Loan Prediction

Contents

[Project Overview 2](#_Toc59568378)

[Problem Statement 2](#_Toc59568379)

[Metrics 2](#_Toc59568380)

[Data pre-processing 2](#_Toc59568381)

[Exploratory Data Analysis 4](#_Toc59568382)

[Feature-1: Loan ID: 4](#_Toc59568383)

[Feature-2: Married (Y/N) 4](#_Toc59568384)

[Feature-3: Education (Graduate/Not-Graduate) 5](#_Toc59568385)

[Feature-4: Property-Area (Urban/ Semi-Urban/ Rural) 5](#_Toc59568386)

[5](#_Toc59568387)

[Feature-5: Applicant Income 6](#_Toc59568388)

[Feature-6: Co-applicant Income 7](#_Toc59568389)

[Feature-7: Gender (Male/Female) 7](#_Toc59568390)

[Feature-8: Dependents (0,1,2,3+) 8](#_Toc59568391)

[Feature-9: Loan Amount Term 9](#_Toc59568392)

[Feature-10: Loan Amount 9](#_Toc59568393)

[Feature-11: Self Employed 10](#_Toc59568394)

[Feature-12: Credit History 11](#_Toc59568395)

[Data Wrangling and Feature Engineering 11](#_Toc59568396)

[Algorithms 13](#_Toc59568397)

[Hyperparameter tuning 15](#_Toc59568398)

[Result 16](#_Toc59568399)

# Project Overview

Loans allow for growth in the overall money supply in an economy and open up competition by lending to new businesses.[[1]](#footnote-1) Thus, with the growing pace of emerging businesses, speeding up of the approval process for potential customers becomes very important.

In this project, I attempted to build a Machine Learning model based on 7 features that is capable of predicting whether an individual should be eligible for loan approval.

# Problem Statement

The goal is to create a classifier that can predict whether an individual should be approved for loan or not; the tasks involved are the following:

1. Download and pre-process the data.
2. Gain insights about the data, engineer features (If necessary) and make it ready for the machine learning algorithms.
3. Train the data and test for evaluation metrics.
4. Fine tune the hyperparameters.
5. Make predictions on the testing set.

# Metrics

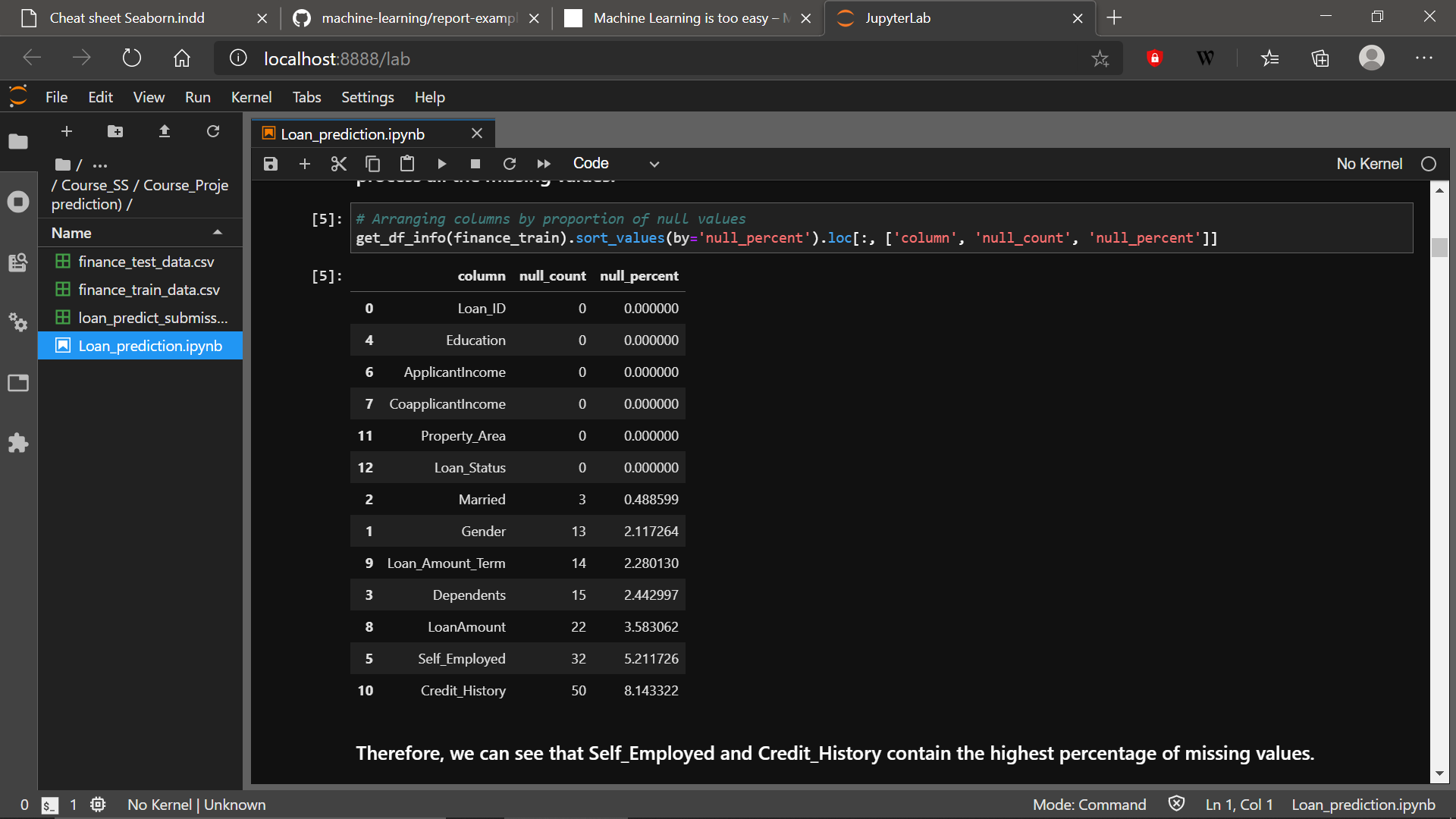
The most common metric for a binary classifier is Accuracy but it is not as robust as some other metrics like Balanced Accuracy, AUC-ROC. Therefore, we can use the following metrics to judge the performance of this classifier:

1. Balanced Accuracy (BAC) = 0.5\*(Sensitivity + Specificity)
2. Recall (Sensitivity) = Tells us what % of True positives were recalled.
3. AUC = Area under curve of a ROC curve.

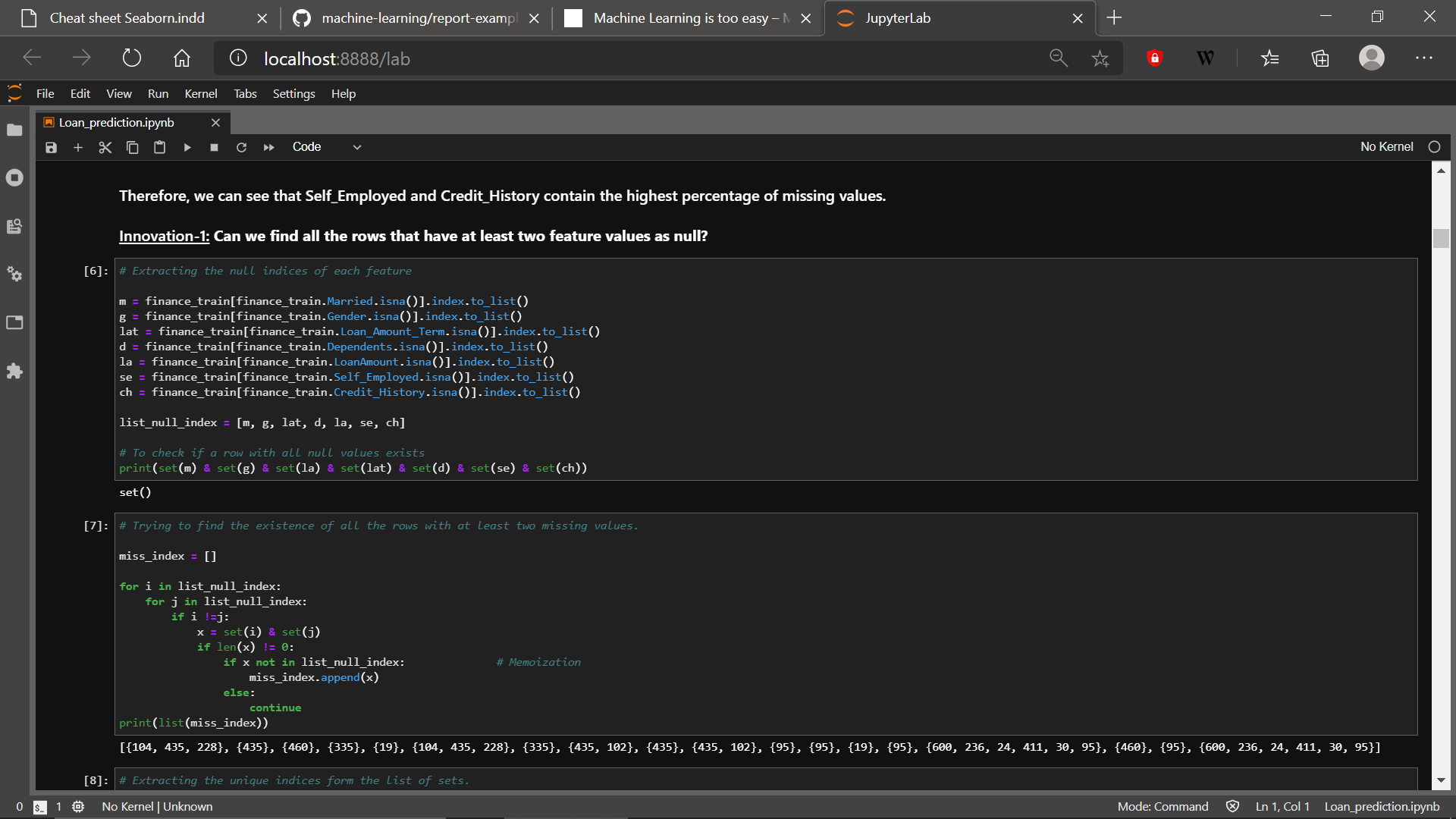
# Data pre-processing

The two Data pre-processing steps that we will apply to our data set are:

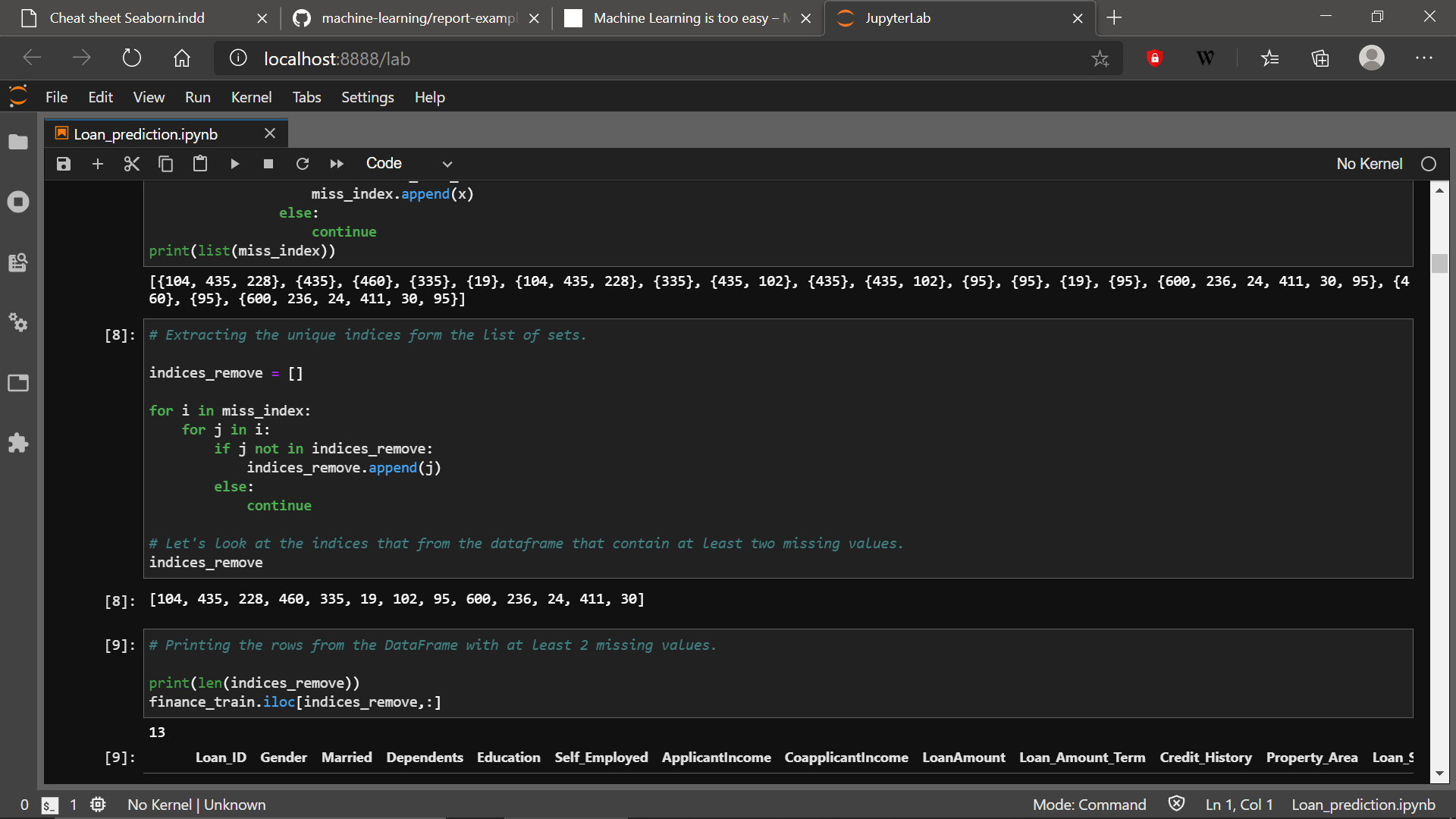
1. Handling Missing values – Filling the holes in our training and testing dataset.
2. Discretization – Converting continuous values into discrete categories.

