DEVELOPING AND EVALUATING PREDICTION MODELS FOR SENSORY ANALYSIS USING VARIOUS DATA SCIENCE TECHNIQUES

BY

SHAW OLUWAGBOUNMI PRECIOUS

STUDENT ID: ST20182956

MODULE CODE: CIS7031

ASSESSMENT ID: WRIT1

PROGRAM: Msc. DATA SCIENCE (INTERNSHIP)

COURSE: PROGRAMMING FOR DATA ANALYSIS

Table of Contents

[1.0 ABSTRACT 3](#_Toc86191156)

[1.1 PURPOSE 3](#_Toc86191157)

[1.2 METHODOLOGY 3](#_Toc86191158)

[1.3 BUSINESS IMPLICATION 3](#_Toc86191159)

[1.4 RESEARCH LIMITATION 3](#_Toc86191160)

[2.0 INTRODUCTION 4](#_Toc86191161)

[2.1 AIM AND OBJECTIVES 4](#_Toc86191162)

[3.0 METHODOLOGY 5](#_Toc86191163)

[3.1 MACHINE LEARNING PROCESS 5](#_Toc86191164)

[3.1.1 Problem formulation 5](#_Toc86191165)

[3.1.2 Data Collection 5](#_Toc86191166)

[3.1.3 Data Processing 5](#_Toc86191167)

[3.1.4 Data Splitting 6](#_Toc86191168)

[3.1.5 Model selection and training 6](#_Toc86191169)

[3.1.6 Model Evaluation 7](#_Toc86191170)

[3.1.7 Hyperparameter tuning 7](#_Toc86191171)

[3.1.8 Final Model 7](#_Toc86191172)

[3.2 JUSTIFICATION OF TOOLS AND TECHNIQUES USED 7](#_Toc86191173)

[4.0 ANALYSIS AND INTERPRETATION OF RESULTS 9](#_Toc86191174)

# 1.0 ABSTRACT

## 1.1 PURPOSE

The purpose of this study is to provide smart phone producers with an idea of the activities their customers perform while using their mobile phones.

## 1.2 METHODOLOGY

Machine learning methods and neural networks for the data analysis and Python for visualisations.

## 1.3 BUSINESS IMPLICATION

The business implication of the study is that it might provide smart phone producers with ideas on the phone features to be improved on, upgraded or developed based on the activities carried out by the user.

## 1.4 RESEARCH LIMITATION

The limitation of this study is that it only analyses the data of one smart phone user carrying out three activities. In a case where data on an activity not present in this study is introduced, the model would not be able to predict the activity.

# 2.0 INTRODUCTION

Most smart phone producers produce and release a new version of phones every year and new software updates every now and then. Each year the new phone always has a feature that distinguishes it from the previous one.

**How do the producers know what feature to include in the software update or new phone?**

There are various ways the producers can achieve this, such as: requiring the users to fill a questionnaire, social media polls, sensory data etc.

**Why sensory analysis?**

Sensory analysis provides answers to crucial questions regarding a company’s products that translate directly to profits and customer satisfaction. Sensory analysis is not dependent on conjecture or assumptions, it is in fact a scientific method invented to remove the assumption from developing products for consumers. Companies committed to understanding the likes and dislikes as well as the usage of their product(s) take advantage of sensory analysis in making major decisions that influence millions of dollars of revenue and the overall success of the company.

**Why machine learning methods?**

Machine learning is the process of training systems to make precise predictions when fed data. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions and predictions with minimal human intervention.

These predictions could be predicting whether a given picture of fruit is a mango or an orange, online recommendation offers such as those from Amazon and Netflix, identify that people are crossing the road in front of a self-driving car, whether an email is spam or not, accurately recognize speech enough to generate captions for a YouTube video or google search.

## 2.1 AIM AND OBJECTIVES

The aim of this study is to analyse sensor data collected using MATLAB mobile, create models to predict the activity of the user and select the best model for the prediction.

The objectives of this study are:

* explore and analyse the data
* create a predictive model using Decision trees
* create predictive models using Neural Networks
* determine the best fit predictive model.

# 3.0 METHODOLOGY

## 3.1 **MACHINE LEARNING PROCESS**

### 3.1.1 Problem formulation

The first step in machine learning process is problem formulation or problem definition.

* In this example, we want to investigate "what factors/variables affect a good or a bad loan"
* Make predictions on the activity carried out by the user based on their movement
* Hence our Dependent variable (y) is the Activity (Walking /Jumping /Swinging). We want to the model predict if the user is walking, jumping or swinging.

