

**MICRO-CREDIT DEFAULTER PROJECT**

Submitted by:

Mohd Tarab

**ACKNOWLEDGMENT**

I'd like to express my heartfelt gratitude to the Flip Robo for providing a detailed description of the project while also answering questions. My subject matter expert (SME) or mentor, Shubham Yadav, who assisted me in the formation of this project where I was stuck with some problems that were cured by my SME.

I am grateful to Data-Trained Education for the valuable advice, suggestions, and experience they provided me during the training period; it is only because of them that I was able to complete this Project.

There were some errors and problems that occurred during the project solution that I was able to correct with the help of the internet or webs such as Kaggle and GitHub, among others.

**INTRODUCTION**

* Business Problem Framing

Companies are having difficulty selecting their customers based on their ability to repay the loan amount within 5 days of loan issuance. Loan amount = 5 and 10 Indonesian Rupiah on mobile balance, payback amount = 6 and 12 Indonesian Rupiah This is a problem that many general companies face because the loan amount is not always recovered because these companies do not properly select their customers, as well as whether or not they are or will be able to repay their loan amount

Microfinance companies make small loans; however, most people forget to repay their small loan amount to the company because nothing bad will happen to them with such a small amount. As a result of these factors, our world now has more than $ 70 billion Outstanding Loans.

* Conceptual Background of the Domain Problem

A Microfinance Institution (MFI) is a business that provides financial services to low-income people. Microfinance services (MFS) are especially useful when targeting unbanked low-income families living in remote areas with few sources of income. MFIs offer a variety of MFS such as Group Loans, Agricultural Loans, Individual Business Loans, and so on.

Many microfinance institutions (MFI), experts, and donors support the concept of using micro financial services (MFS), which they believe are more convenient, efficient, and cost-effective than the traditional high-touch model used for delivering microfinance services for a long time. Despite the fact that the MFI industry is primarily focused on low-income families and is very useful in these areas, MFS implementation has been uneven, with both significant challenges and successes.

* Review of Literature

As per my observation and understanding of the project, it has been elevated that the organisation has been experiencing a problem with predicting their customers, as there has been nearly 12.5 percent of the data we have of defaulters who are unable to pay the loan amount of 6 or 12 within 5 days of issuance.

According to my observations and understanding of the project, it has been raised that the organisation has been having difficulty predicting their customers, as nearly 12.5 percent of the data we have of defaulters who are unable to pay the loan amount of 6 or 12 within 5 days of issuance.

* Motivation for the Problem Undertaken

The motivation and goal of this project are to work on the data using all of the techniques and improve the model results so that our model predicts a higher accuracy score. This is due to the fact that our clients will be able to predict whether their customers will be defaulters or non-defaulters.

This project focuses on providing services and products to low-income families or poor customers, which aids in defining these customer references of whether or not to repay the loan.

**Analytical Problem Framing**

* Mathematical/ Analytical Modelling of the Problem

1. Client - Indonesian Telecom Industry

2. A microfinance institution (MFI) is a non-profit organisation that provides financial services to low-income individuals and families.

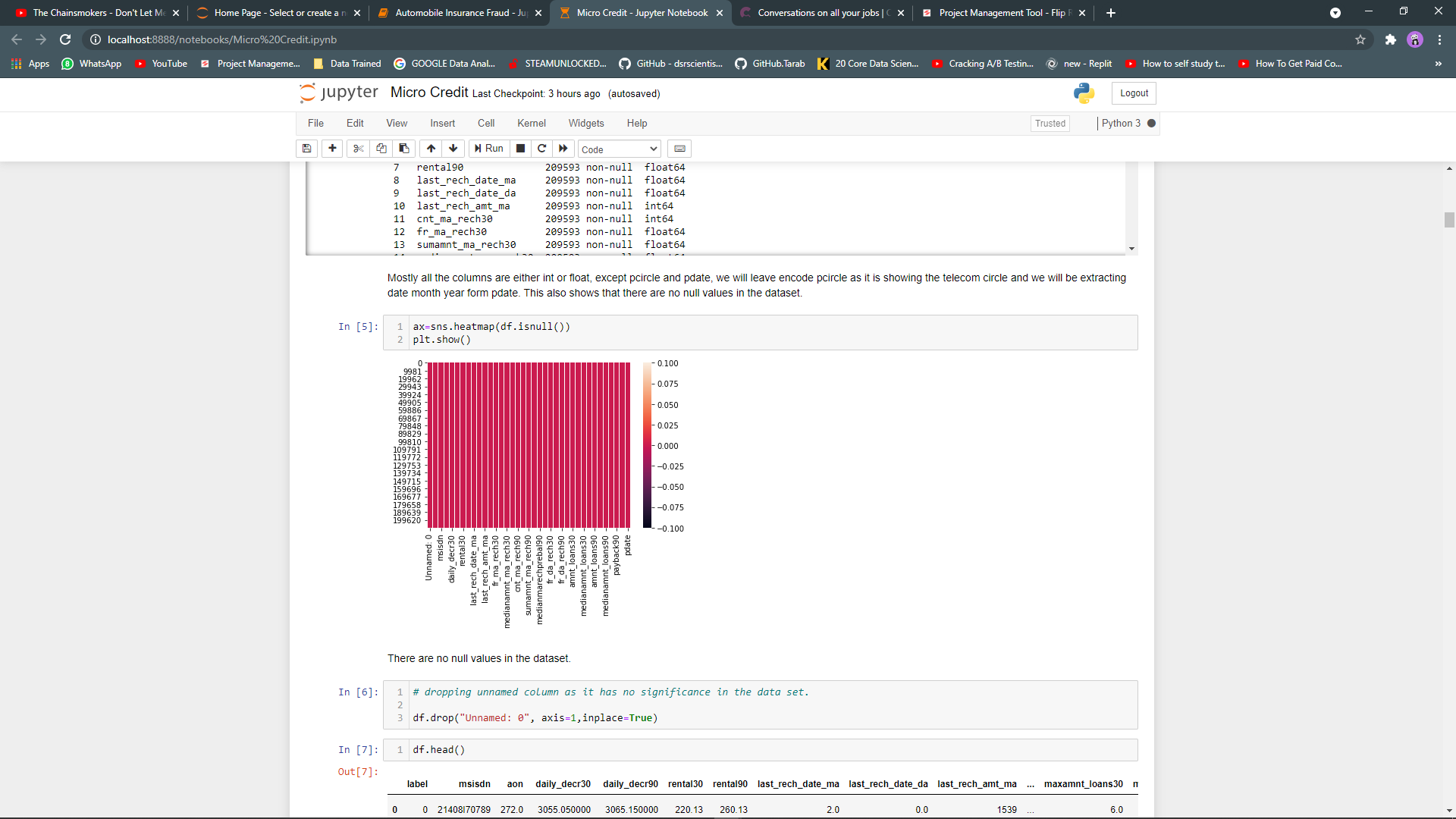
3. They are collaborating with a microfinance institution to provide microcredit on mobile balances.

4. These credits must be repaid within 5 days or he or she will be considered a defaulter.

5. There are two loan amounts: 5 and 10 Indonesian Rupiah, with payback amounts of 6 and 12 Rupiah.

6. The data set has 209593 rows and 37 columns, with a column with the feature name Unnamed: 0 that is removed because it is useless in predicting the target variable.

7. The dataset does not contain any null values.



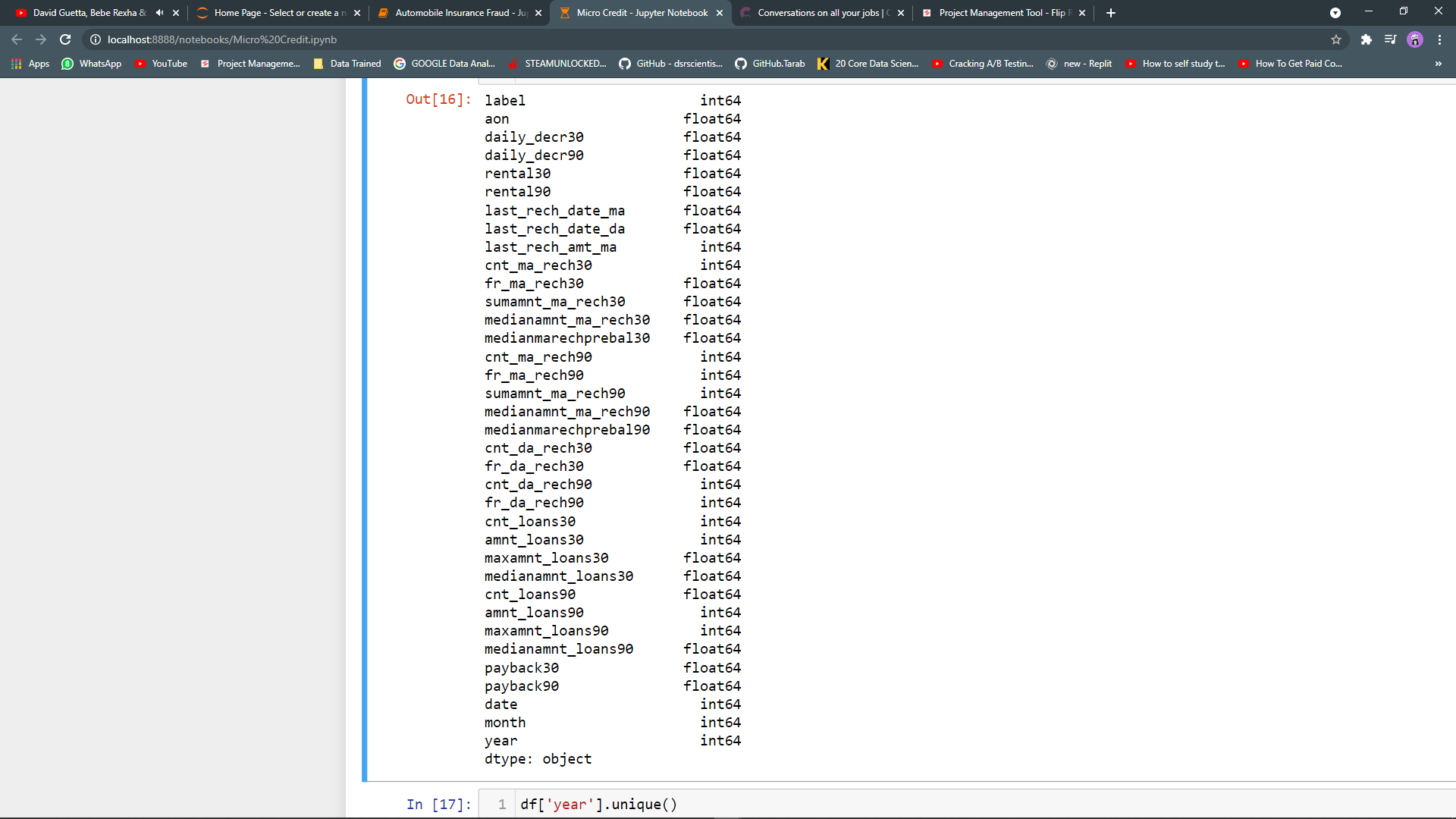
The above chart illustrates that if there are any null values in the above dataset, I discovered that the red colour represents the '0' value, which indicates that neither of the values are empty (Null Values).