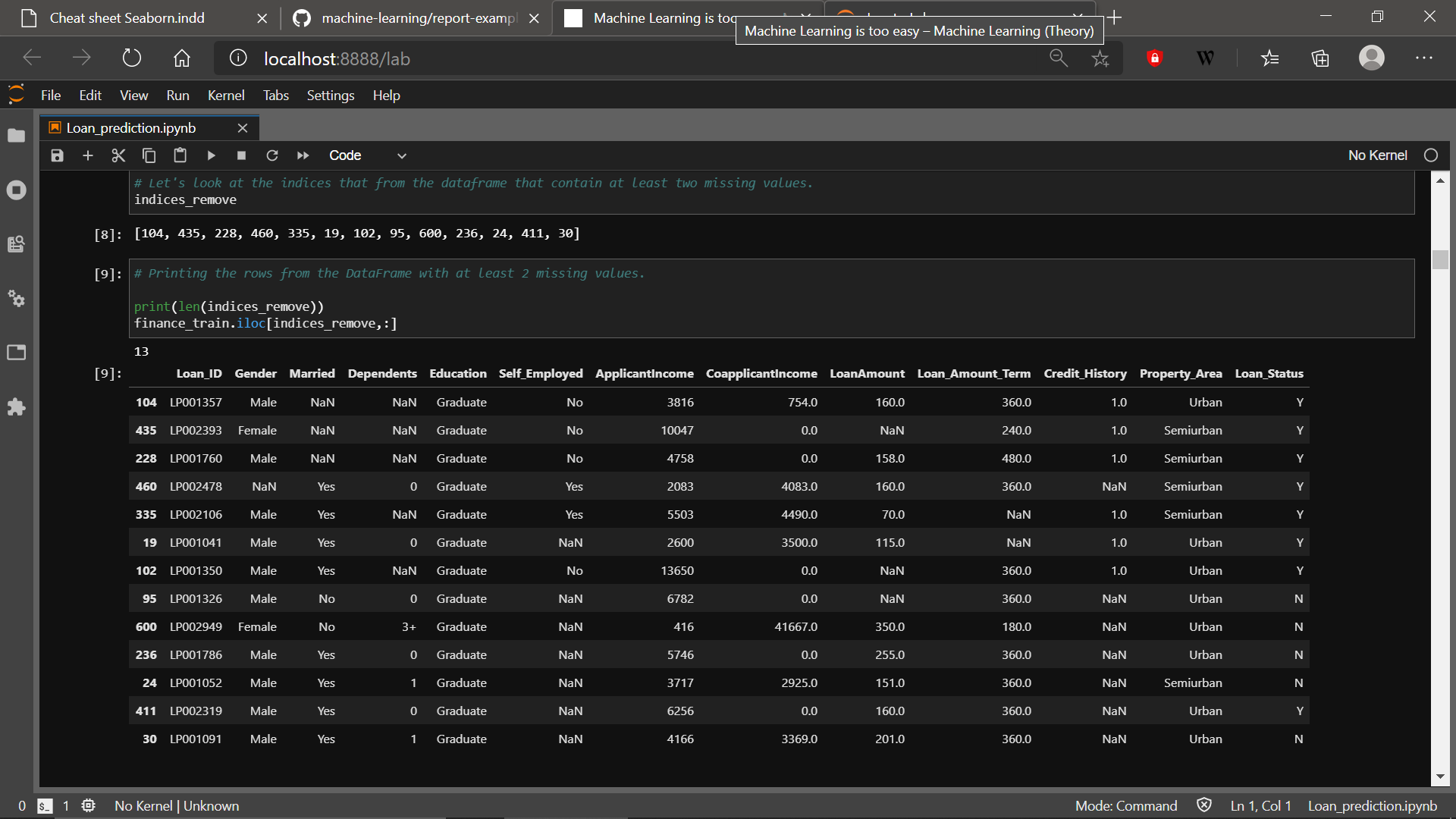
The missing values per feature is shown in the table as follows:

**Innovation: There are a lot of ways to handle missing values but to decrease the probability of any random error leakage into the training data, can we find and drop all the rows which have at least two feature values missing? Let’s see.**



The code above helps to find a list of set of indices containing at least two missing feature values.

Now, we need to extract all the unique values from the list of sets.

The above are the unique indices with at least two missing values. We can have a look at the rows for verification and then drop these 13 rows.

# Exploratory Data Analysis

Since the number of features were small, we can begin the process of exploring the data feature by keeping these three questions in mind:

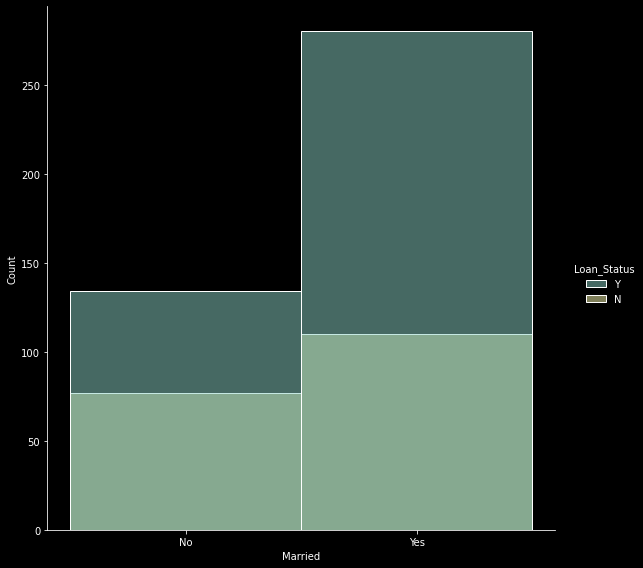
1. What does the distribution of each feature look like?
2. If the variables are categorical, how much does each category impact the Loan status?
3. If there are missing values, what should be the best strategy to impute them?

## Feature-1: Loan ID:

This feature had all unique values and was thus should not be considered a part of our model building.

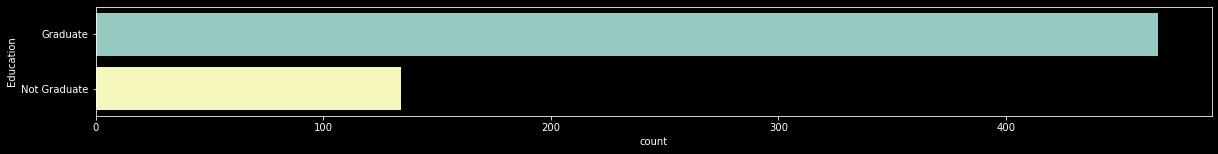
## Feature-2: Married (Y/N)

1. There were a greater number of Married people in the training data.
2. It’s a categorical feature and on checking whether being married impacts the Loan status or not, we can observe this:

In hard numbers, 72% of the people who were **married** got approved whereas 65% people who were **not** **married** got approved indicating that being married has a significant impact on the loan status.

1. There were no missing values in this feature.

## Feature-3: Education (Graduate/Not-Graduate)

1. Majority of the applicants were graduates.
2. This is a categorical feature as well and it also impacts the Loan approval status quite a bit.
3. There were no missing values in this feature.

## Feature-4: Property-Area (Urban/ Semi-Urban/ Rural)

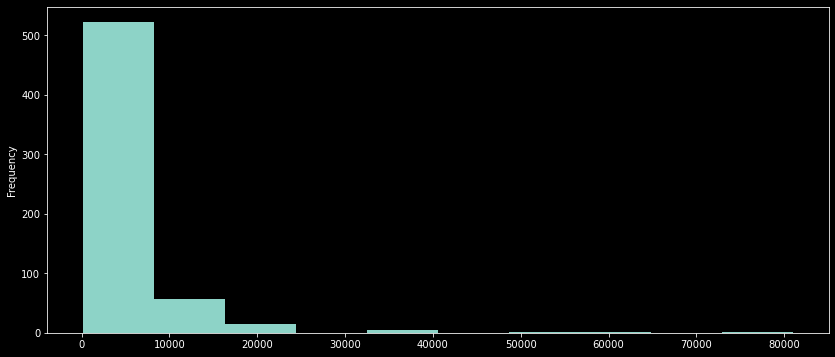
1. Its distribution is somewhat linked to that of Education as people from Rural areas tend to have a lower chance of being Graduated. Rural and Urban populations did not have a huge difference whereas the Semi-urban population was the highest.
2. This is again a categorical feature and its relation to the loan being approved is quite interesting. We would expect people from the Urban population to have the highest chance of Loan approval but the training data showed this:

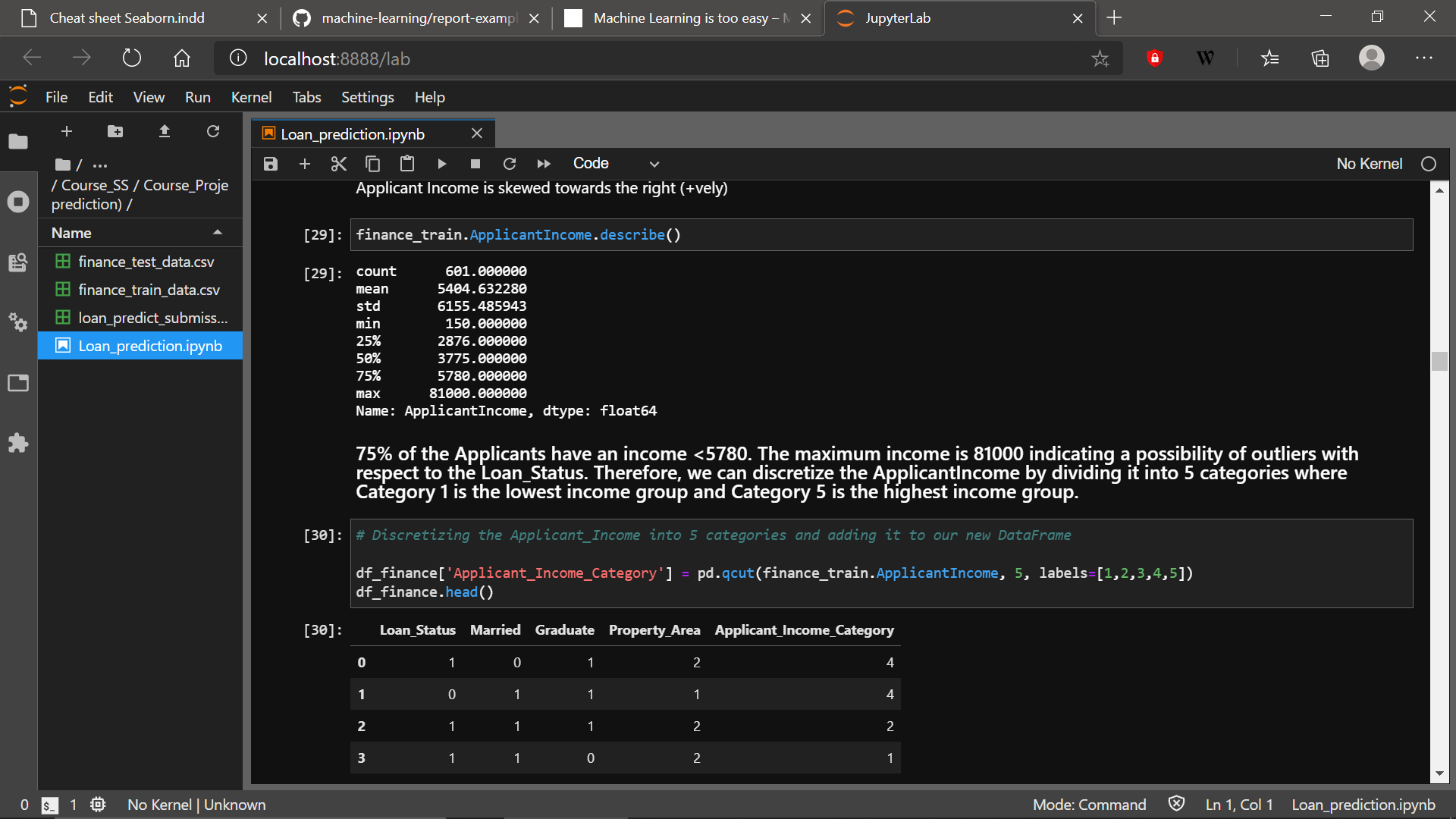
## 

Therefore, we observe that the semi-urban population were the ones with the highest chances of being approved. Quite an interesting insight.