### 3.1.2 Data Collection

The data analysed in this study was derived from mobile MATLAB sensors for Acceleration, Angular Velocity, Magnetic Field and Orientation across the X, Y and Z axis for each sensor.

The sensor data for each activity is saved in four xlsx files. For example, `the walking activity has its sensor data saved as:

* Walking\_Acceleration. xlsx
* Walking\_AngularVelocity. xlsx
* Walking\_MagnecticField. xlsx
* Walking\_Orientation. Xlsx

The sensor data for the other two activities are also saved in a similar manner. The next step is to load the data just as seen below.

### 3.1.3 Data Processing

There are three steps in processing data, namely:

* **Data exploration**

This process in involves getting to know you data set, understanding the relationship between the features and observations, checking for null values and missing values. Examine the distribution and correspondence of the features in the data set.

The .info() function was used to check the data type of the column, this is to make sure that the column is correspondence with the data type.

The .isnull().sum() function was used to check for null values and it found three null values on row 2400.

* **Data cleaning**

If null or missing values have been detected, visualise their row and column so as to determine if the rows will be dropped or manipulated. Missing or null values can be manipulated by replacing them with one of the following: the mean, median, the value before or after it.

From the data exploration, we found out that there are 3 null values in the Acceleration data across the three axis. To enable our model run smoothly, the row with the null values was dropped.

* **Feature selection**

This is the selection of the features that affect the output of the independent variable. In this case, all the features are selected except ‘Timestamp’ because they all play vital roles in determining the activity of the user. For instance, if we decide to remove Acc\_X which is the acceleration of the user on the x-axis and the model is fed with data that has that variable, the model will not know what to do with that data because it was not trained with that variable. Hence, this is a classification problem.

The features for this model are: Acc\_X, Acc\_Y, Acc\_Z, AngV\_X, AngV\_Y, AngV\_Z, Mag\_X, Mag\_Y, Mag\_Z, Orin\_X, Orin\_Y and Orin\_Z.

* **Data normalization**

This is the process of converting every entry in the data set to into values between 0 and 1.

**Why normalize data?** Normalized data helps keep the data organised, it reduces the redundancy of the data set and it ensures data consistency and flexibility of the data sets.

A function getnorm () was defined to normalized all the X variables. We can see the difference in the first five elements of the un-normalized data and normalize data below:

### 3.1.4 Data Splitting

This involves splitting the data into variables X and y, where X is the features of the model (independent variable) and y is the dependent variable (the variable to be predicted).

The X and y data is further split into three data sets: training, testing and validation data sets. The train data set is used to teach the model to recognize the data, the testing data set is used to test how the model will predict using data sets it does not recognize and the validation is also used to test the predictions of the model on a newer data set.

In this project, the X and y variables are split into 80% and 20% for the training and testing data set respectively. The training data set is further split into 90% and 10% which leaves the training data set with 70% and the validation data set with 10%.

### 3.1.5 Model selection and training

The type of model used is a function of the nature of the problem. To determine the type of model to be used, we have to understand our y variable and the type of data it contains. There are different types of models such as: classification, regression and unsupervised learning.

In this study, the problem is a classification model because we are want the model to assign a class or label of activity to the data it reads. We are not using the linear regression model because the prediction does not depend more on a particular variable than the other.

Some examples of classification problem are: k-Nearest neighbors, decision trees, naive Bayes, random forest, gradient boosting.

We will be making use of decision trees and neural network to build our predictive model.

### 3.1.6 Model Evaluation

This is the use of metrics to evaluate your model on the 20% unseen testing data. There are quite a number of metrics that can be used to evaluate your model like: R squared, accuracy, log loss, confusion metrics, precision, recall, mean square error, etc.

In this study, we are making use of the accuracy metric because it is classification metric. It calculates the percentage of how accurately the model is able to predict the activity of the user.

We also explored the overfitting of the model. Overfitting is when you either feeding the model more or less data you should. Here are the conditions for over fitting:

1. Overfitting: if training loss << validation loss
2. Underfitting: if training loss >> validation loss
3. Just right: if training loss ~ validation loss

### 3.1.7 Hyperparameter tuning

This is the exploration and selection of the optimal model structure. Parameters such as learning rate, epochs, hidden layers, hidden units and activation functions are explored to select to optimal parameters to train the model with.

### 3.1.8 Final Model

This is the model with the optimal parameters to best predict the activity of the phone user based on the sensor data collected.

