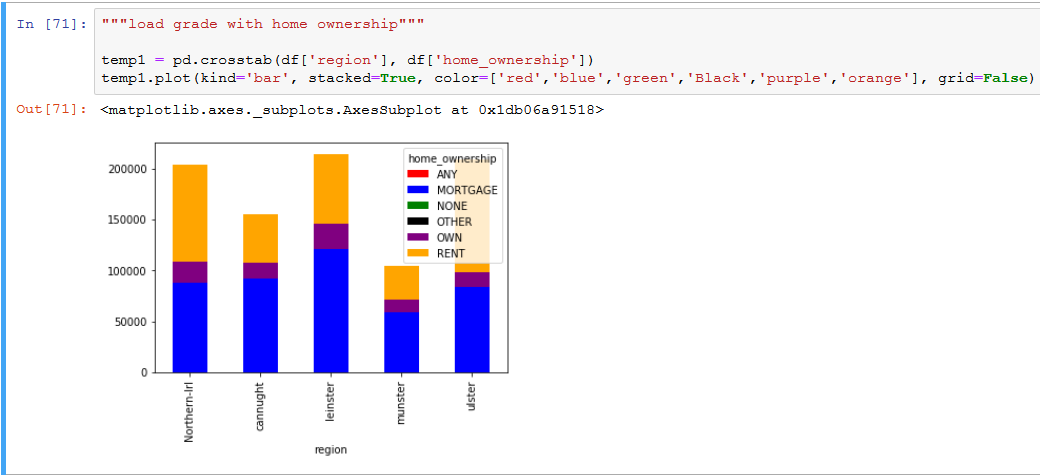
This boxplot reflects the same mean as what we can see in the histogram above also showing the upper and lower bound of the data set.



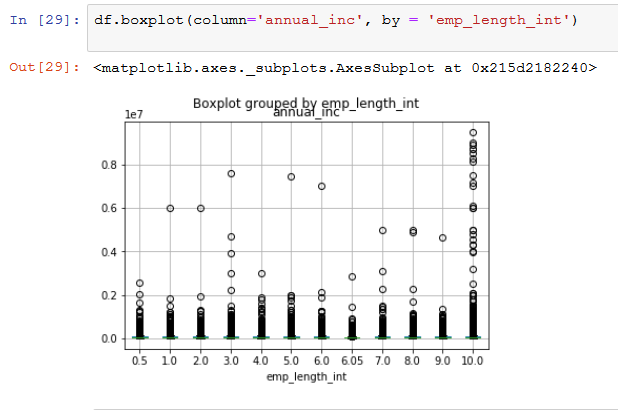
This histogram shows the frequency of the income of the loan holders we can see most are in the low income category



This boxplot shows the how annual income effects home ownership in the data set.

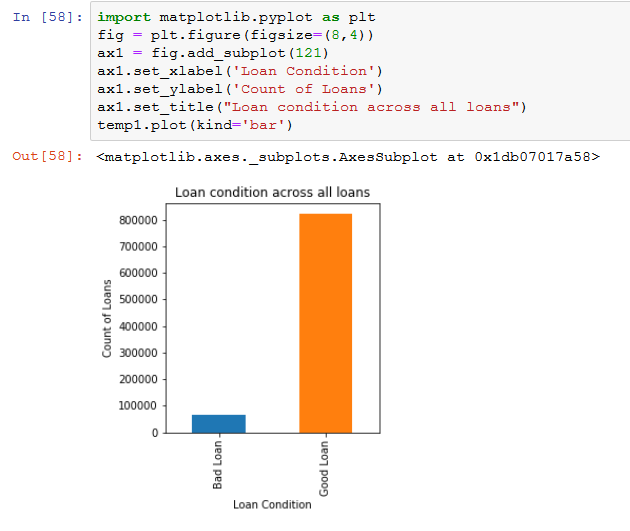


Here I wanted to continue to the description of home ownership, I wanted to examine if there is any correlation between home ownership and the region.