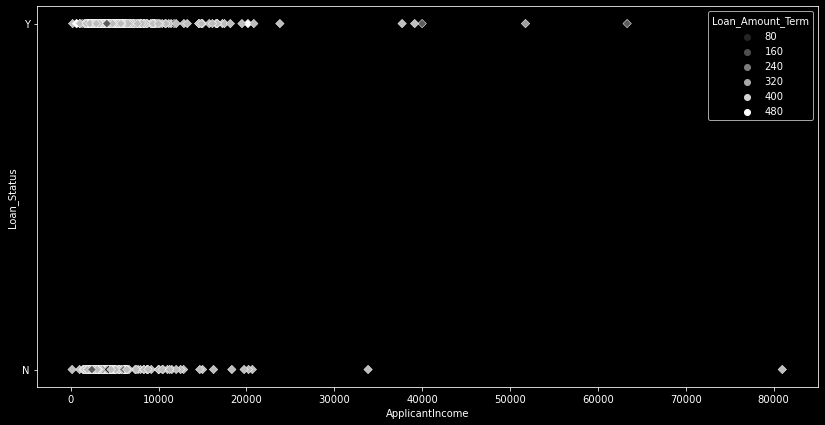
1. There were no missing values in this feature.

## Feature-5: Applicant Income

1. The distribution of income was somewhat right skewed.

On seeing hard numbers, Income appears to have some outliers. 75% of the income is below 5780.

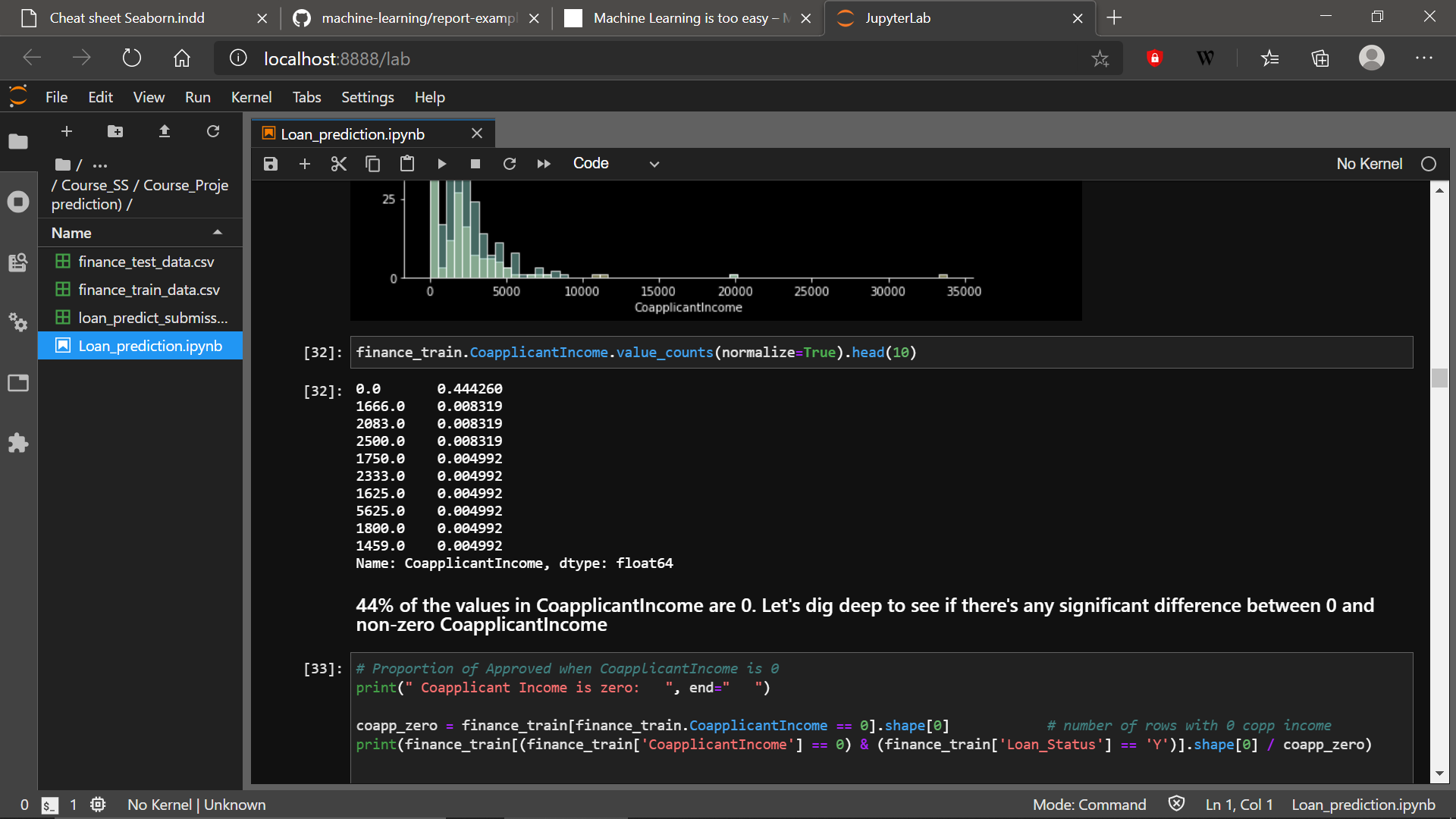
1. It’s impact on the Loan Status is shown below



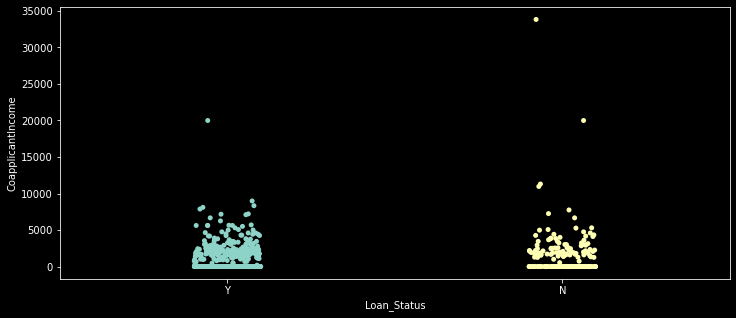
1. There were no missing values in the Applicant Income.

## Feature-6: Co-applicant Income

1. There were a lot of unique values in this feature.
2. It’s a continuous in nature. Let’s visualize its distribution in terms of proportion.

We observe that about 44% of the co-applicants have a 0 income.

So, the next sensible thing to do would be to check how various co-applicant incomes impact the Loan approval status.

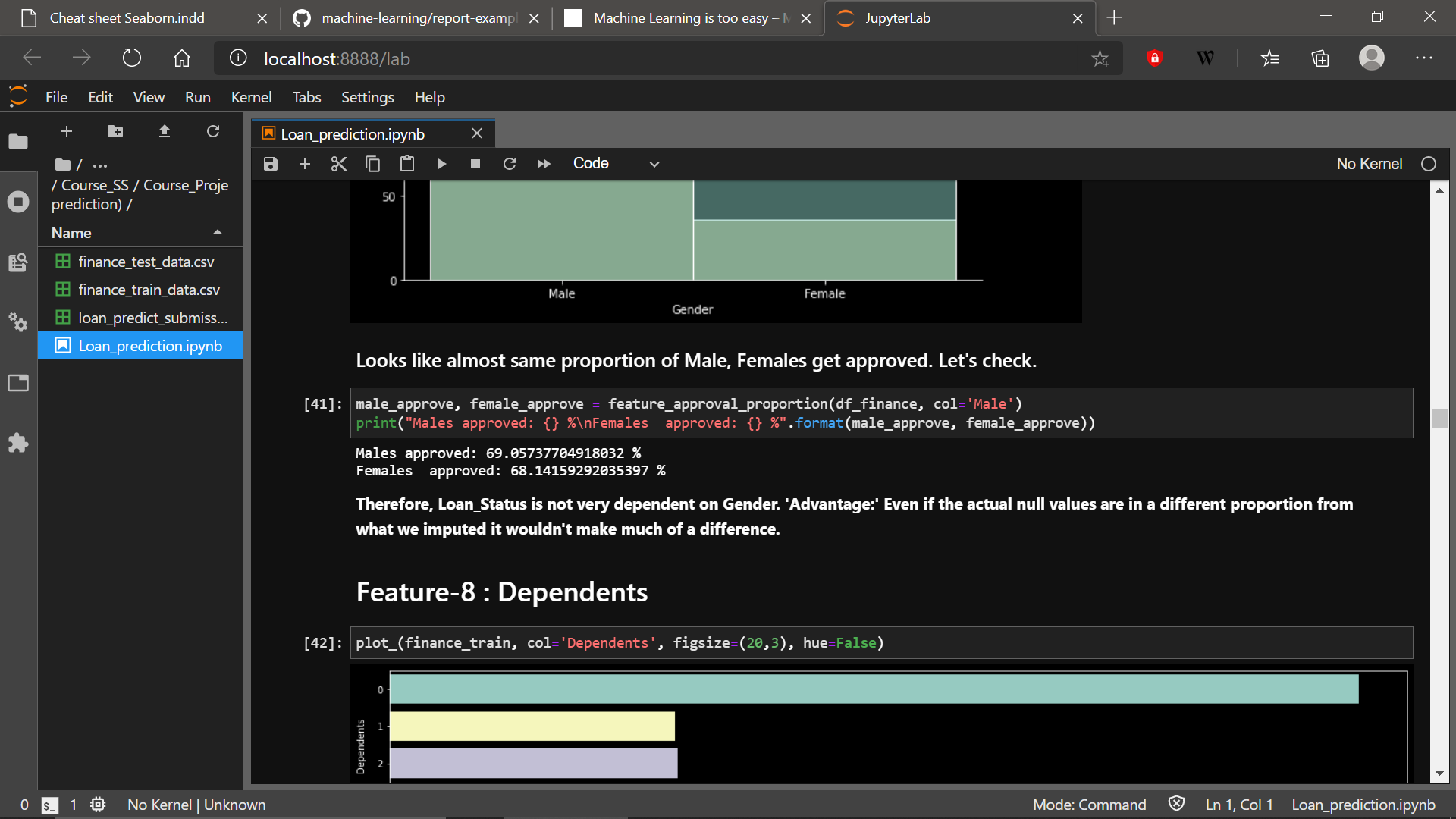


Therefore, there was no visible change in the approval status with the increase in co-applicant income. We can use this insight to engineer a feature which is covered in the next section.