8. Target variable is “Label” 1 = Non-Defaulters with 87.5% records, 0= Defaulters with 12.5% records.

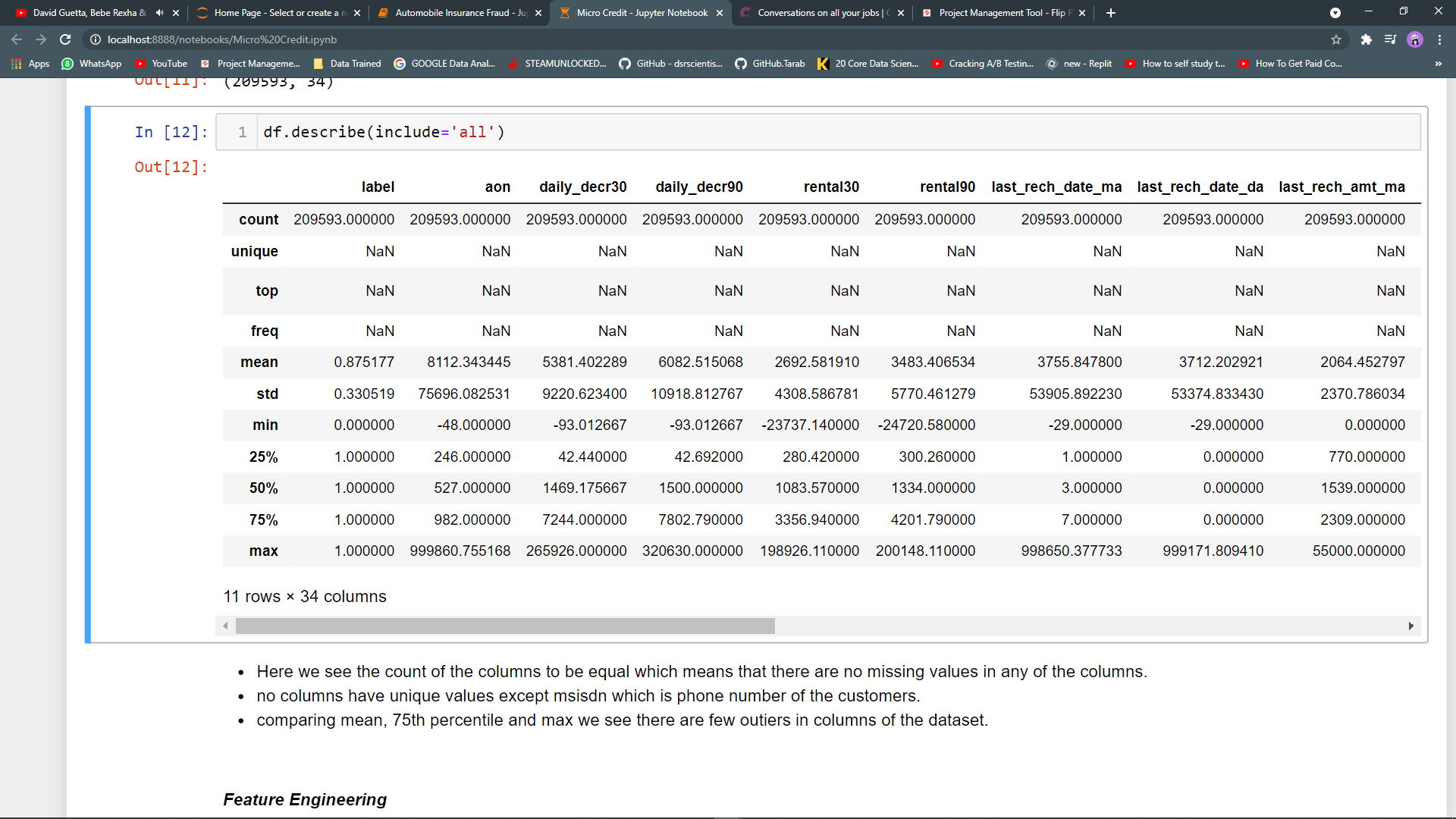


* Data Sources and their formats

1) Data types are int64(12), object(1), float(21), datetime64(1).



Data Description:



This states that mean, min, max, standard 25 percent, and 75 percent, for example, taking into account the aon, which is the age on the cellular network in days with a count of 209593, where the mean or average days is 8112. The customer specifies a minimum of -48 days and a maximum of 999860 days. It also states the 25%, which is 246 and the 75%, which is 982. As we can see, the third quartile is greater than the mean of the data, indicating that the dataset is continuous. As we can see, the third quartile is greater than the mean of the data, indicating that the dataset contains outliers that must be treated. 50 percent is nothing more than the data feature's median, which is 527; similarly, we could detect other variables such as rental 30 or last rech date da, among others.

* Data Pre-processing Done

Treatment of the Outliers Data:

For treatment of outliers, we cannot lose data as the company advised us not to, of that I will cap outliers with the maximum value of each columns.

And not treating the target column and categorical columns

#checking outliers

for i in df.columns[1:]:

print(i)

print(df[i].plot.box())

plt.show()

#function for detect Outliers when feature has skewness (by IQR)

def outlier\_IQR(data\_frame,feature\_name):

IQR = data\_frame[feature\_name].quantile(0.75)-data\_frame[feature\_name].quantile(0.25)

lower\_bridge = data\_frame[feature\_name].quantile(0.25)-(IQR\*1.5)

upper\_bridge = data\_frame[feature\_name].quantile(0.75)+(IQR\*1.5)

return (lower\_bridge,upper\_bridge)

for i in df.columns[1:]:

print(i)

print(outlier\_IQR(df,i))

#making function to treat outliers

def out(data\_frame,feature\_name):

IQR = data\_frame[feature\_name].quantile(0.75)-data\_frame[feature\_name].quantile(0.25)

upper\_bridge = data\_frame[feature\_name].quantile(0.75)+(IQR\*1.5)

lower\_bridge = data\_frame[feature\_name].quantile(0.25)-(IQR\*1.5)

data\_frame.loc[data\_frame[feature\_name]>=upper\_bridge,feature\_name] = upper\_bridge

data\_frame.loc[data\_frame[feature\_name]<= lower\_bridge,feature\_name] = lower\_bridge

for i in df.columns[1:]:

out(df,i)

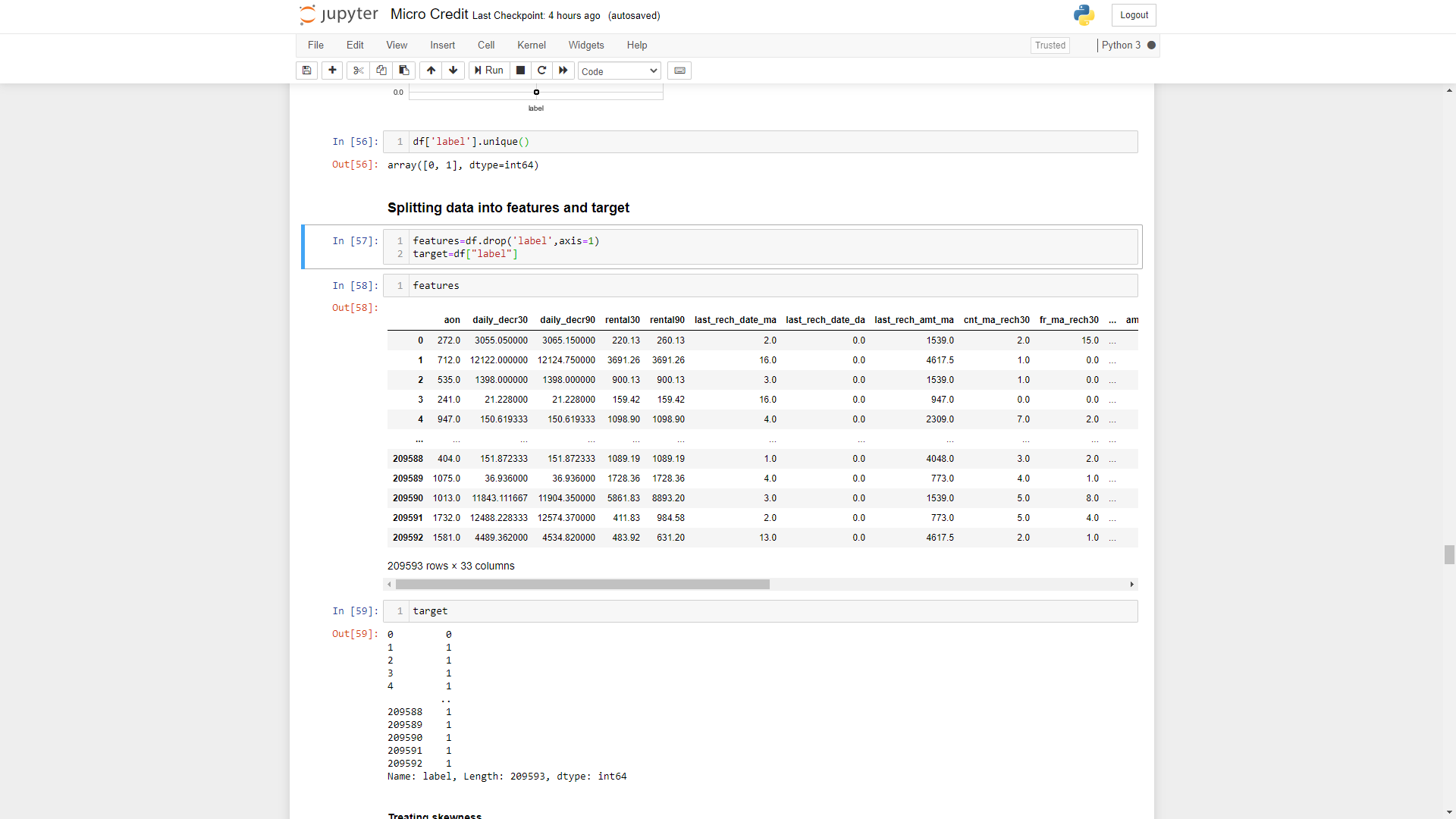
for i in df:

print(i)