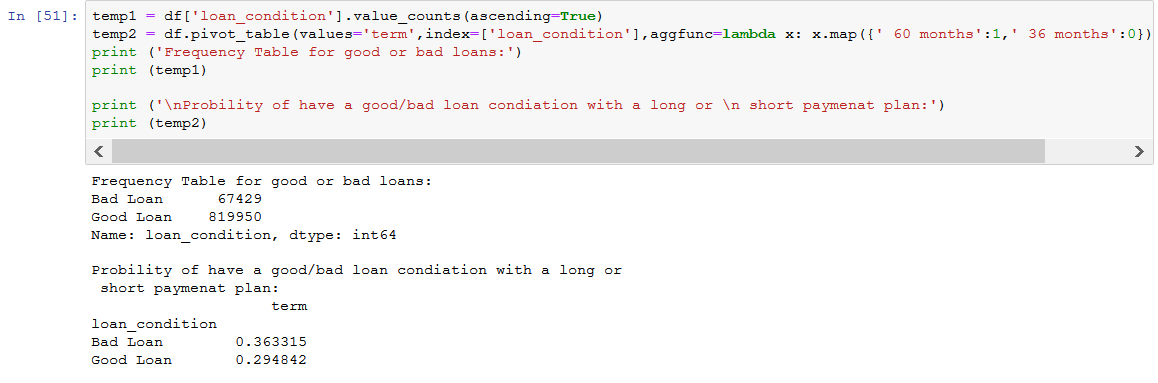


To continue getting some insight in this data set I wanted to examine the correlation between employment lent and annual income this boxplot shows a clear relationship between the two columns.

From here on I wanted to start taking a more in depth look at some of the information in this data set, I choose to look at Loan Condition.