1. There were no missing values in this feature.

## Feature-7: Gender (Male/Female)

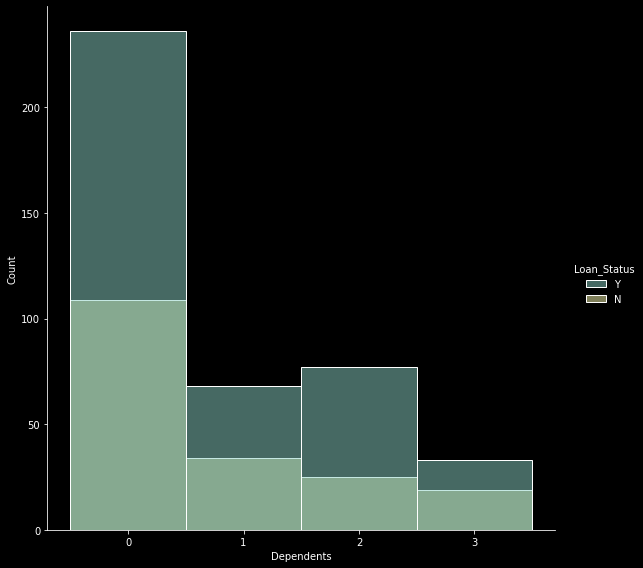
1. About 80% of the applicants were male.
2. Gender didn’t impact the loan status.



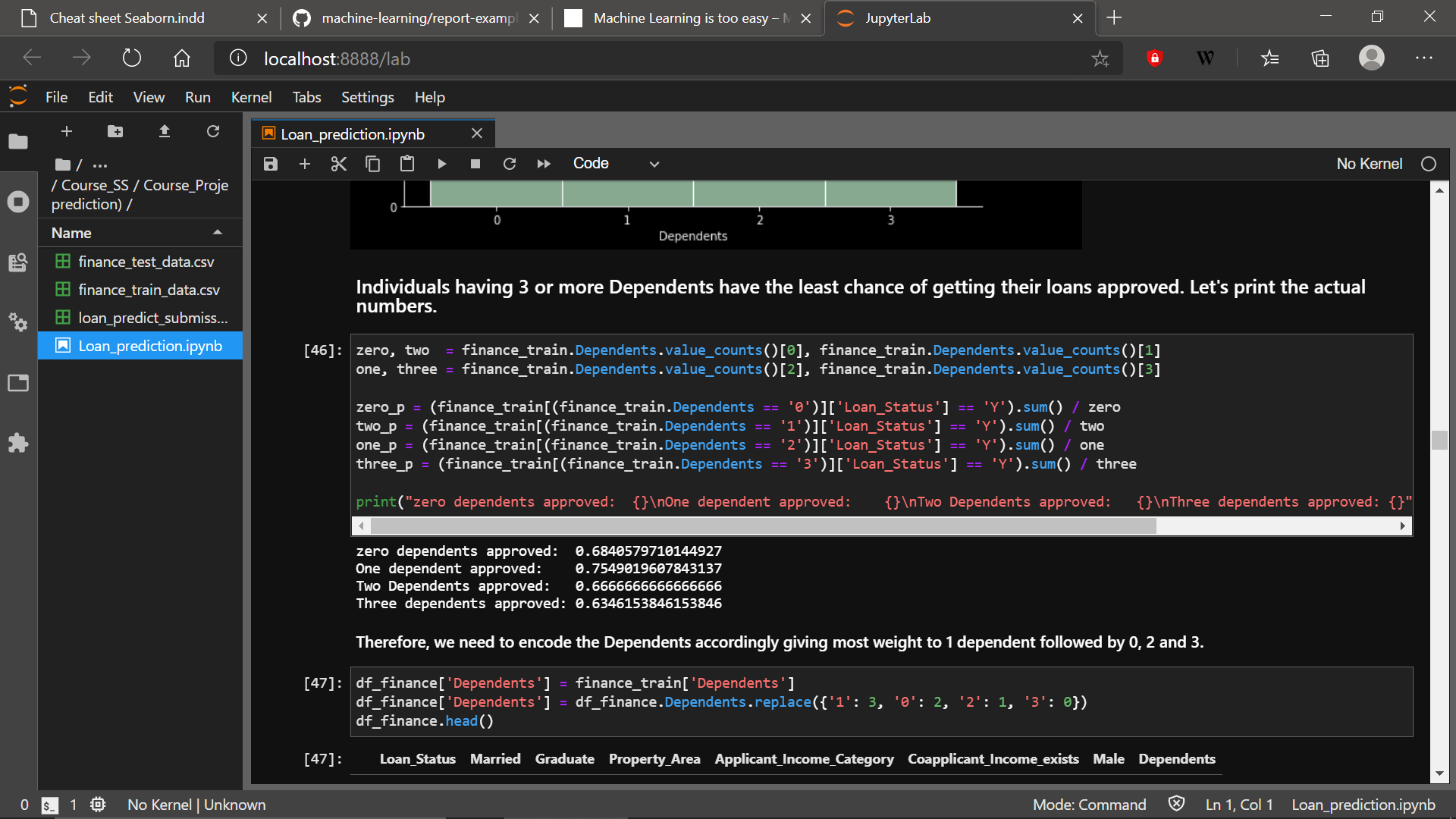
1. Gender had some missing values. Since it’s a categorical feature which is highly skewed towards the male population, imputing the missing values in the 80-20 male-female ratio appears to be the best way to proceed.

## Feature-8: Dependents (0,1,2,3+)

1. Majority of individuals applying for Loan had 0 dependents.
2. Individuals with 3+ dependents had the least chance of getting their loans approved. Let’s see the result in hard numbers.



About 75% people with exactly 1 dependent got approved, which was the highest. This may indicate that having 1 dependent is ideal to be granted a loan.

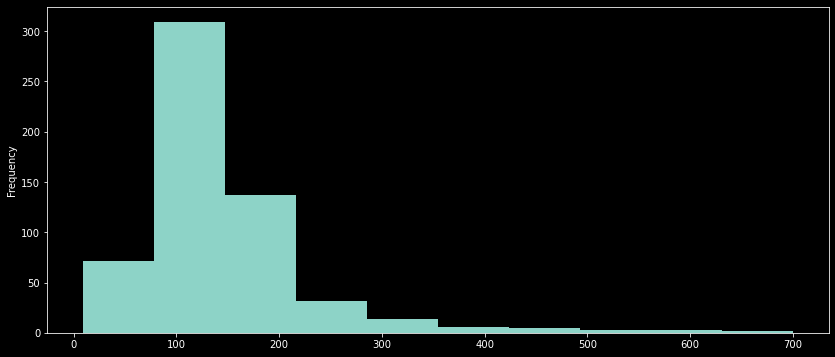


1. This is again a categorical feature and missing values can be imputed by replacing them with values of Dependents in the same proportions as they appear in the training set.

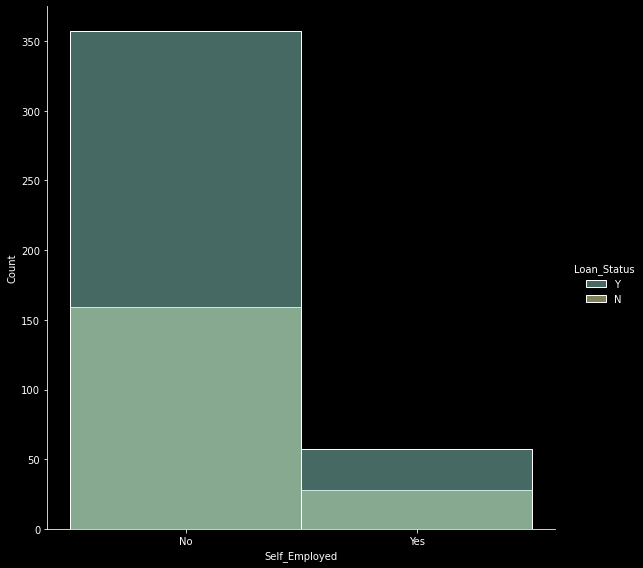
## Feature-9: Loan Amount Term

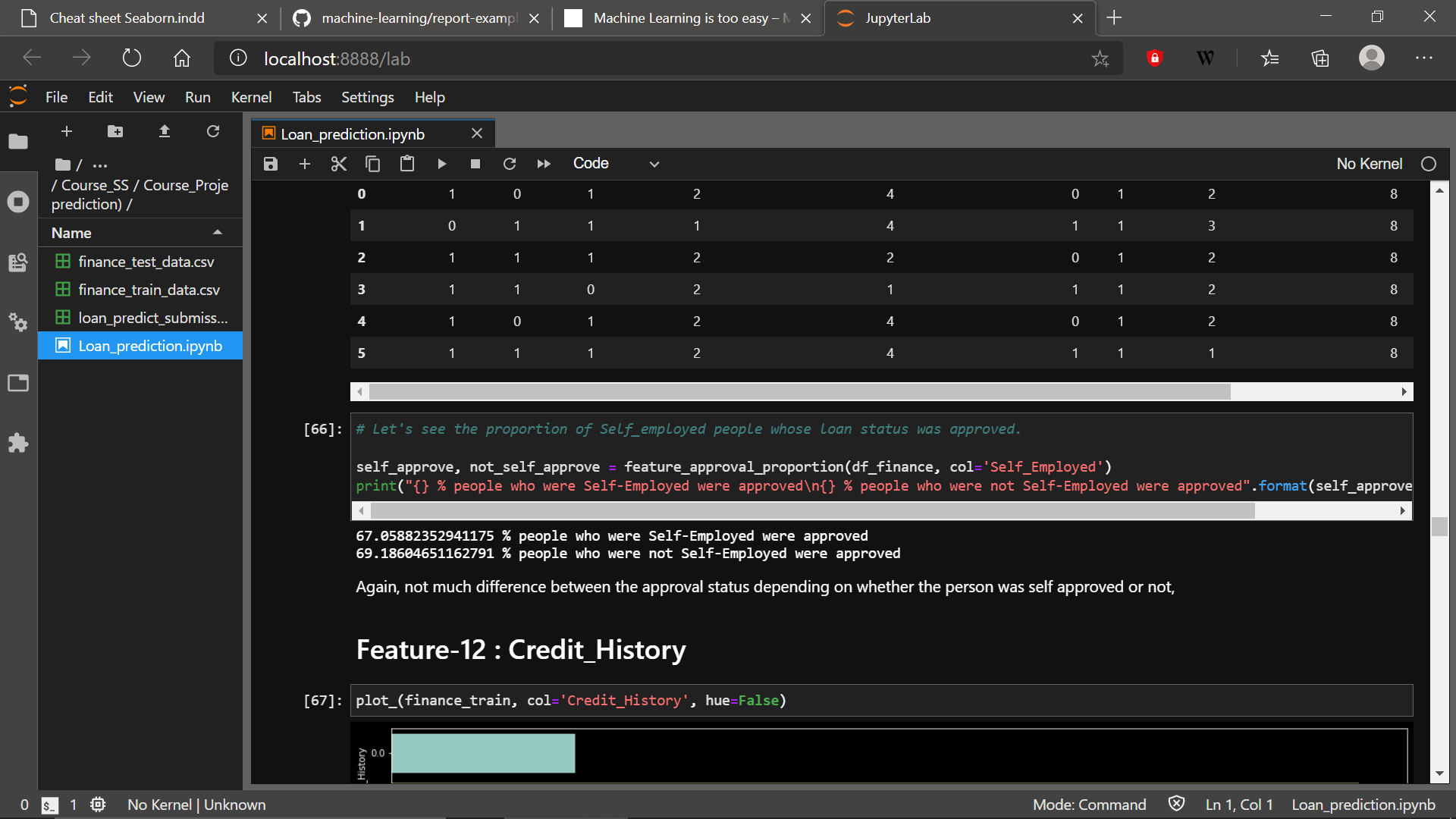
1. Loan Amount term has got 10 unique values with about 80% of the values skewed towards 360 months.
2. There does not seem to be any significant difference between receiving the approval status if to loans are taken for a different tenure. Therefore, this may not be an important feature.
3. This is a categorical feature as well and the missing values can be randomly filled with top 2 most frequently occurring values in the ratio 80:20.

## Feature-10: Loan Amount

1. This is a continuous feature with the following distribution.
2. The Loan Amount is almost Normal with some outliers indicated by the long tail. We need to handle this in the data wrangling part.
3. The null values could be replaced by the median of the distribution as outliers don’t impact the median.

## Feature-11: Self Employed

1. Only about 14% of the applicants are self-employed.
2. Visualizing the impact of Loan approval status on being self-employed.

The visualisation is unclear. Therefore, let’s see the hard numbers.