## 3.2 JUSTIFICATION OF TOOLS AND TECHNIQUES USED

**3.2.1 Activation**

There are different types of activation functions such as Softmax, Relu, Sigmoid etc. In this study, we have three layers; relu activation function was used for the first two layers and softmax was used for the outer layer.

**Why is the activation function of the last layer different?**

For multi-class classification problems where class membership is required on more than two class labels, softmax is used as the activation function. In the case of our study, our classification problem requires three class labels to be predicted hence, the reason for using softmax.

**3.2.2 Metrics**

In this report, the metric used was accuracy. There are other metrics like the mean squared error (mse), precision, recall, etc. but accuracy was chosen because we want to know how correct our model was able to predict the activities of the user based on the data it was trained with. Accuracy tells us how well our model is being trained and how it will perform generally. However, the accuracy metric does not perform well in the case of class imbalance.

**3.2.3 Loss function**

As part of the optimization algorithm, the error for the current state of the model must be estimated repeatedly. This requires the choice of an error function, also called a loss function that can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation.

Neural network models learn a mapping from inputs to outputs from examples and the choice of loss function must match the framing of the specific predictive modelling problem, such as classification or regression. Also, the configuration of the output layer must also be appropriate for the chosen loss function.

For this study, sparse categorical loss cross entropy function was used because it is suitable for multi-class classification problems and the outer layer activation function works well it. Albeit, the drawback of using sparse categorical loss cross entropy function is that the variables need to be one hot encoded and this could pose as problem when dealing with large datasets.

**3.2.4 Optimizer**

Optimizers are used to train models to reduce its error rate such as Adaptive Moment Estimation (Adam), Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), Adaptive Gradient Algorithm (AdaGrad) etc. In this report, Stochastic Gradient Descent (SGD) was used as the optimizer not because it is best but because unlike Adam, it has a lower probability not to converge to an optimal solution when working with multi-class classification problems.

**3.2.5 Kernel Initializer**

The term kernel initializer is another term for which statistical distribution or function to use for initialising the weights. Kernel initializers include he\_uniform, Xavier etc. We have used he\_uniform in this report because it is simple approach and works well with neural networks.