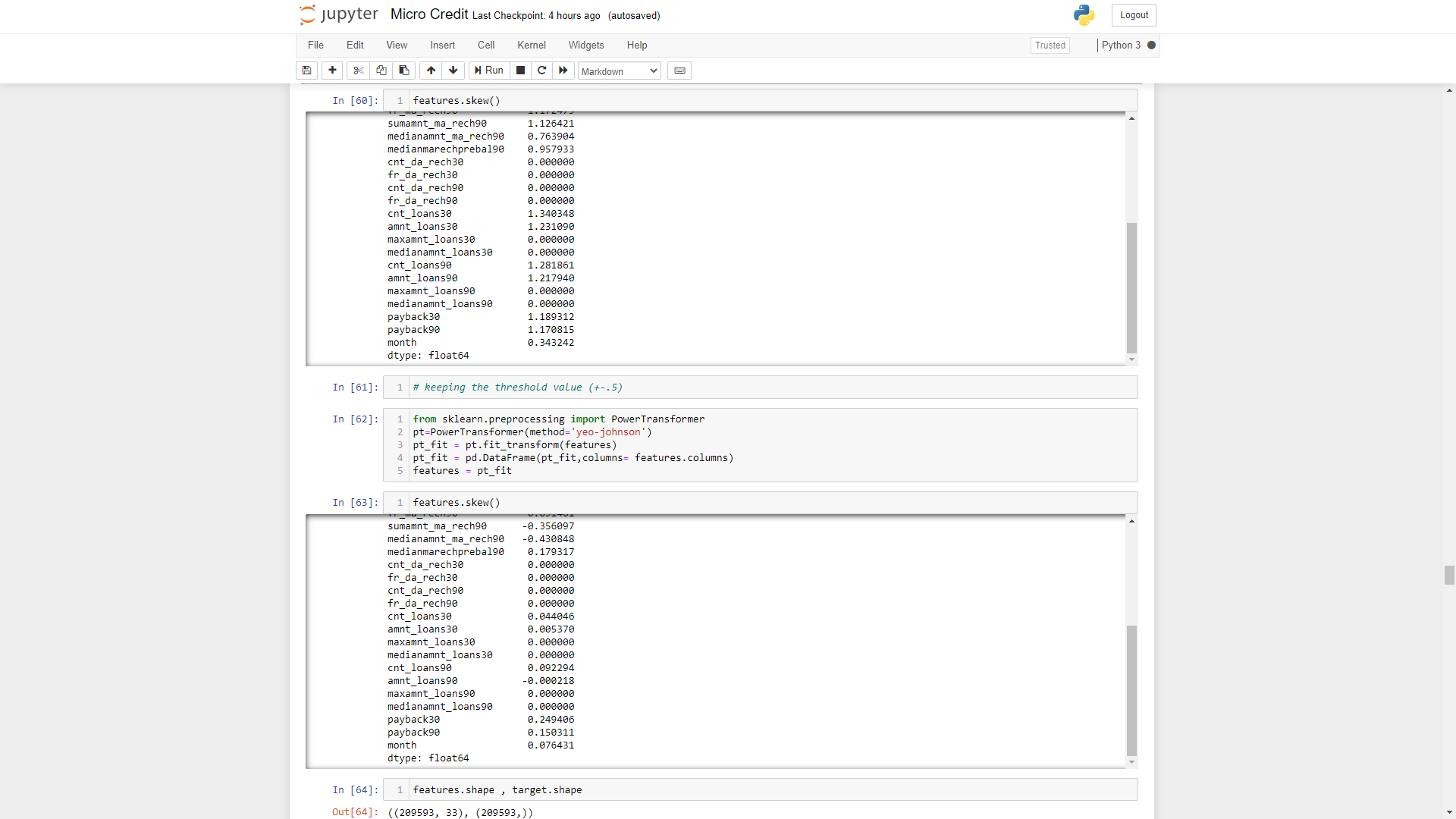
print(df[i].plot.box())

plt.show()

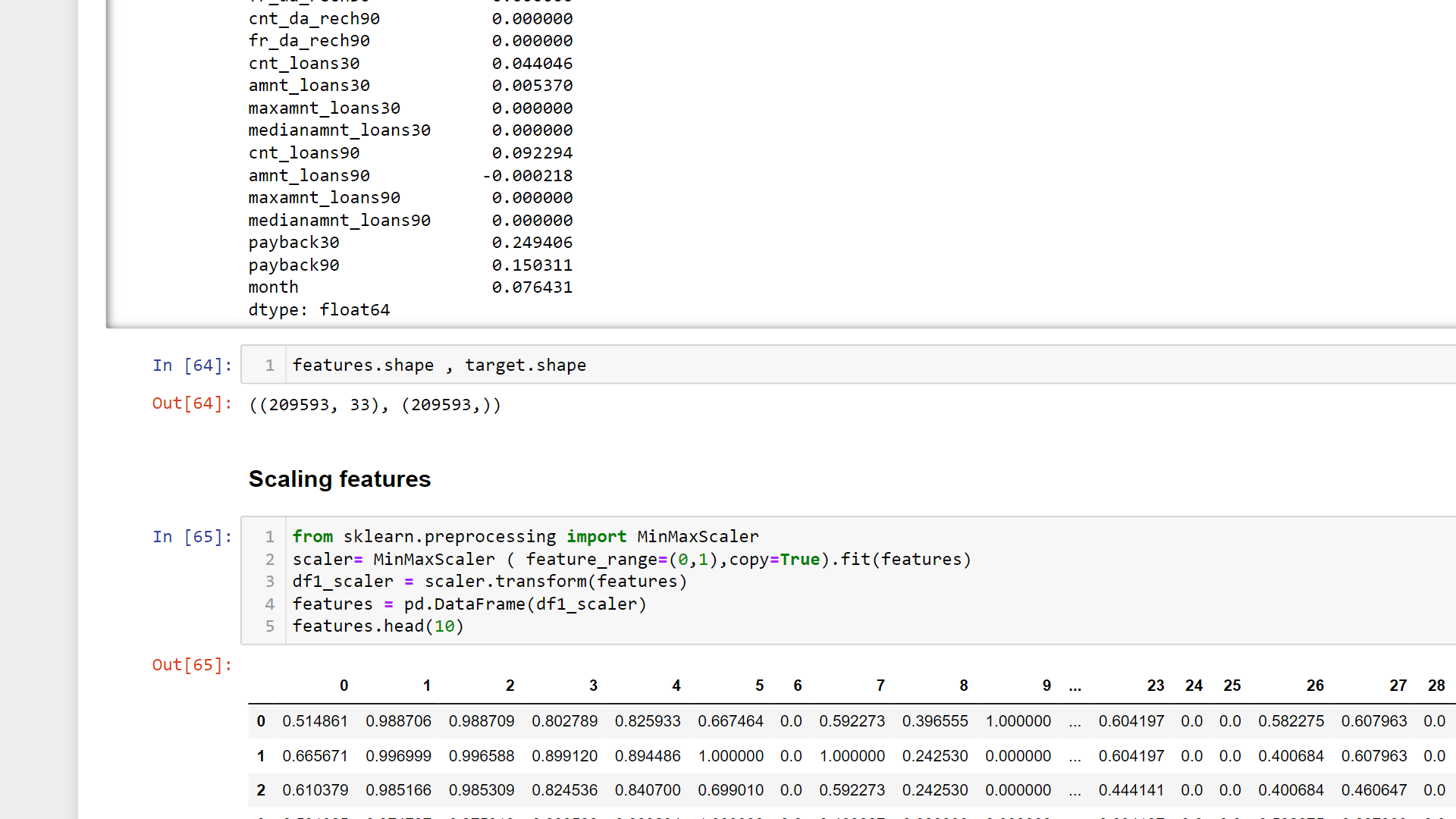
**Splitting data into features and target**



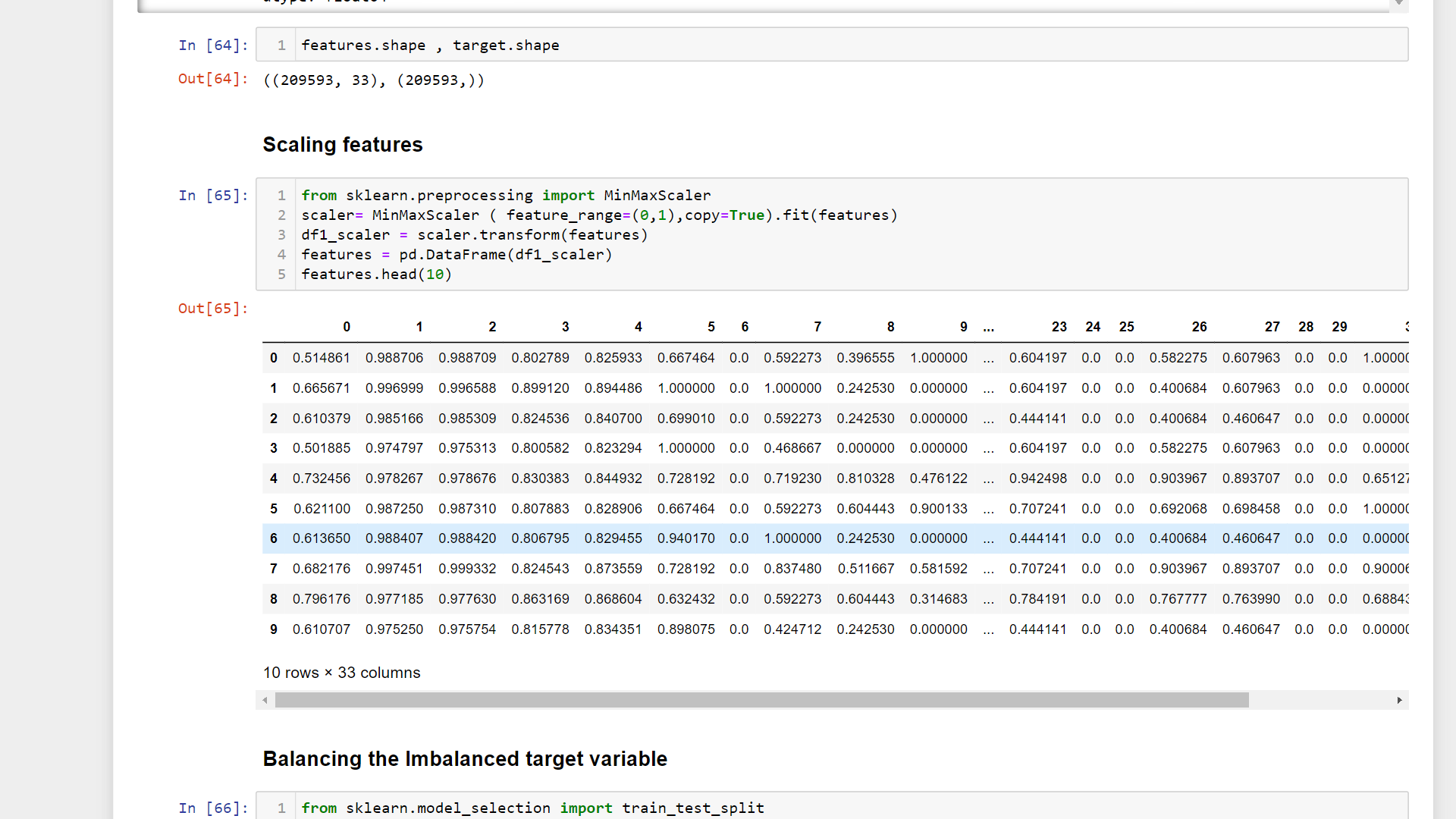
**Treating skewness using PowerTransform**