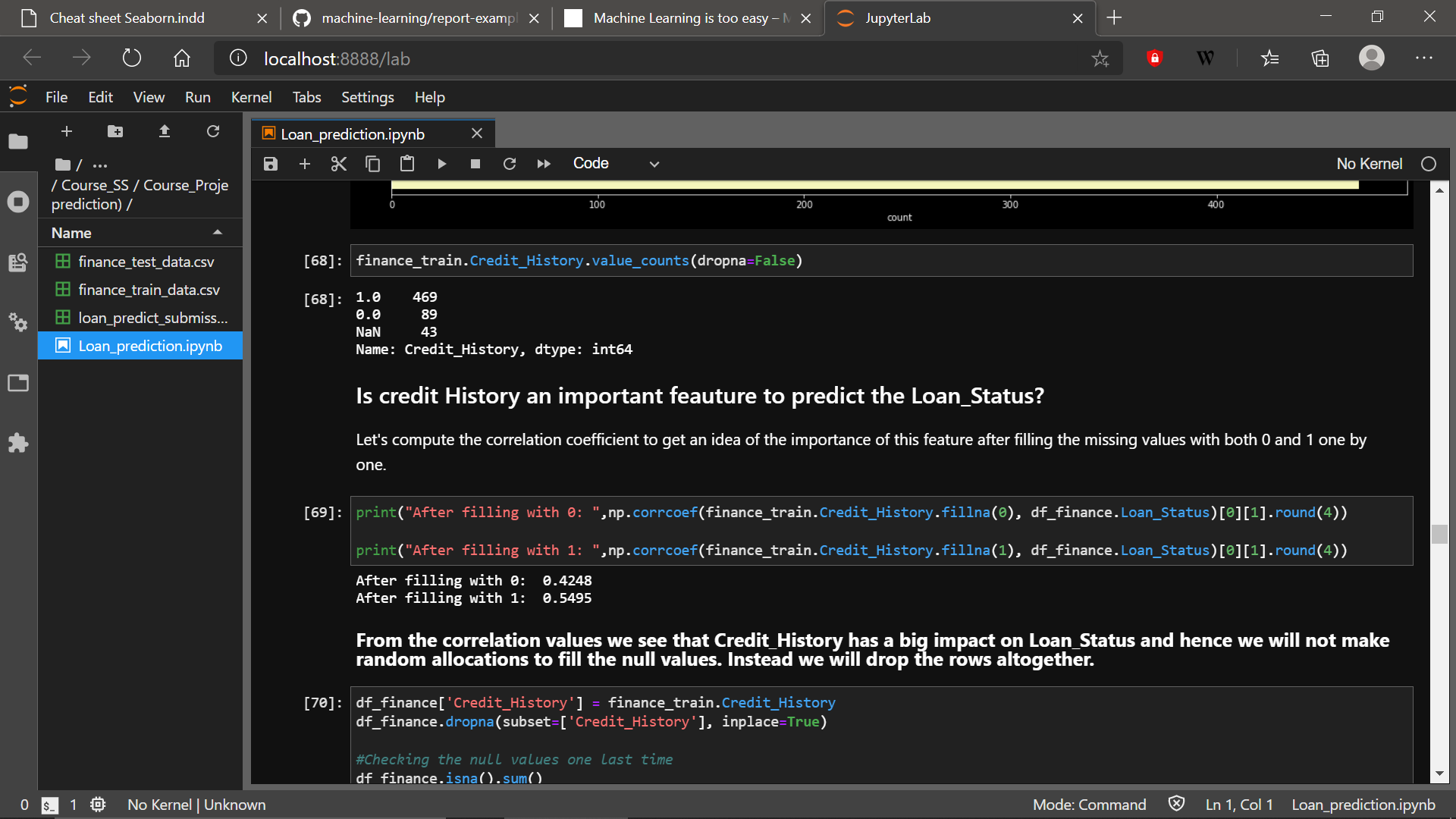
There was not much difference between approval being granted based on the Self-employed status. Therefore, this may not be an important feature.

1. This is a categorical variable and the missing values can be imputed in the same proportion as they appear in the training set.