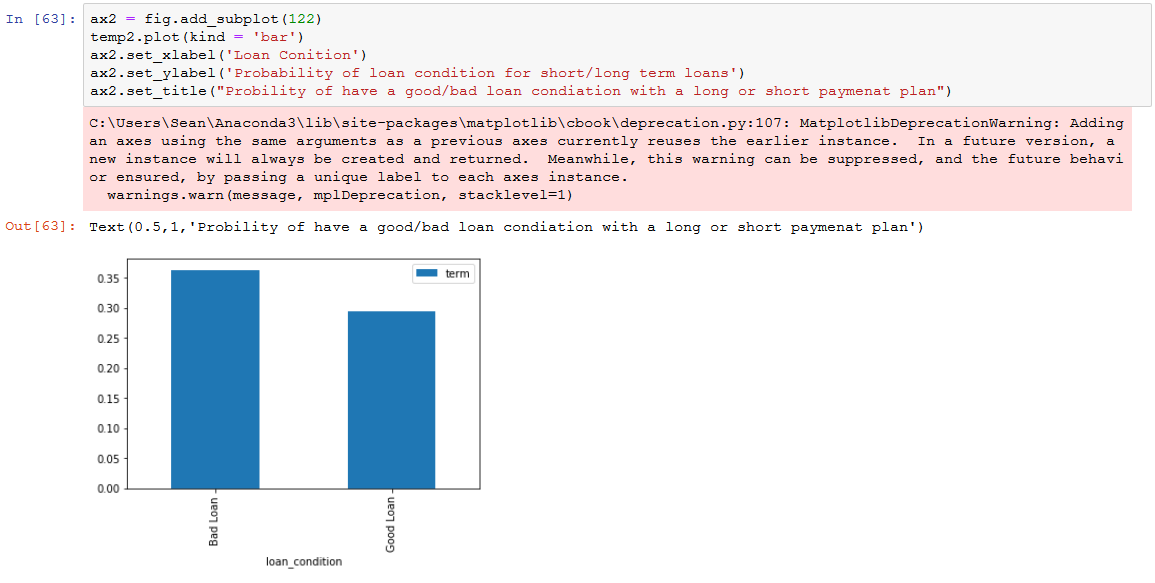


This bar chart shows the frequency of Good and Bad loans



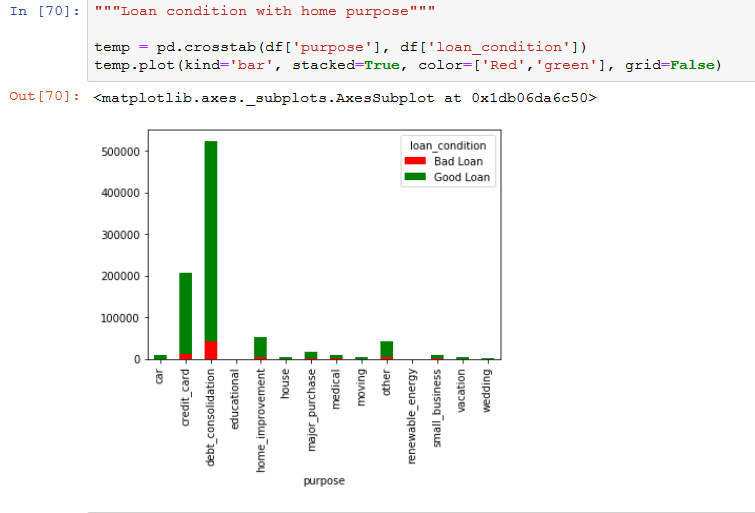
This is a more in-depth description of how a loans conidiation is effected by a loan being long or short term.

This is the above probability in a bar chart.

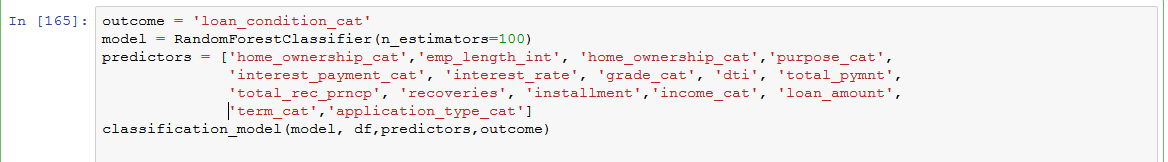


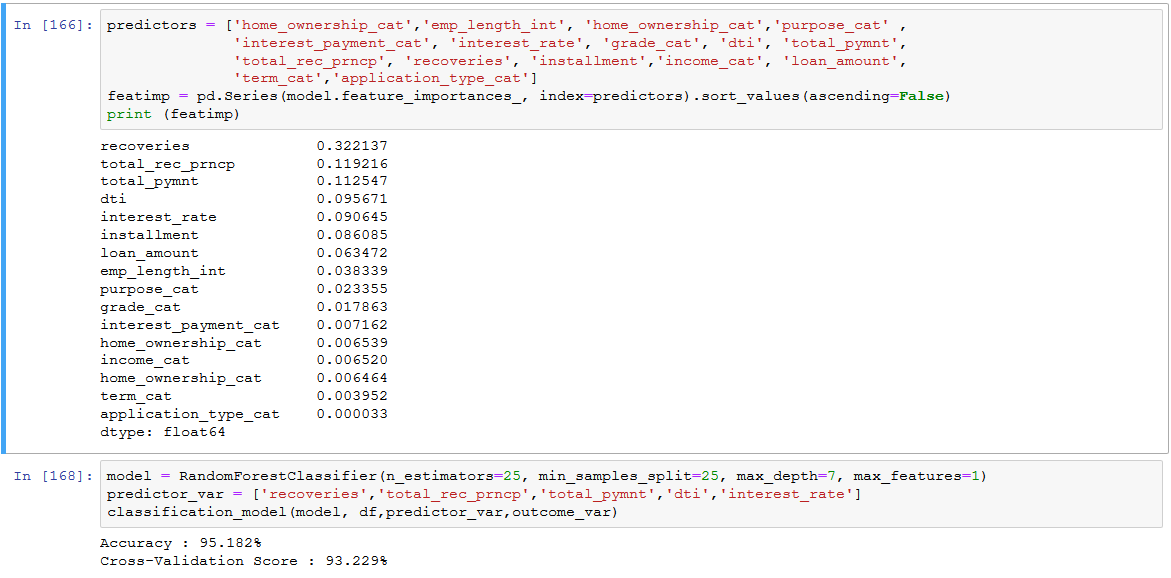
The last way I described loan condition is with the purpose of the loan to see what most of the loan were taken out for, we can clearly see the most common loan purpose was for debt consolidation.

We can also see that regardless of the purpose we had most loans had a good condition.