****

**Checking Shape of Features and Target**

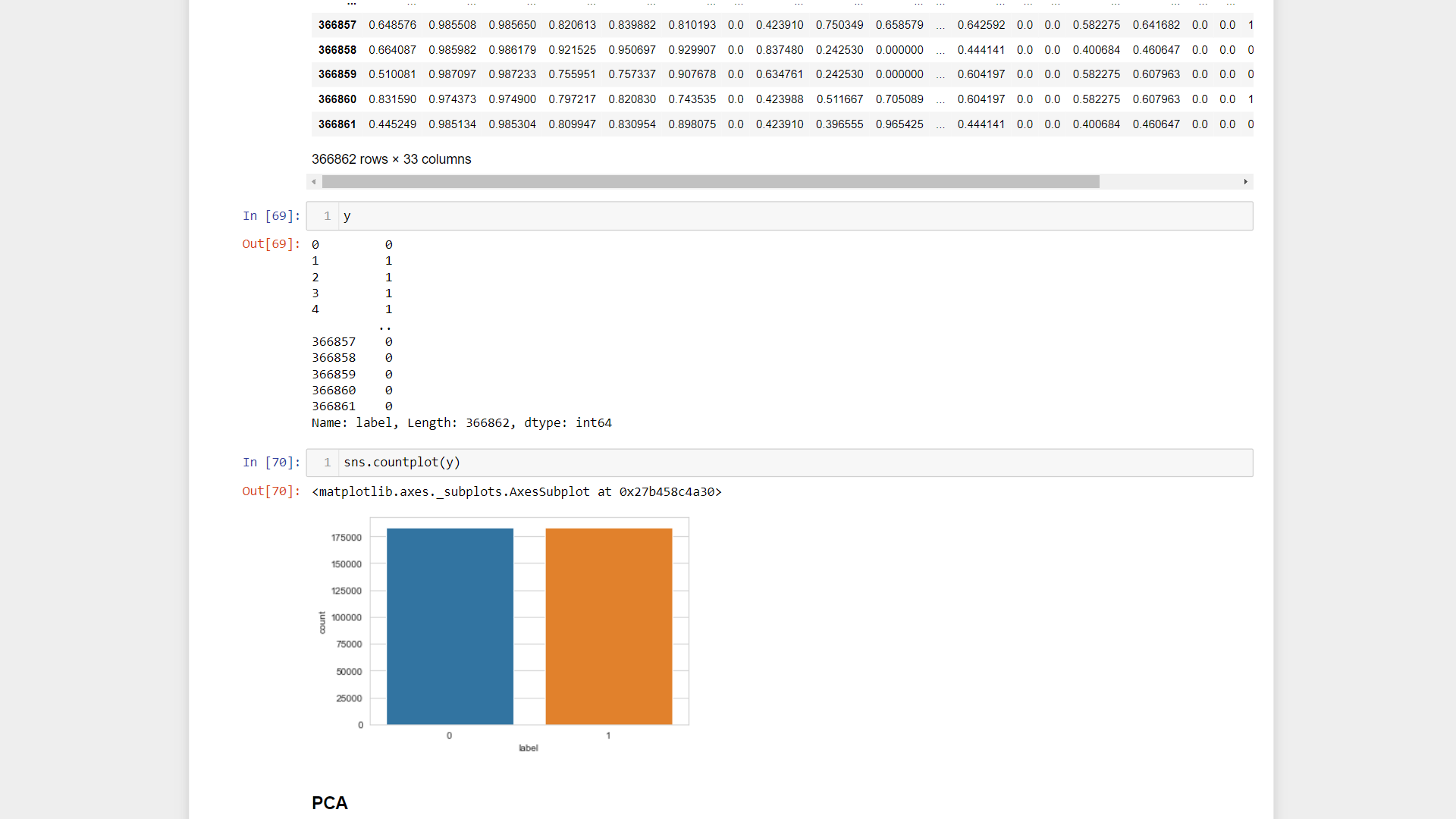


**Scaling**

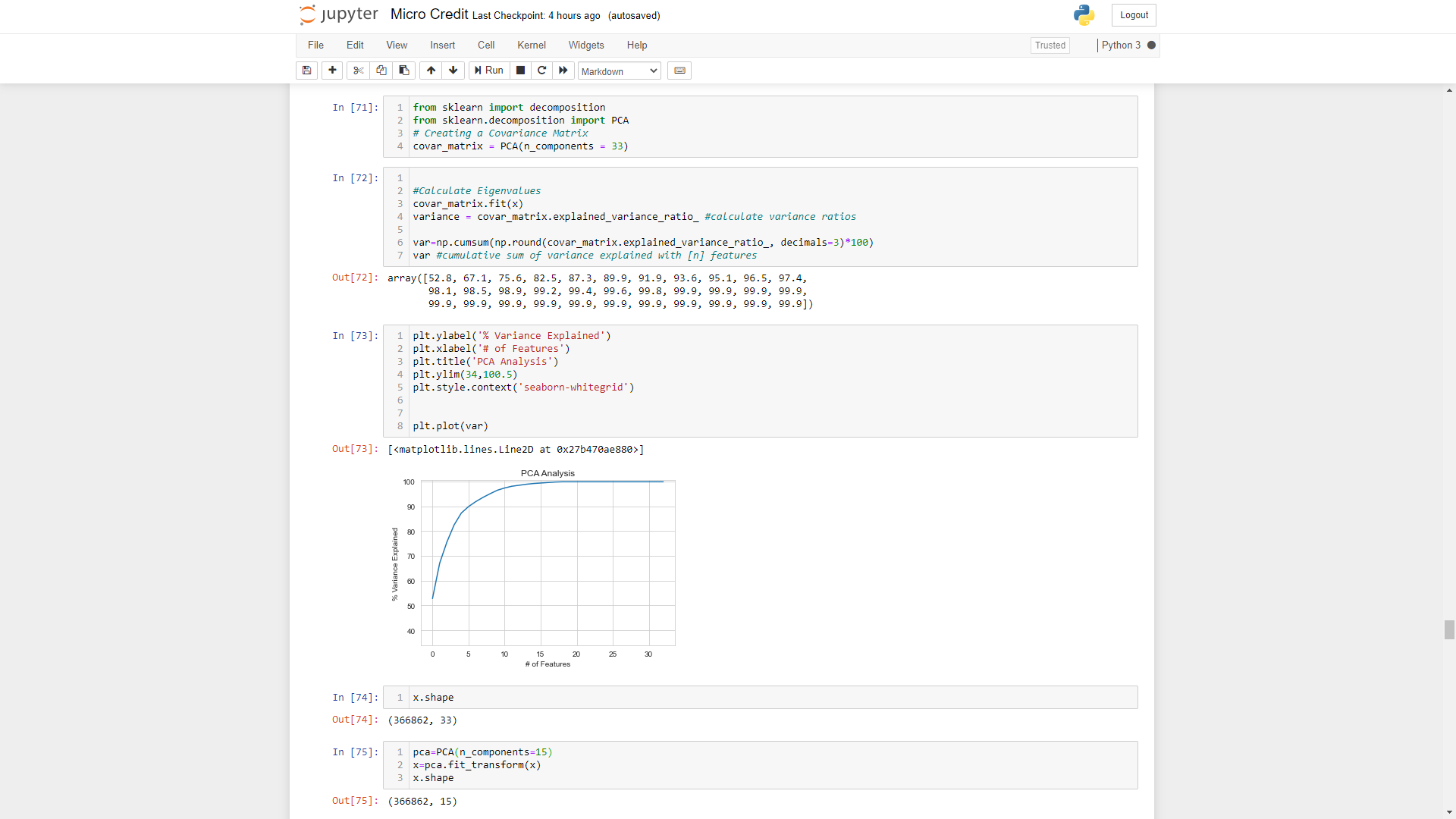
****

### Balancing the Imbalanced target variable

### 



**PCA :** (PRINCIPAL COMPONENT ANALYSIS) As we can notice in our final dataset, there are 35 columns present in order to transform the data without affecting any fields data, lowering the number of columns. After transforming, x was assigned 34 columns and now has 15 columns.