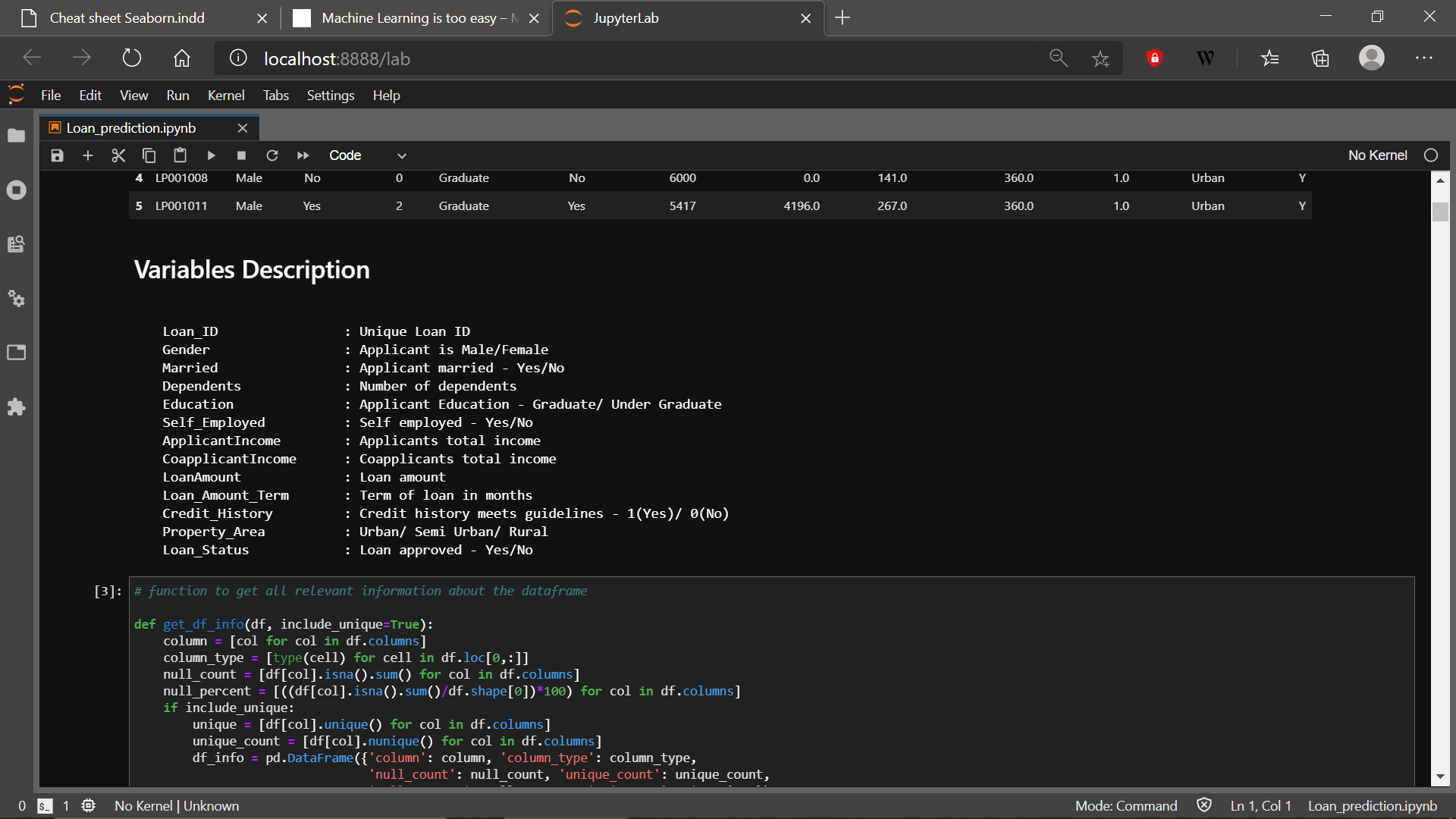
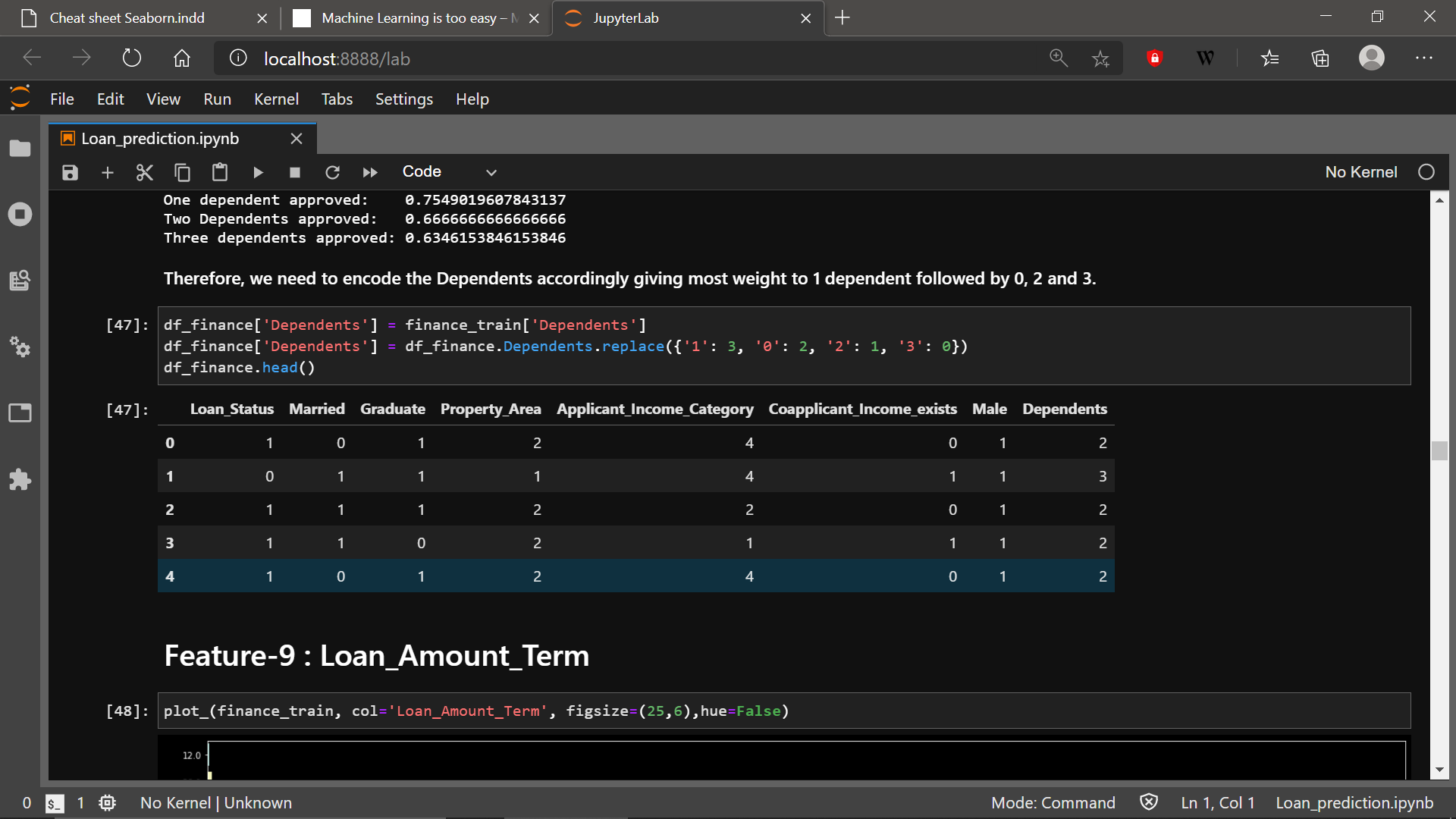
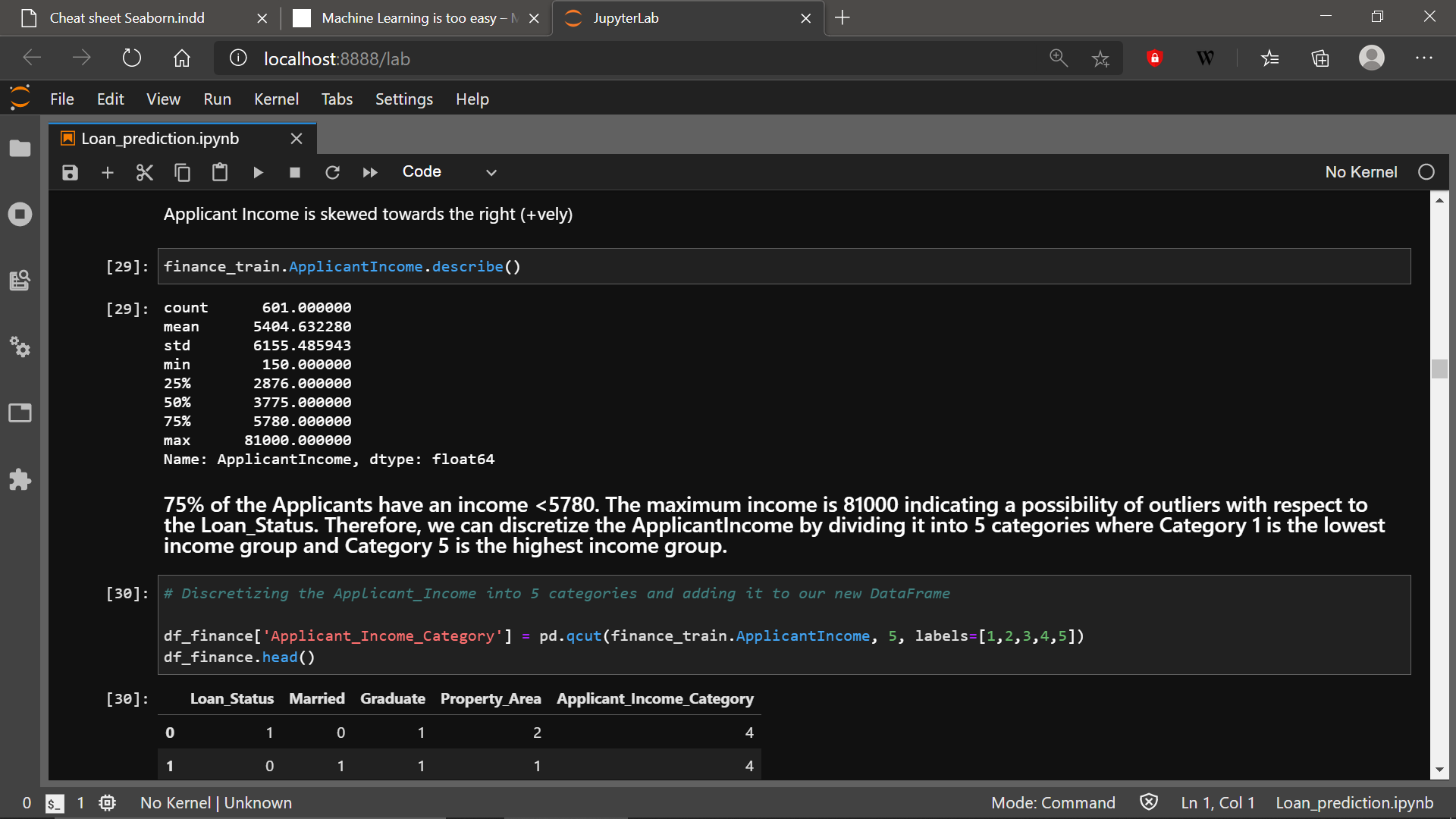
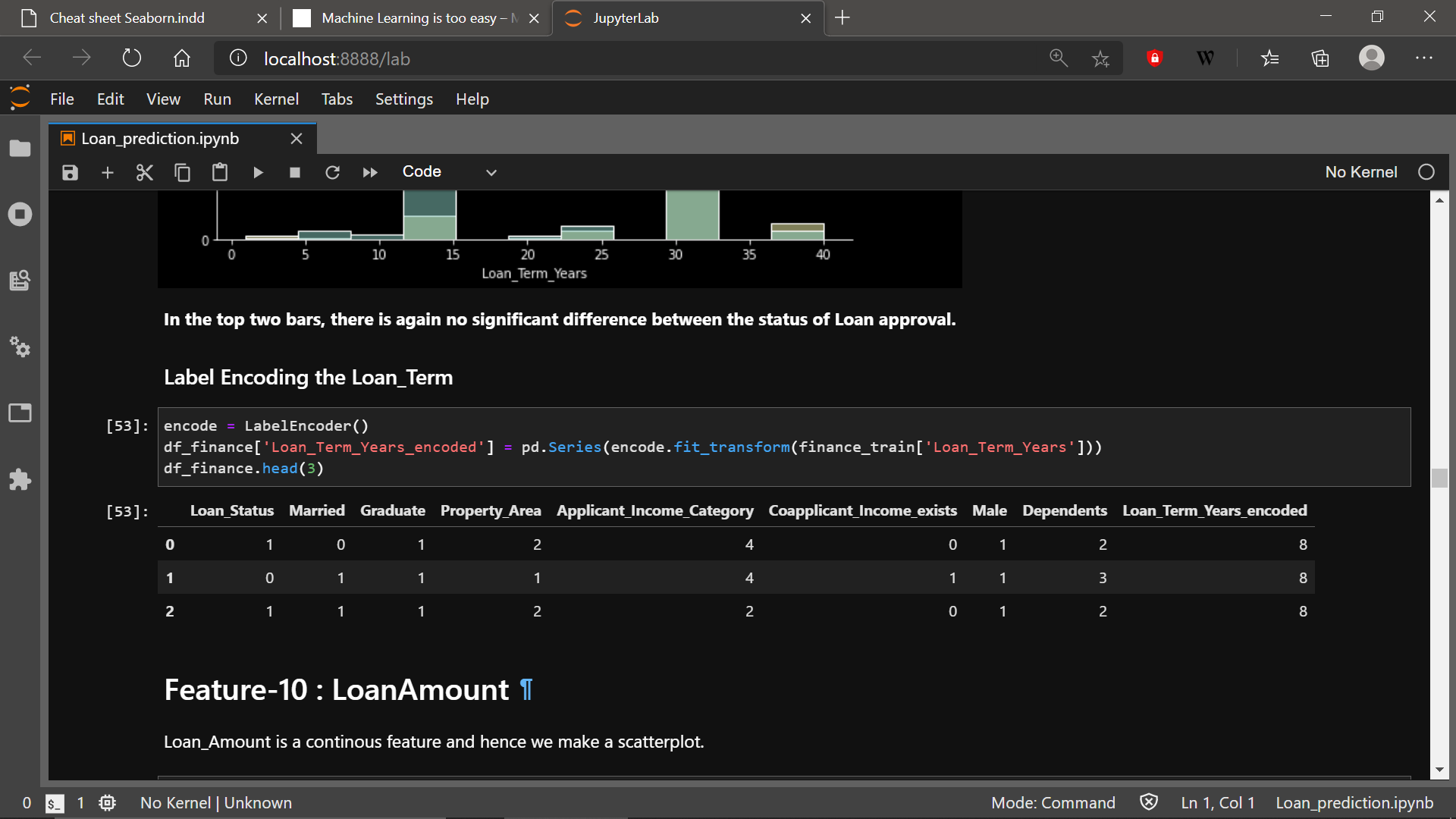
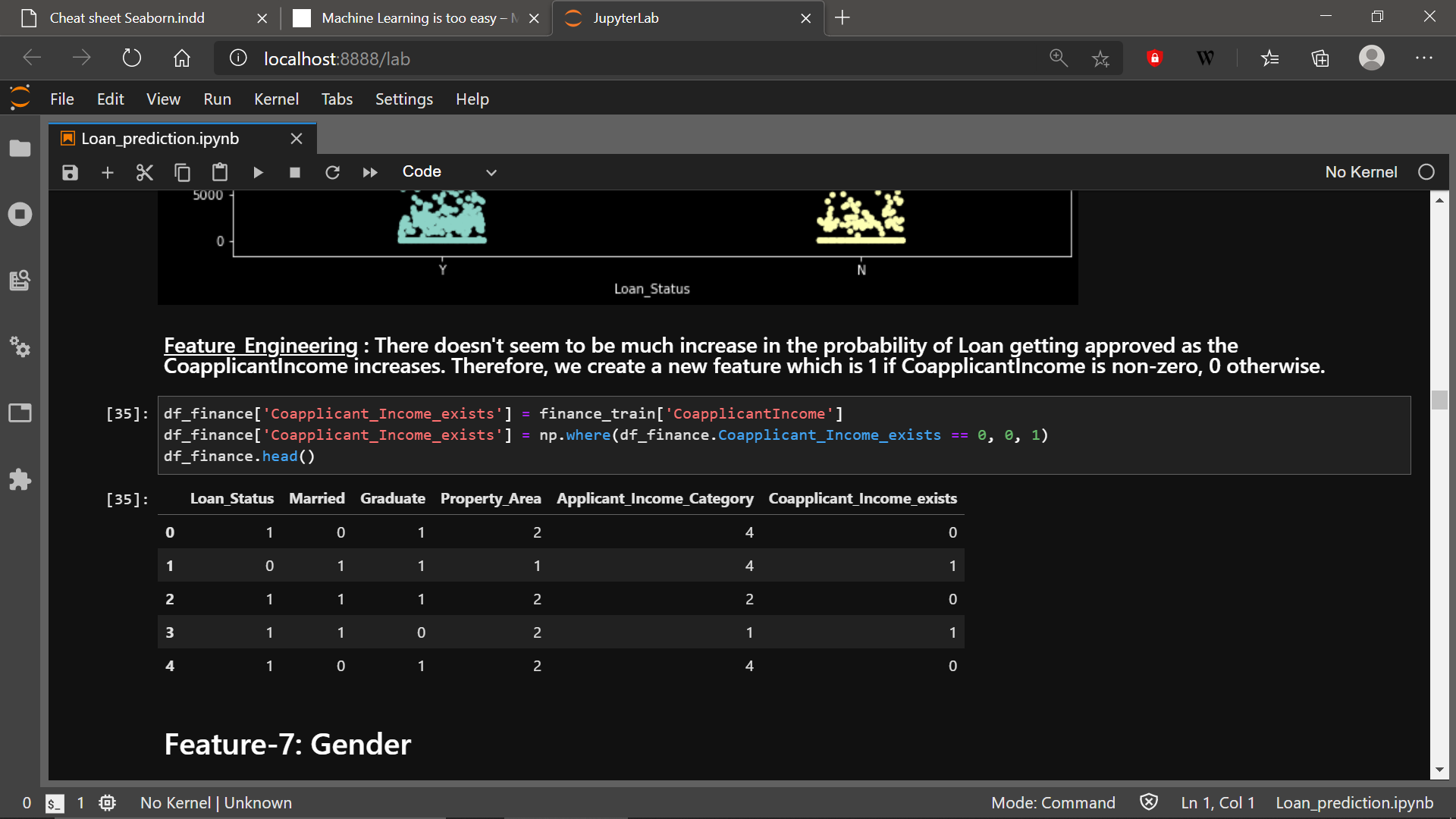
## Feature-12: Credit History

1. Majority of values were 1 meaning that a credible credit history was available



1. From the description, this looks like the most important feature of all but this was also the feature with the most missing values. One way to approximate the dependency of Credit History on Loan Status is to impute the missing values once by 0 , once by 1 and compute the correlation coefficient for both the imputations.
2. The above piece of code shows that the correlation between Loan Status and Credit History cannot be ignored and hence imputing the missing values with random values can possibly mess up with the pattern. Therefore, the best technique here would be to drop the rows which has missing values of Credit History.

# Data Wrangling and Feature Engineering

1. The data set has 8 categorical features and 3 continuous features.
2. Binary features like Gender, Married, Education, Self-employed, Credit-History, Loan status were all One Hot encoded. Example: Gender was encoded 1 if the applicant was male, 0 otherwise.
3. A person with 1 dependent had the highest chance of getting their loan approved. Therefore, while encoding maximum weight was given to applicants with 1 dependent.
4. Applicant Income had outliers. So, discretization of Applicant Income, based on their sample quantiles, into 5 categories was a valid solution.
5. Loan Amount was almost Normally distributed but it had few outliers which we had to deal with. Therefore, after filling the null spaces with the median of the distribution, the feature Loan Amount was also, based on the sample quantiles, discretized into 5 categories.
6. Loan Amount term had only 10 unique values in the form of the number of years. Therefore, it was label encoded.
7. EDA of Property Area showed that individuals living in semi-urban areas were more likely to get their loans approved. Therefore, most weight was given to the applicants coming from semi-urban areas.
8. Finally, the last feature i.e., Co-applicant Income. This feature provided some pretty interesting insights as mentioned in the EDA part. Because the increase in co-applicant income did not play a big role in increasing the chances of approval of an applicant combined with the fact that 44% of the values were exactly 0, **we can engineer a new feature and call it “*co-applicant income exists*”** which is 1 if the co-applicant income is non-zero, 0 otherwise.