Although my main focus was on describing the data set I also worked closely with MJ on the ML algorithm, the result we found most interesting as what the Random Forest Classifier returned when we searched for the best predictors for loan\_condition\_cat, we found the same top five in a different order then when we used these as predictors on the loan condition we both had very high accuracy and high cross-validation score(see below).



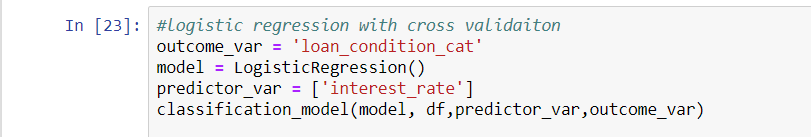


Michael Keegan – Linear Regression, Logistic Regression, Decision Trees, Random Forest

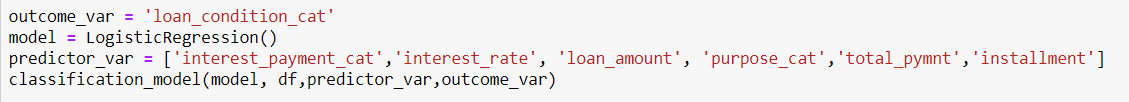
Since our dataset has been set up for modelling I decided to create a predictive model on our data set. I checked which columns were numeric and used these columns in my study. Since sklearn requires no missing values in the dataset I checked for missing values, luckily there wasn't any null values. I defined my classification function which is an algorithm that maps input data to a category.

I recorded the Accuracy and Cross-Validation scores using 10 folds.

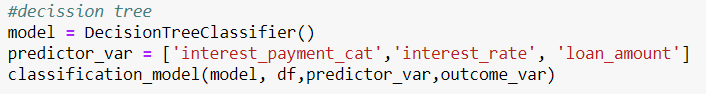
I started my Jupyter project making a logistic regression model on ‘loan\_condition\_cat’ (loan condition category). Logistic Regression is used to describe data and to explain the relationship between one dependent binary variable and one or more other variables. The loan condition category shows a list of good and bad loans. The more bad loans the less chance of a loan being given, and vice versa. Therefore, my first model was on loan condition. My Accuracy was: 92.401% and my Cross-Validation was: 92.401%.



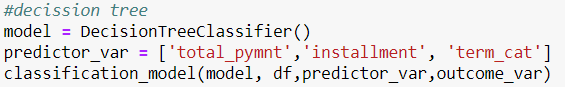
Adding more variables should increase accuracy, I tested this with less important variables and it only changed my results slightly for accuracy: 92.504% and cross-validation: 92.504% .



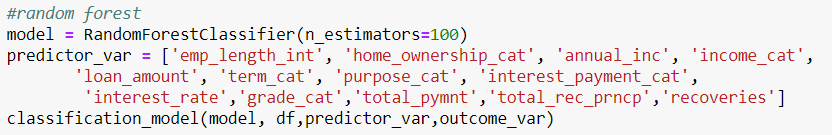
Feature engineering where I could derive new information could change these results or using different modelling techniques. Using a decision tree where my predictor variables were (interest payment cat, interest rate and loan amount) I managed to get an accuracy of: 93.254% and a cross validation score of: 91.081%.



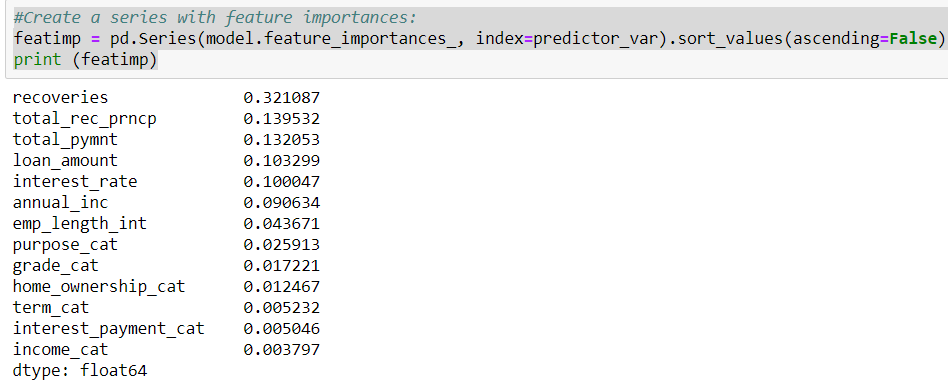
It can be seen here my accuracy went up but by cross-validation went down. I tried this again using different variables (total payment, instalment and term category) and my results were as shown: accuracy: 99.157% and a cross validation score of: 58.025%. This is the result of model over-fitting the data. Over-fitting is a modelling error which occurs when a function is too closely fit to a limited set of data points.