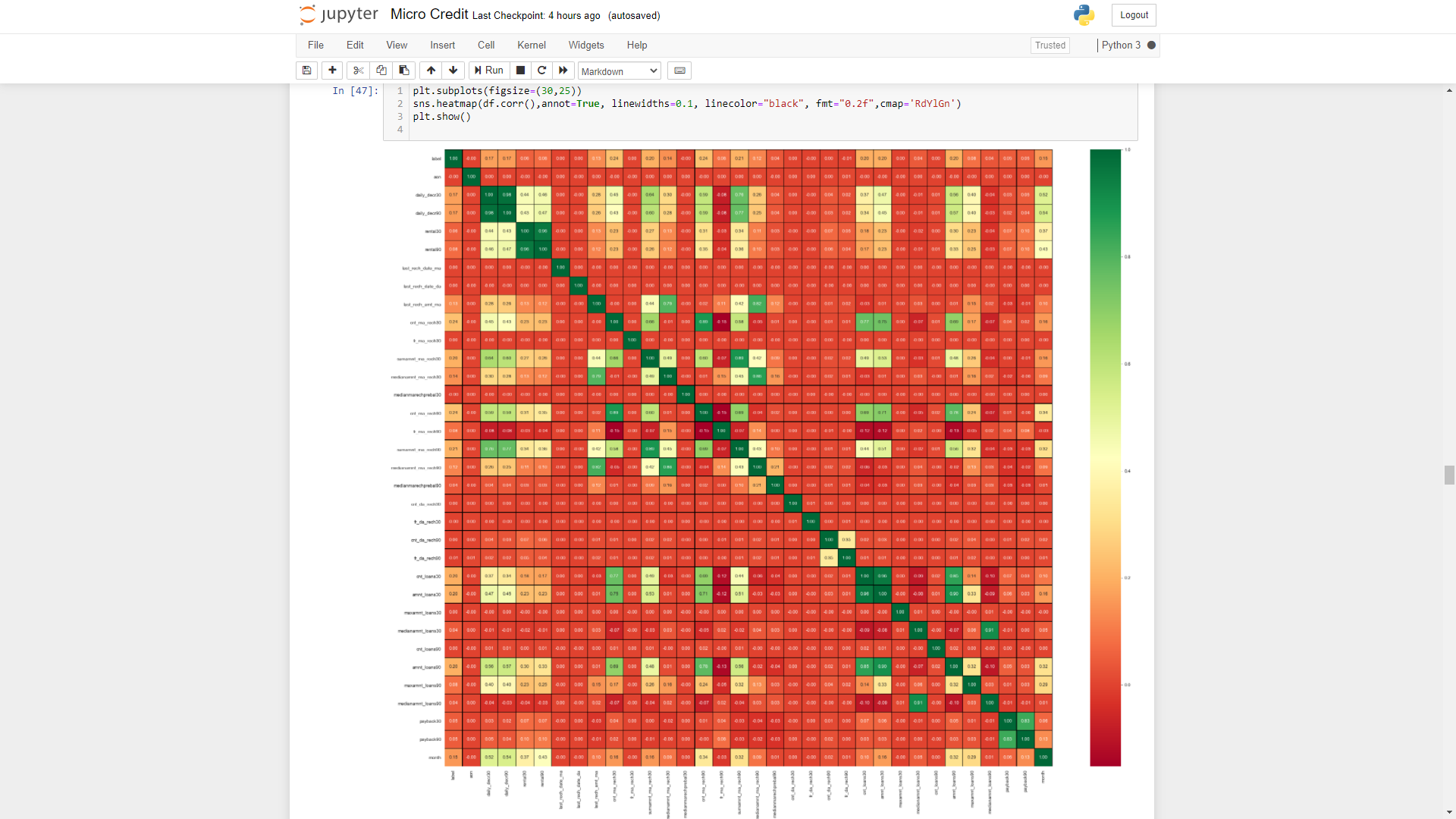


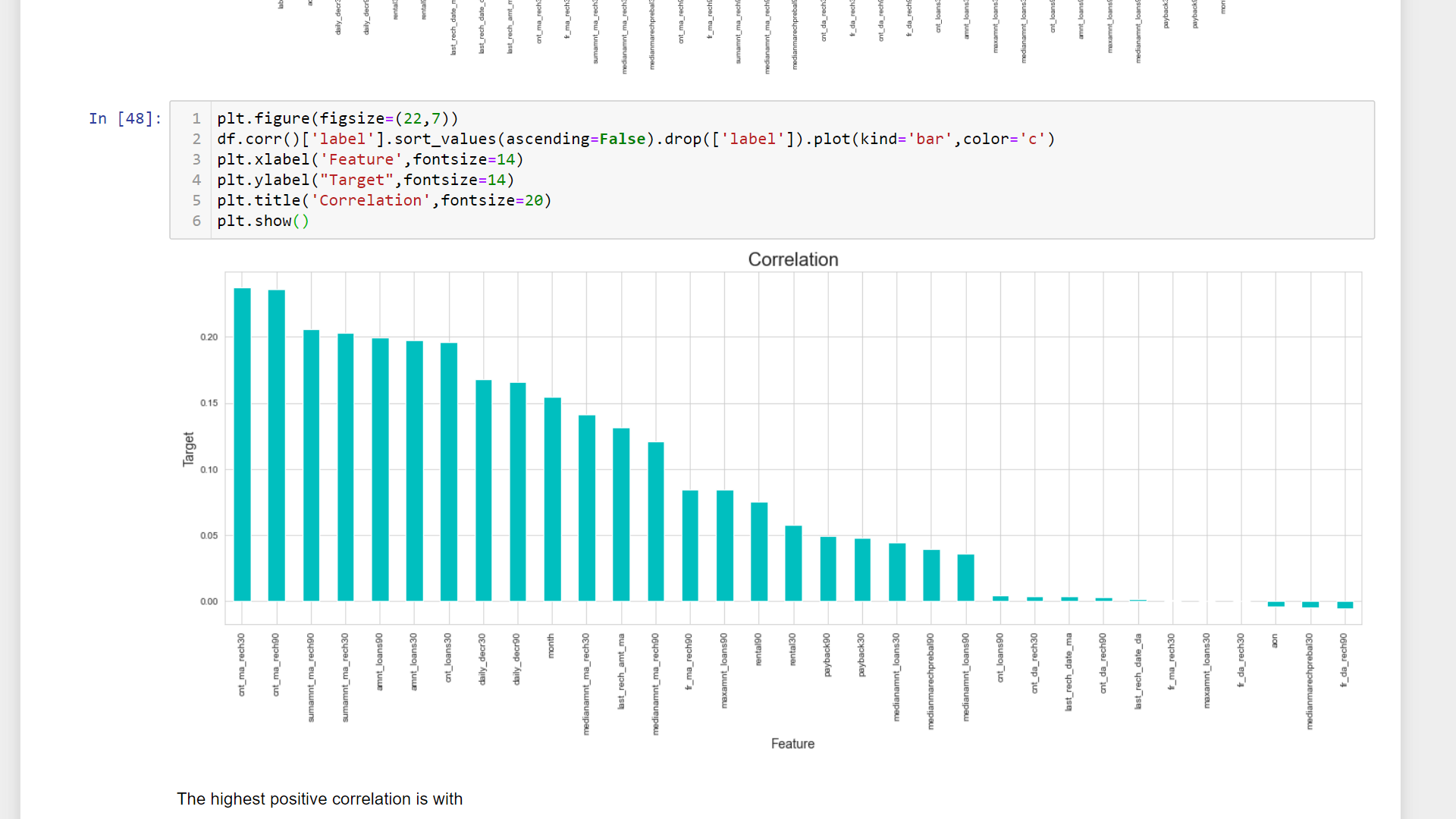
* Data Inputs- Logic- Output Relationships

Data input are the x variable which are independent variable through which we will be

able to predict the output data that is the targeted variable or Label data or y variable :

To find the relationship between the output and input data:





The highest positive correlation is with

* cnt\_ma\_reach30
* cnt\_ma\_reach90
* sumamnt\_ma\_reach30
* sumamnt\_ma\_reach90
* amnt\_loans90
* amnt\_loans30
* daily\_decr30
* daily\_decr90

While negative correlation is with

* fr\_da\_reach90
* medianmarechprebal90
* aon
* Hardware and Software Requirements and Tools Used
* **import numpy as np:** numpy is used in the dataset for working with the arrays, working in domain of linear algebra, fourier transform, and matrices
* **import pandas as pd :** Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
* **import matplotlib.pyplot as plt:** It is used to check the missing values in our dataset also used for the histogram which is to detect the count of various features lying in different groups .
* **from sklearn.linear\_model import LogisticRegression:** It helps in predicting the model with logistic regression where , Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit
* regression) is estimating the parameters of a logistic model (a form of binary regression ).
* **from sklearn.metrics import classification\_report:** A Classification report is used to measure the quality of predictions from a classification algorithm. ... The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives.
* **from sklearn.tree import DecisionTreeClassifier:** The decision tree classifier (Pang-Ning et al., 2006) creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute.
* **from sklearn.metrics import confusion\_matrix:** A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.
* **from sklearn.metrics import accuracy\_score :** In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y\_true.
* **from sklearn.metrics import roc\_curve:** ROC is a plot of signal (True Positive Rate) against noise (False Positive Rate). ... The model performance is determined by looking at the area under the ROC curve (or AUC). The best possible AUC is 1 while the worst is 0.5 (the 45 degrees random line).
* **import matplotlib.pyplot as plt**
* **from sklearn.metrics import roc\_auc\_score:** ROC stands for curves receiver or operating characteristic curve. It illustrates in a binary classifier system the discrimination threshold created by plotting the true positive rate vs false positive rate. ... The roc\_auc\_score always runs from 0 to 1, and is sorting predictive possibilities.
* **import seaborn as sns :** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
* **from sklearn.model\_selection import GridSearchCV:** GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.
* **from sklearn.preprocessing import StandardScaler:** Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance). It is a standardized feature which is extracted by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. StandardScaler makes the mean of the distribution 0. About 68% of the values will lie be between -1 and 1
* **from sklearn.model\_selection import train\_test\_split:** train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. ... By default, Sklearn train\_test\_split will make random partitions for the two subsets. However, you can also specify a random state for the operation.
* **from sklearn.model\_selection import cross\_val\_score:** It takes the features df and target y , splits into k-folds (which is the cv parameter), fits on the (k-1) folds and evaluates on the last fold. It does this k times, which is why you get k values in your output array.
* **from sklearn.naive\_bayes import GaussianNB:** A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous 18 values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.
* **from sklearn.neighbors import KNeighborsClassifier:** The K-nearest neighbors ( KNN ) algorithm is a type of supervised machine learning algorithms. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks. ... KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data.
* **from sklearn.model\_selection import GridSearchCV : GridSearchCV** is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters
* **from sklearn.ensemble import RandomForestClassifier :** A random forest is a meta
* estimator that fits a number of decision tree classifiers on various sub-samples of the
* dataset and uses averaging to improve the predictive accuracy and control over-fitting.
* The sub-sample size is controlled with the max\_samples parameter if
* bootstrap=True (default), otherwise the whole dataset is used to build each tree.
* **from sklearn.metrics import r2\_score:** A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0. Parameters y\_truearray-like of shape (n\_samples,) or (n\_samples, n\_outputs) Ground truth (correct) target values.
* **import warnings**
* **warnings.filterwarnings('ignore') :** The warn() function defined in the ' warning ' module is used to show warning messages. The warning module is actually a subclass of Exception which is a built-in class in Python. filter\_none. # program to display a warning message. import warnings.
* **from sklearn.externals import joblib:** To save the file in object format.