# 

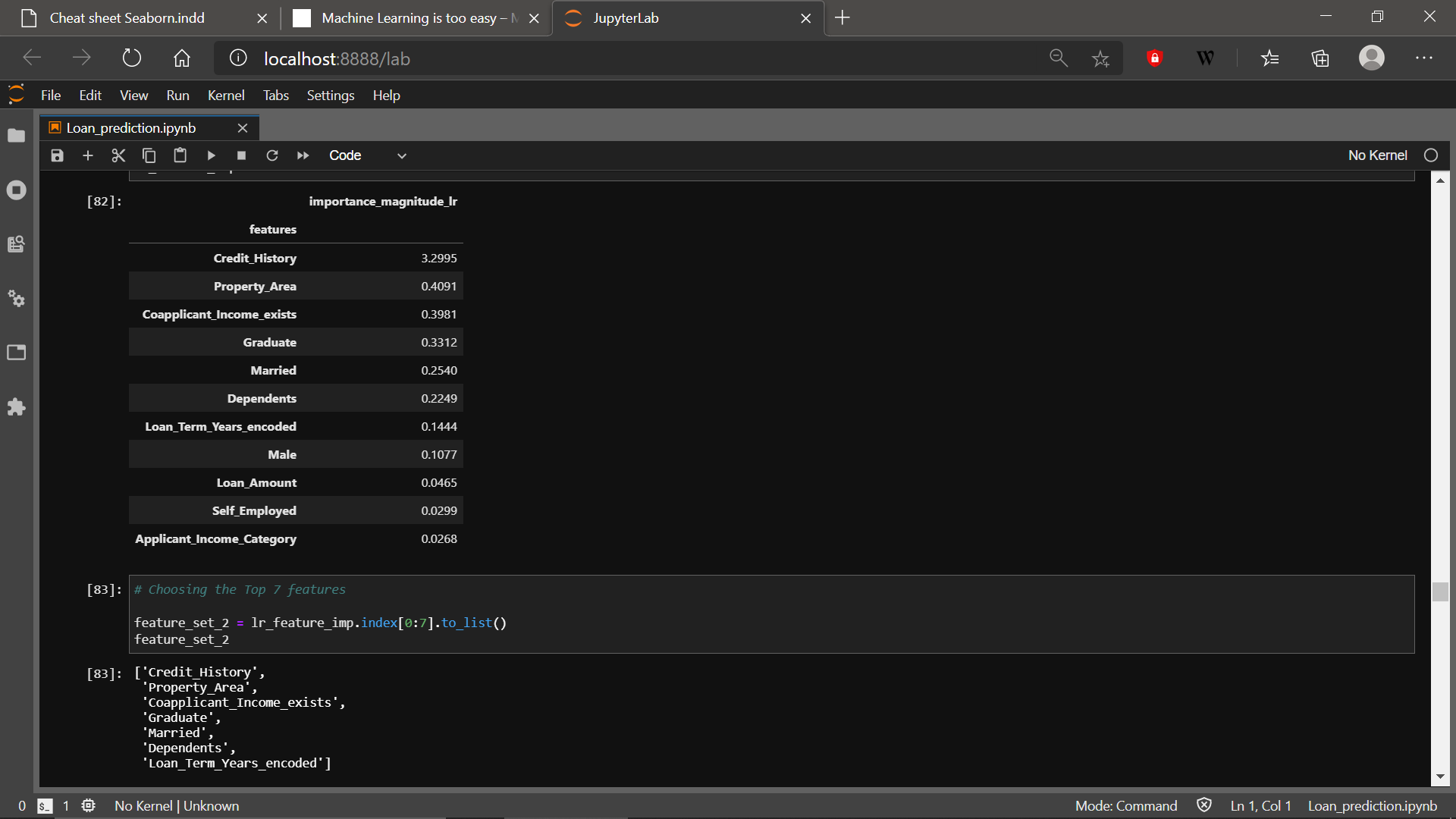
# Algorithms

The algorithms used in this project are the following:

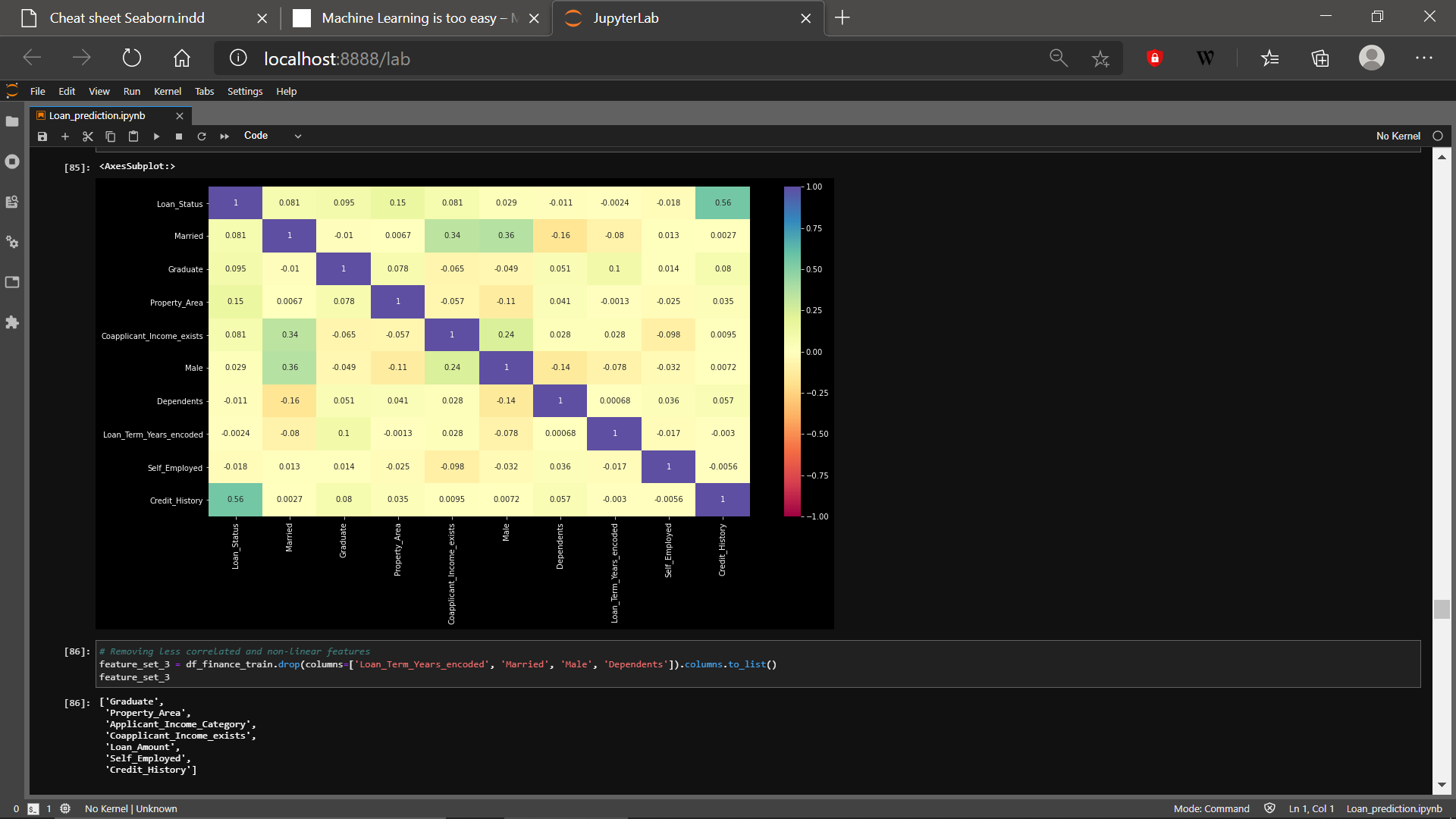
1. Random Forests
2. Logistic Regression
3. Support Vector Classifier
4. Gaussian Naïve Bayes[[2]](#footnote-2)

The number of observations in the training set were pretty small so it made more sense to choose **algorithms with a high bias and low variance** (ii, iii, iv)

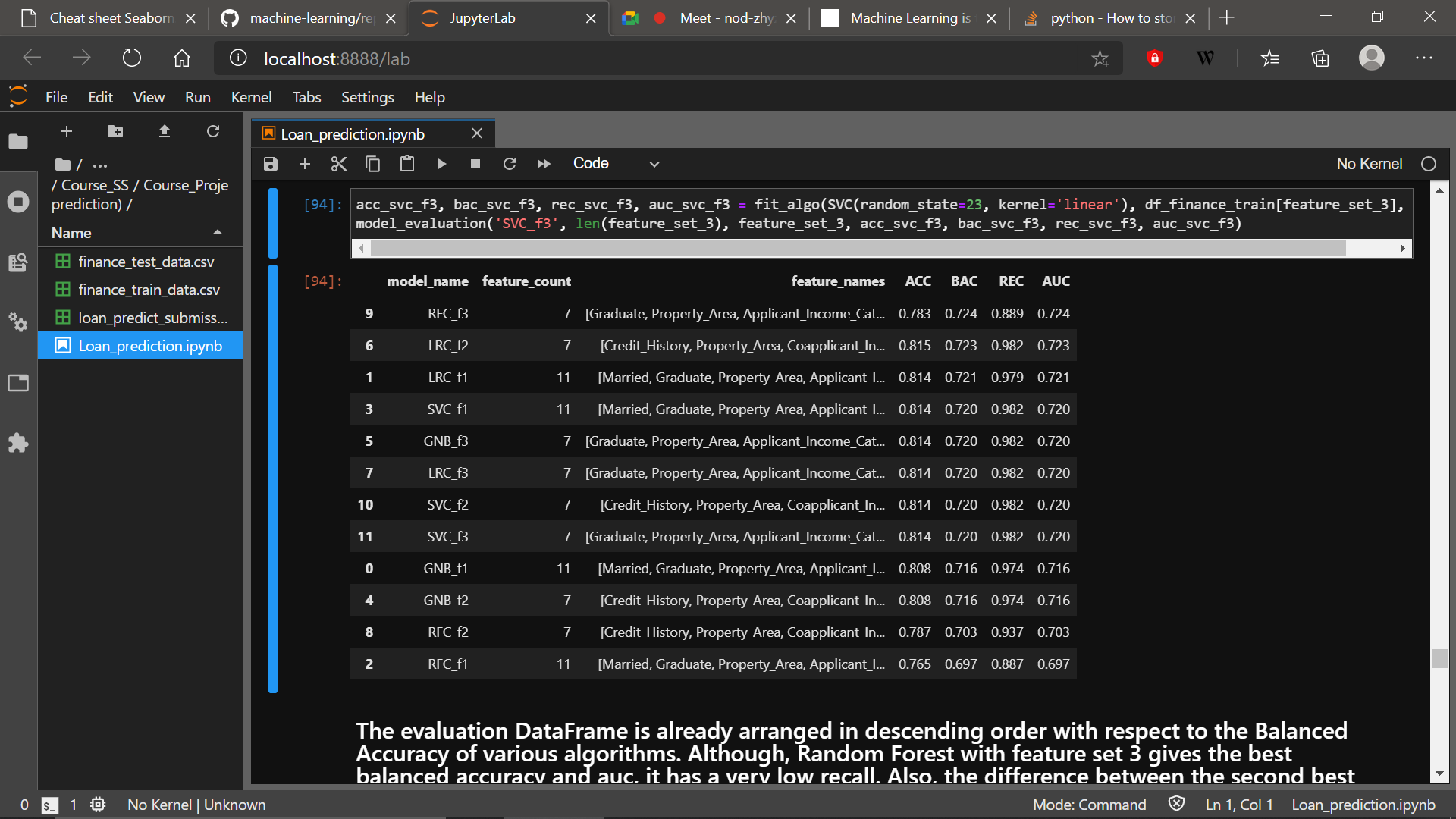
Random Forests serve as a comparison model to see how they perform against the other three.

* Initially, all 11 features were used to evaluate the performance of the algorithms.
* Once, we got the top 2 performing models from the first iteration, their feature importance values were used to create a second feature set

.

* Finally, a third feature set was created based on the correlation analysis of all 11 features. The features that were least corelated with Loan status were dropped and the ones that were highly corelated among themselves were also taken care of.

# Hyperparameter tuning

After experimenting with 4 algorithms on 3 different feature sets, the following table evaluation table was obtained.