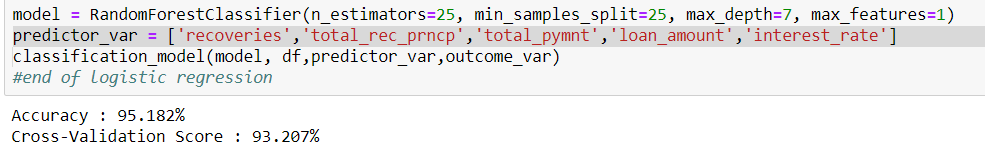


So for my next step I used Random forest ML algorithm which is even more sophisticated than my previous decision trees. Results for my random forest were as follows: accuracy: 99.988% with a cross validation of: 91.678%, giving us a better result than the decision trees. Seeing as the accuracy is 99.988% this is very close to the ultimate case of over fitting and can be resolved by reducing the number of predictors.

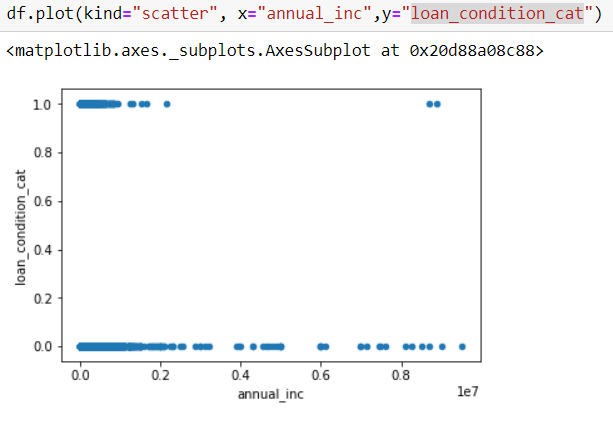


However to reduce the number of predictors we need to know which ones to use that are most important to us, we can see this by using a matrix importance table and taking the top 5 and creating a model. Results of using the top 5 more important predictors gave an accuracy of: 95.182% and a cross validation of: 93.207%. Although accuracy reduced the cross validation score improved meaning that the model is generalizing well.

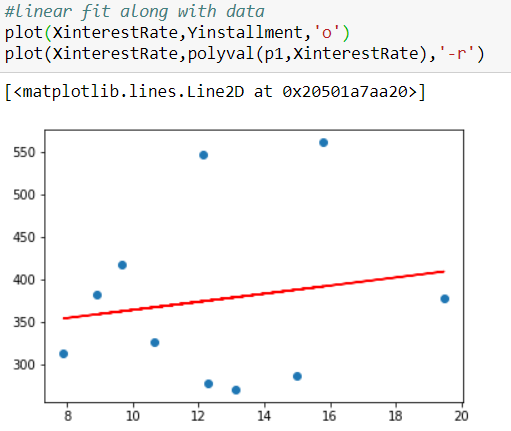




I continued my project by using linear regression on the loan dataset. I made a scatter plot with loan condition category against annual income as a test, as you can see in my Jupyter notebook this was not very useful as loan condition isn't really effected by or relevant to a persons annual income.



I then spit my dataset by taking 5% of the dataset without replacing anything. I took random selected values from interest rate and instalment and plotted them on a graph to show fit, intercept and slope of data.



Gavin Randles - KNN

I decided to use the k-nearest neighbours(kNN) algorithm because it is used to discover useful patterns and correlations within the data. I felt like the Dataset we choose was sparse in terms of available columns to use with kNN . I was unable to perform the algorithm on the numeric columns and some of the object columns couldn’t be used be the data had no direct correlation with other columns. In the end I believe I successfully performed the kNN algorithm on the “grade”, “income\_category” and “loan\_condition” columns. Please view results in my corresponding Jupyter Notebook.