**Model/s Development and Evaluation**

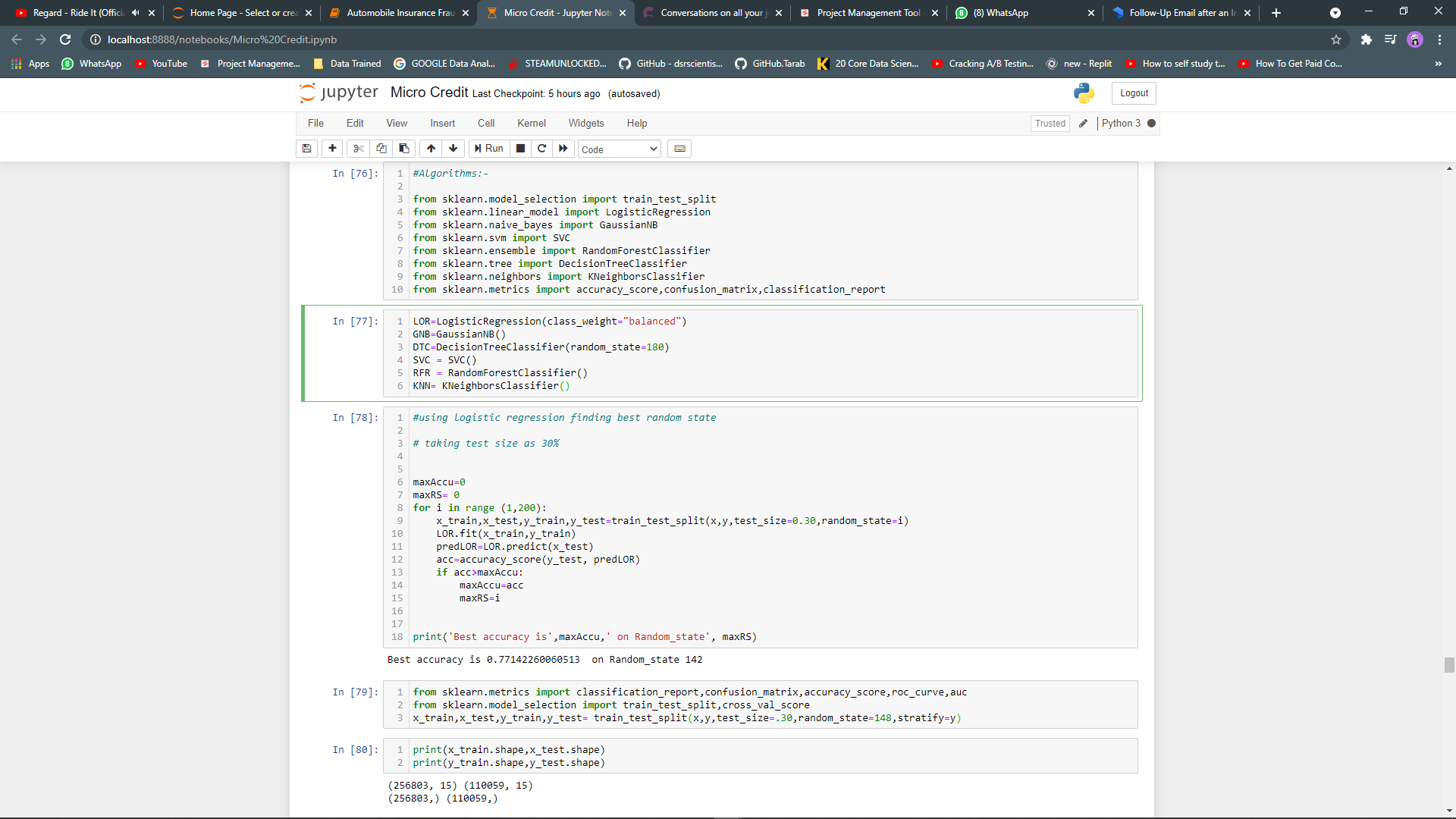
* Identification of possible problem-solving approaches (methods)

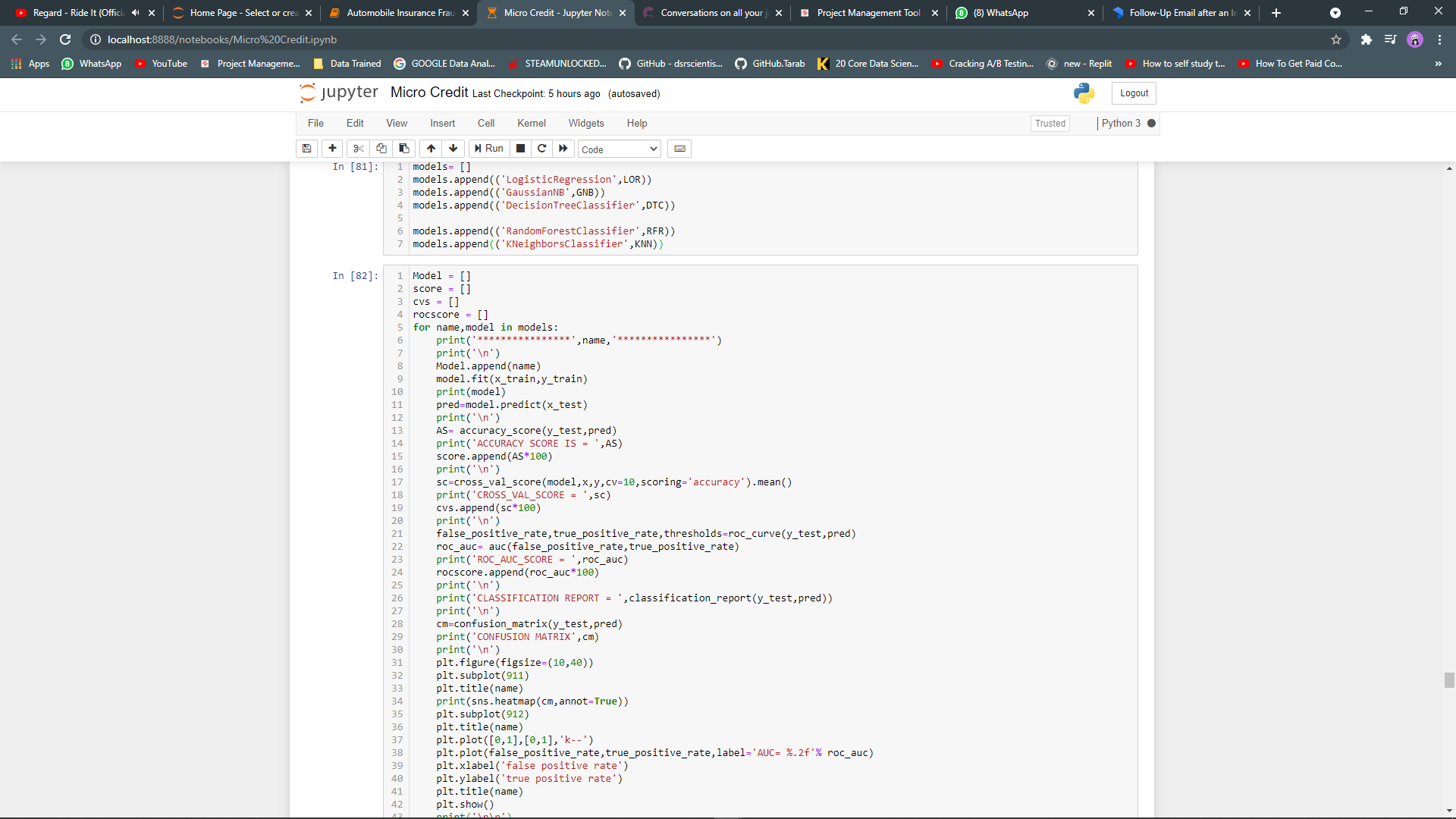
Uneven target variable – I used over sampling to correct it.

* Testing of Identified Approaches (Algorithms)

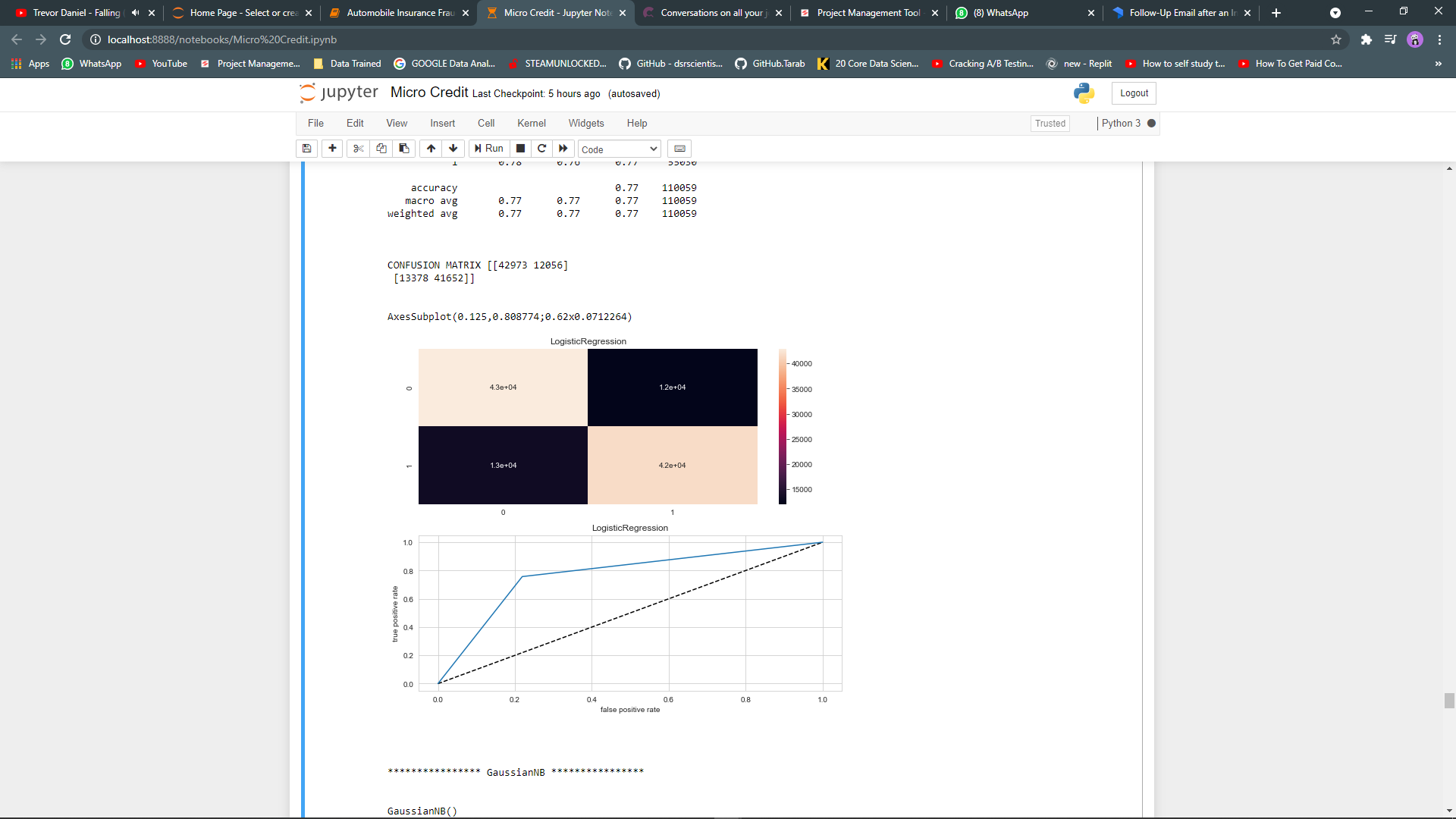
1. LOR=LogisticRegression(class\_weight="balanced")
2. GNB=GaussianNB()
3. DTC=DecisionTreeClassifier(random\_state=180)
4. RFR = RandomForestClassifier()
5. KNN= KNeighborsClassifier()