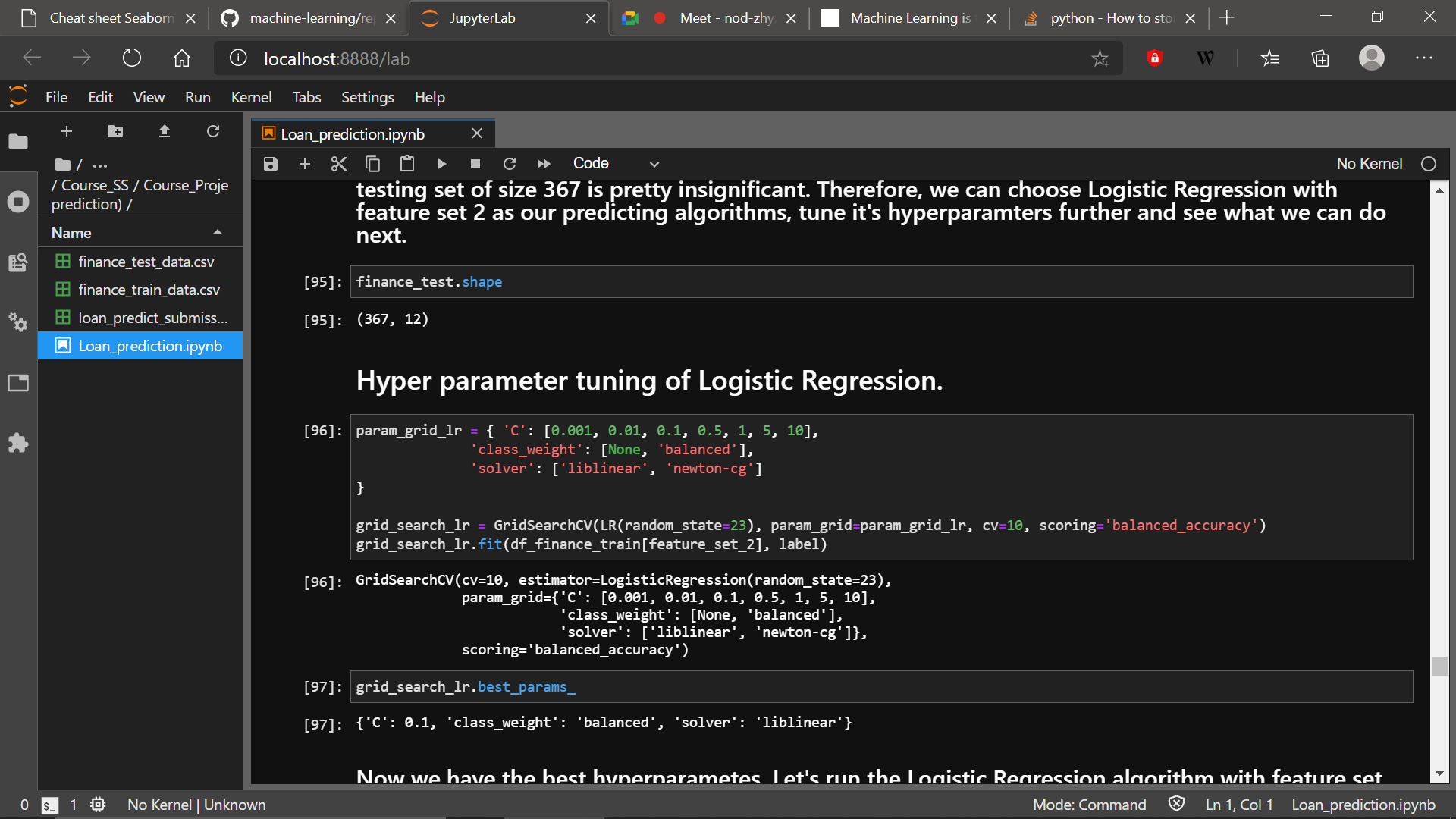
The Top-3 models, when arranged in descending order of their balanced accuracies are:

* Random Forests with feature set 3
* Logistic Regression with feature set 2
* Logistic Regression with feature set 1

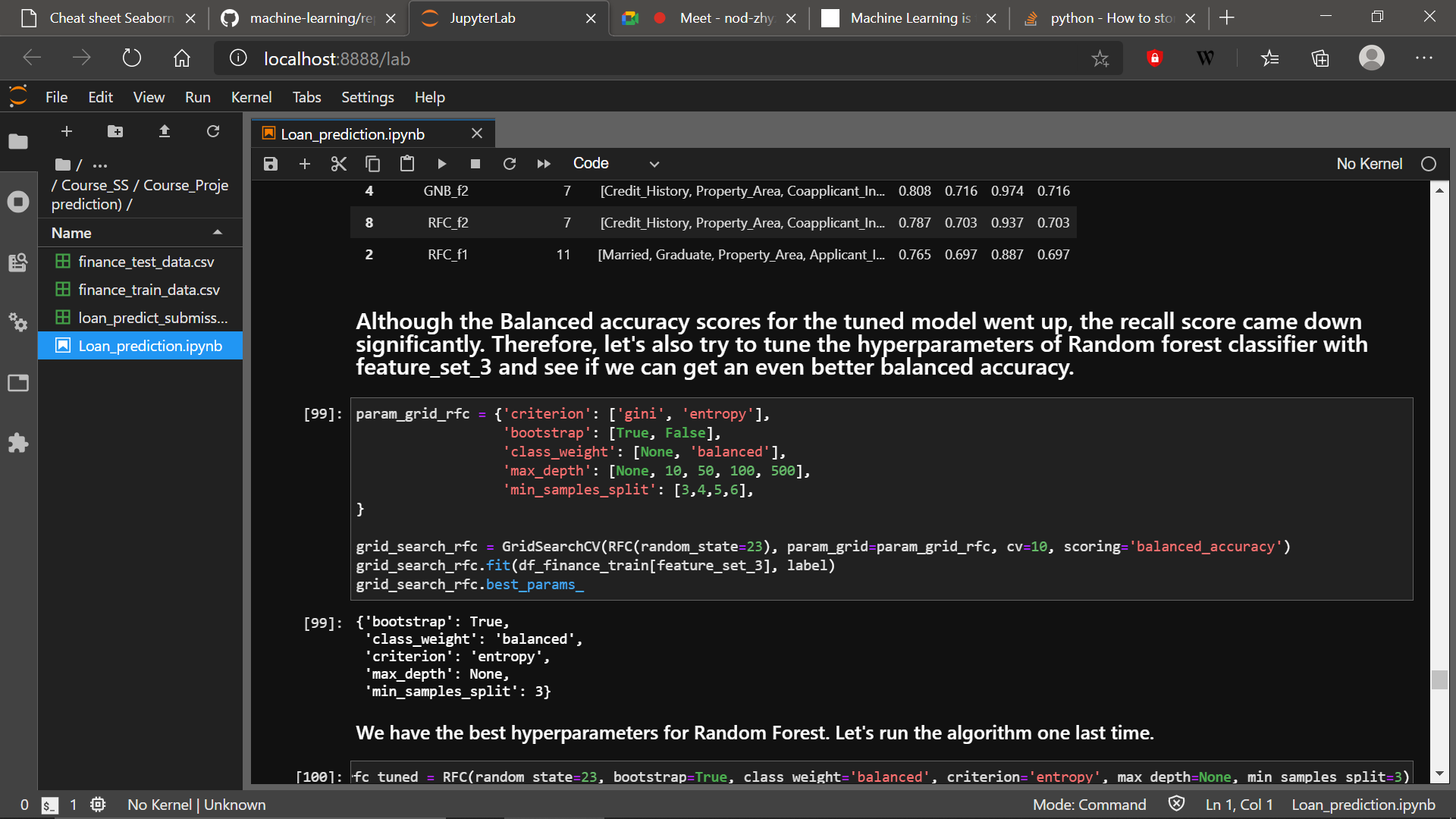
The third model requires all 11 features whereas the top 2 require only 7. Therefore, we can safely ignore Logistic Regression with all features and move towards tuning the hyperparameters of the top two models.

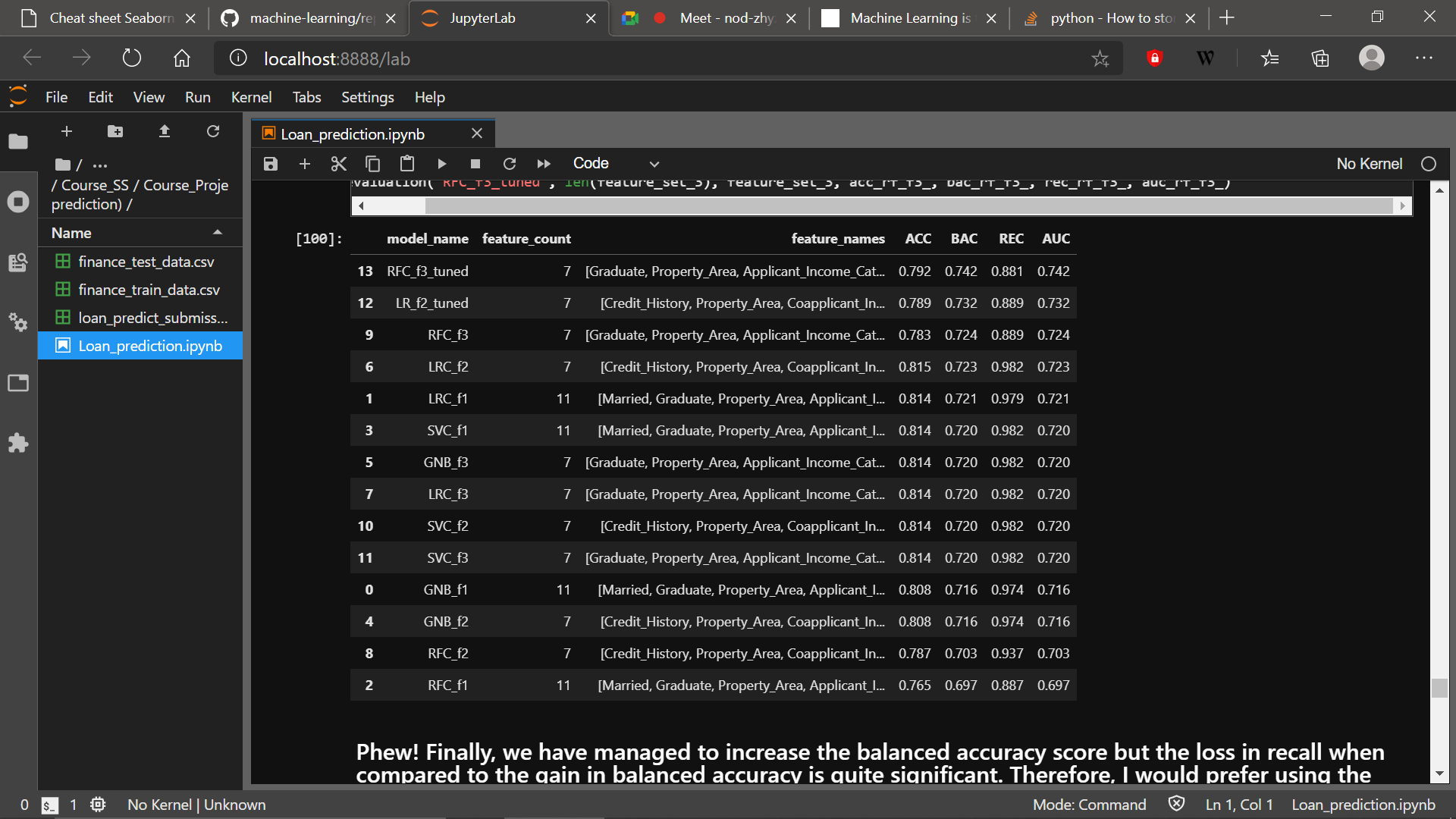
We can use **GridSearchCv** to fine tune the models. The following were the parameter grids for both the models.

1. **Logistic Regression**



1. **Random Forest**



After getting the tuned hyperparameters that fit the training data well, we can test the evaluation scores again to check for the improved values of our performance metrics

So, our GridSearch actually managed to increase the Balanced Accuracy of the Top-2 models but that happened at the cost of Recall scores.

# Result

**Judgement:**  Both Recall and Balanced accuracy are our evaluation metrics. Looking at the final table, I would choose LRC\_f2 or Logistic Regression with feature set 2 to be the model for making predictions on the testing data.

**Reason:** The balanced accuracy score of the topmost model is **0.742** which is higher when compared to **0.723**; the balanced accuracy score of the model of my choice. But the increase is a mere **2%** and for a small testing set of size 347, this increase is pretty insignificant particularly when we factor in the **dip in Recall score values which is about 11%.**

Therefore, I made the predictions on the testing set using a **logistic regression model with feature set 2**.

1. [Loan Definition (investopedia.com)](https://www.investopedia.com/terms/l/loan.asp) [↑](#footnote-ref-1)
2. [The Optimality of Naive Bayes (aaai.org)](https://www.aaai.org/Papers/FLAIRS/2004/Flairs04-097.pdf) [↑](#footnote-ref-2)