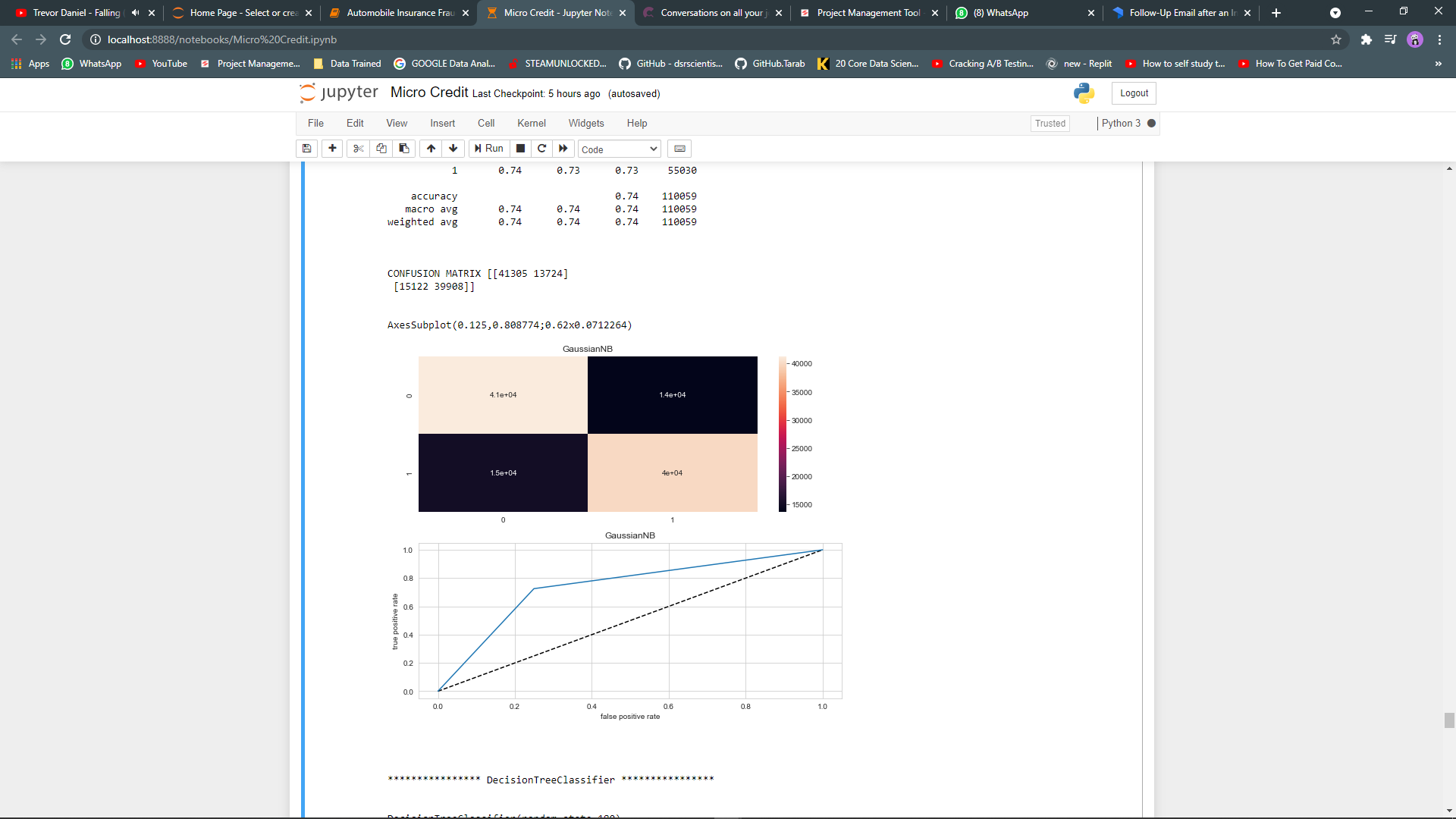
* Run and Evaluate selected models

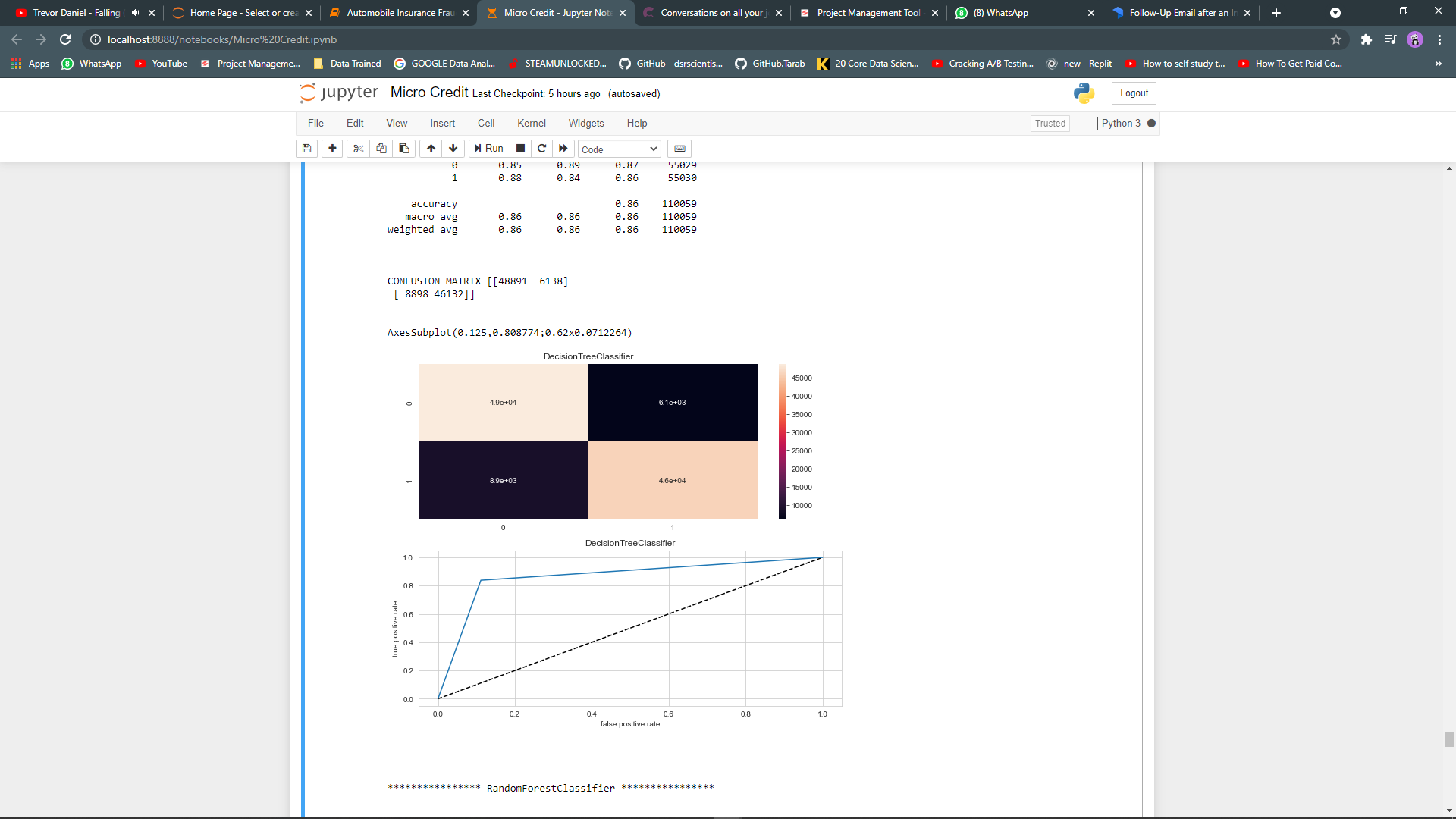


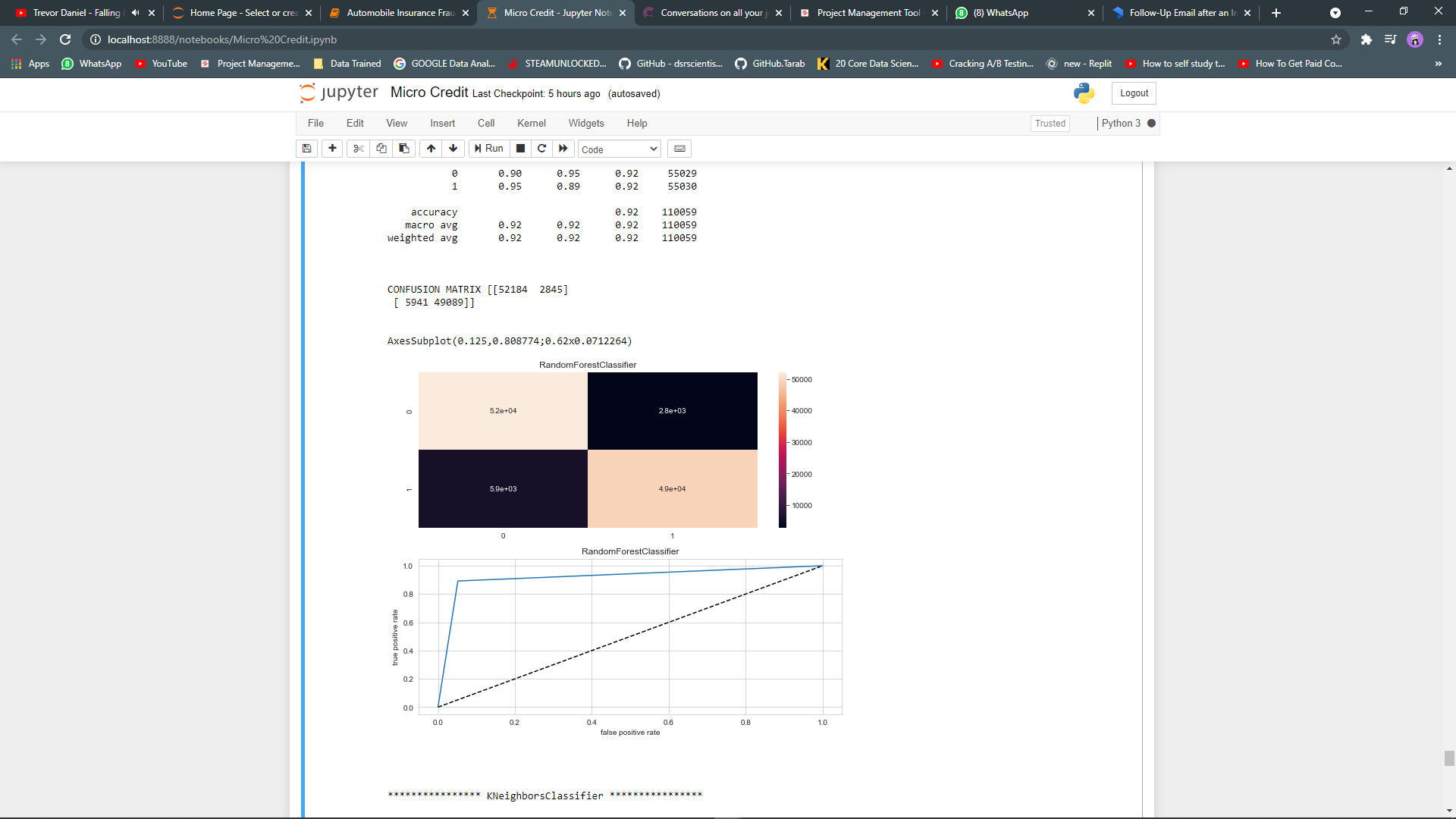


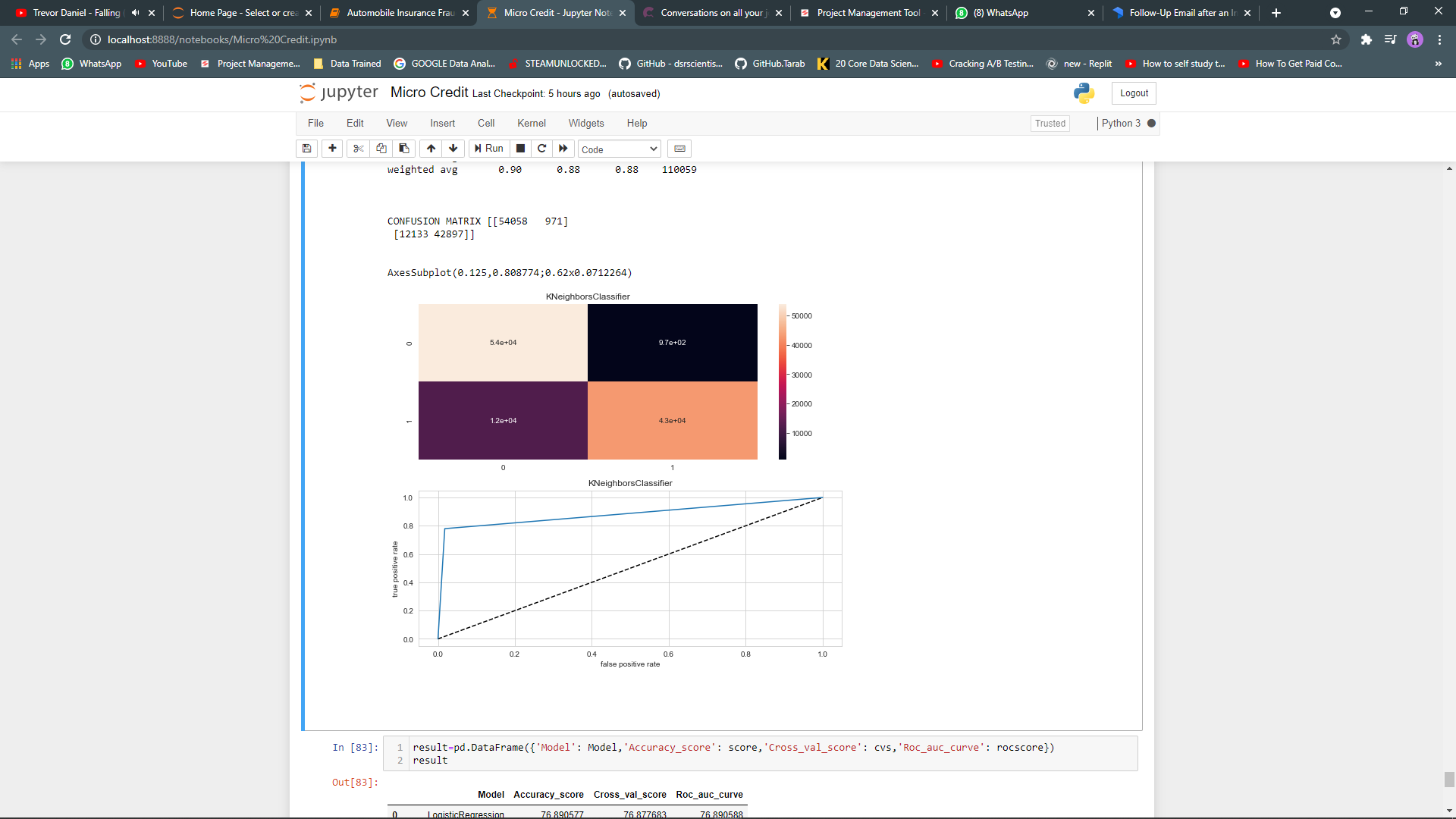
* Key Metrics for success in solving problem under consideration
* ACCURACY SCORE
* CROSS\_VAL\_SCORE
* ROC\_AUC\_SCORE
* precision
* recall
* f1-score
* Visualizations



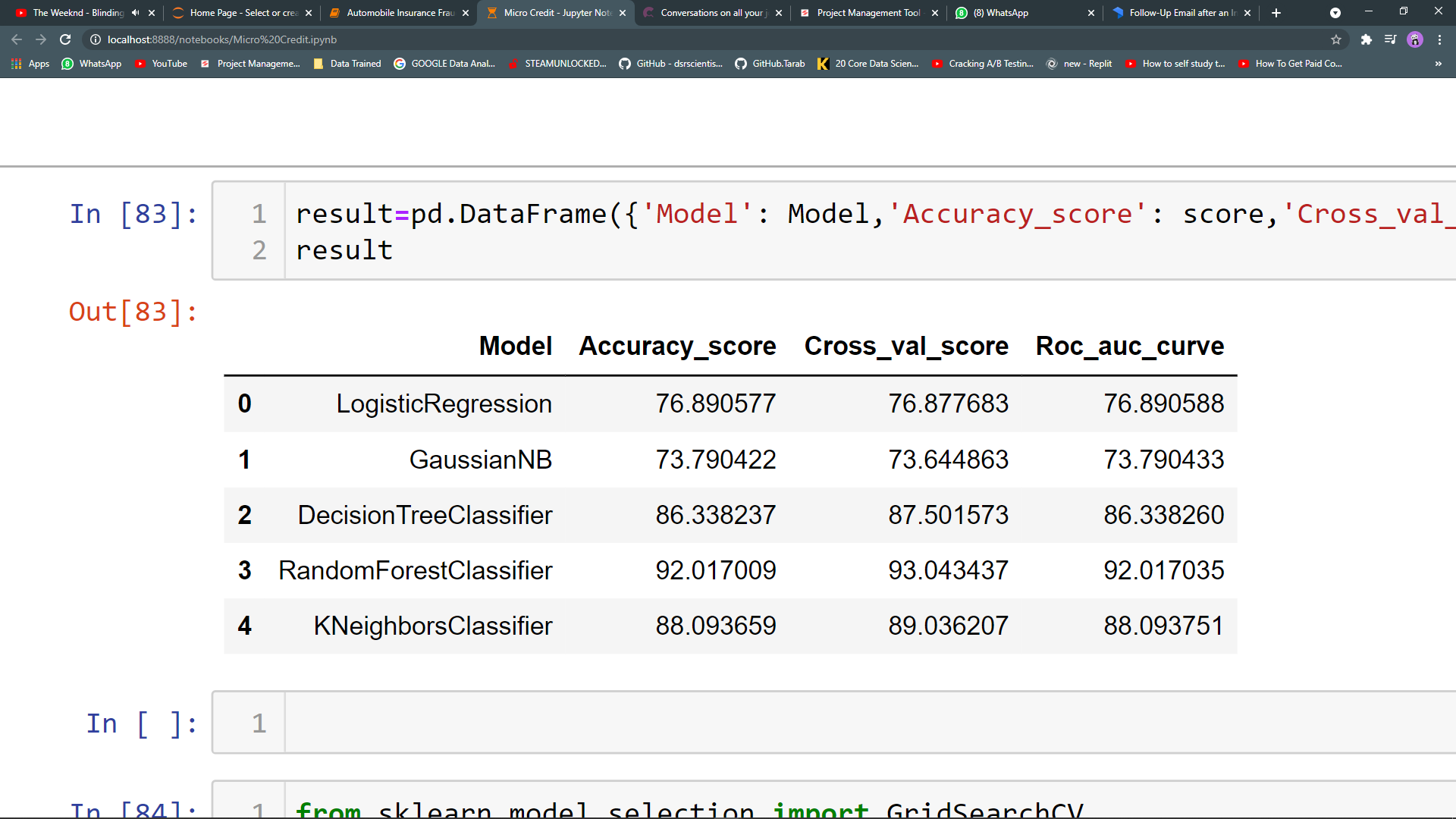








* Interpretation of the Results



**CONCLUSION**

* Key Findings and Conclusions of the Study
* Key findings:
* a) If Age on the cellular network is more than 6000 days or 8000 days then the
* customers are predicted not to be the defaulters and the customer who have
* joined recently or below 6000 days may be there is a chance of predicting them
* as a defaulter.
* b) Maximum loan amount of 6 Rupiah in last 90 days defaulters ranging from 22000
* to 24000 when compared to 12 Rupiah.
* c) Amount loan of 6 Rupiah in the last 90 days has been taken by a maximum
* number of customers.
* d) Maximum defaulters which have not paid the loan or the total amount of loans
* taken by the users in last 90 days is maximum for 6 Rupiah i.e. The amount of 6
* Rupiah is the amount where the maximum number of users have paid and not
* paid the loan.
* e) It has also come to our notice that the maximum amount of loan in the last 90
* days is maximum for 6 Rupiah with respect to the 12 Rupiah..
* Learning Outcomes of the Study in respect of Data Science

Working with such big data was fun and made me keep more patient with the

data.

2) With this project it taught me to be more careful with the data you have been

provided as these data are very expensive and many people have worked day

and night to build this data. If there is a loss of data more than 10 % then it would

have a direct effect on the prediction and analysis of the data.

3) It made me understand that accuracy score is not the only metric to identify the

best model for our prediction because of that we have used various metrics to

perform the prediction of the problem.

4) The Biggest challenge was to face this big data as it was taking too much time torn various algorithms in python but with the help of mentors it was all sorted.

* Limitations of this work and Scope for Future Work

Too big a data set to work on, my device was not able to do the gridsearchcv,SVC

technique as it was hanging all the time whenever I used it for any models above.

2) There are few customers which have no loan history. They have been also

considered in testing, if we could remove this data then we would have got a

more efficient score. We couldn't remove it as we were not allowed to lose the

data more than 8 or 10 %