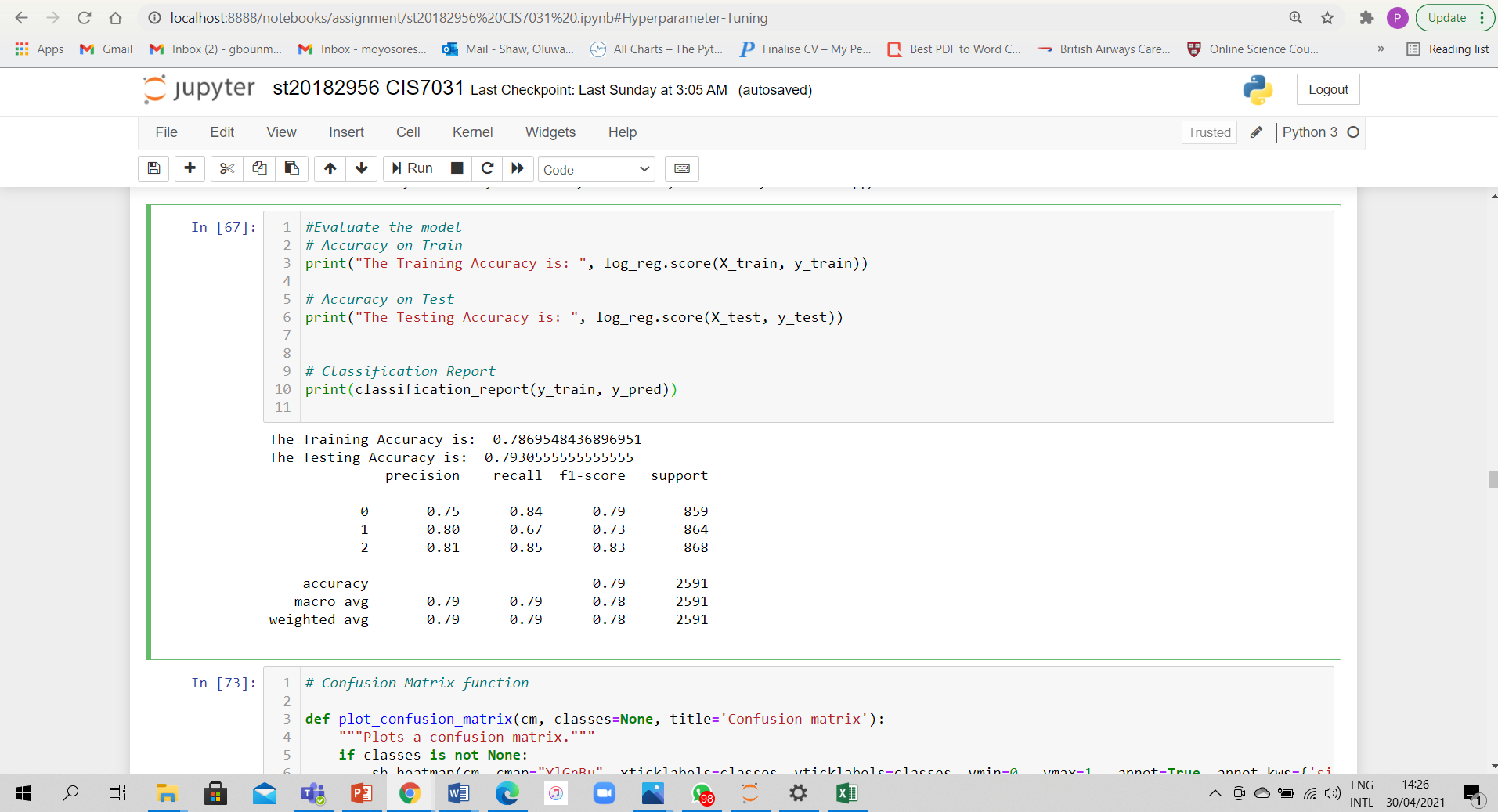
# 4.0 ANALYSIS AND INTERPRETATION OF RESULTS

After all the steps and functions in the above section has been done, we analyse our results.

The first testing and training on the model gave the results below:

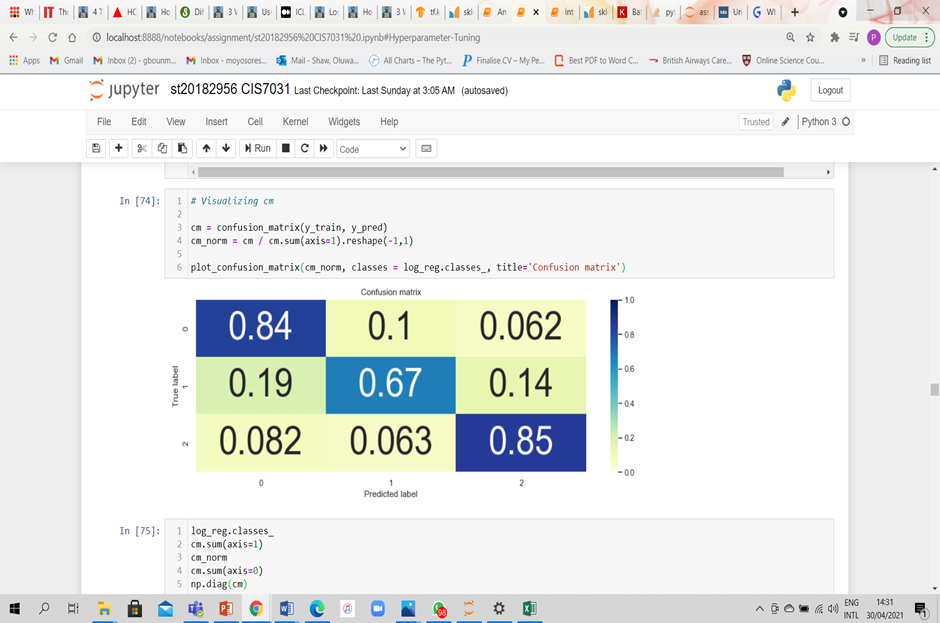
The Training Accuracy is: 0.7869548436896951

The Testing Accuracy is: 0.7930555555555555



We introduced a confusion matrix which is also another metric to measure accuracy. The model predicts 84% of the 0s correctly, 67% of the 1s correctly and 85% of the 2s correctly.

Recall that: 0 = walking activity; 1 = jumping activity; 2 = swinging activity



We also calculated the following values:

The True Positive Rate is: [0.835856 0.670139 0.854839]

The Precision is: [0.754202 0.801939 0.809160]

The False positive rate is: [0.135104 0.082803 0.101567]

The False Negative Rate is: [0.164144 0.329861 0.145161]

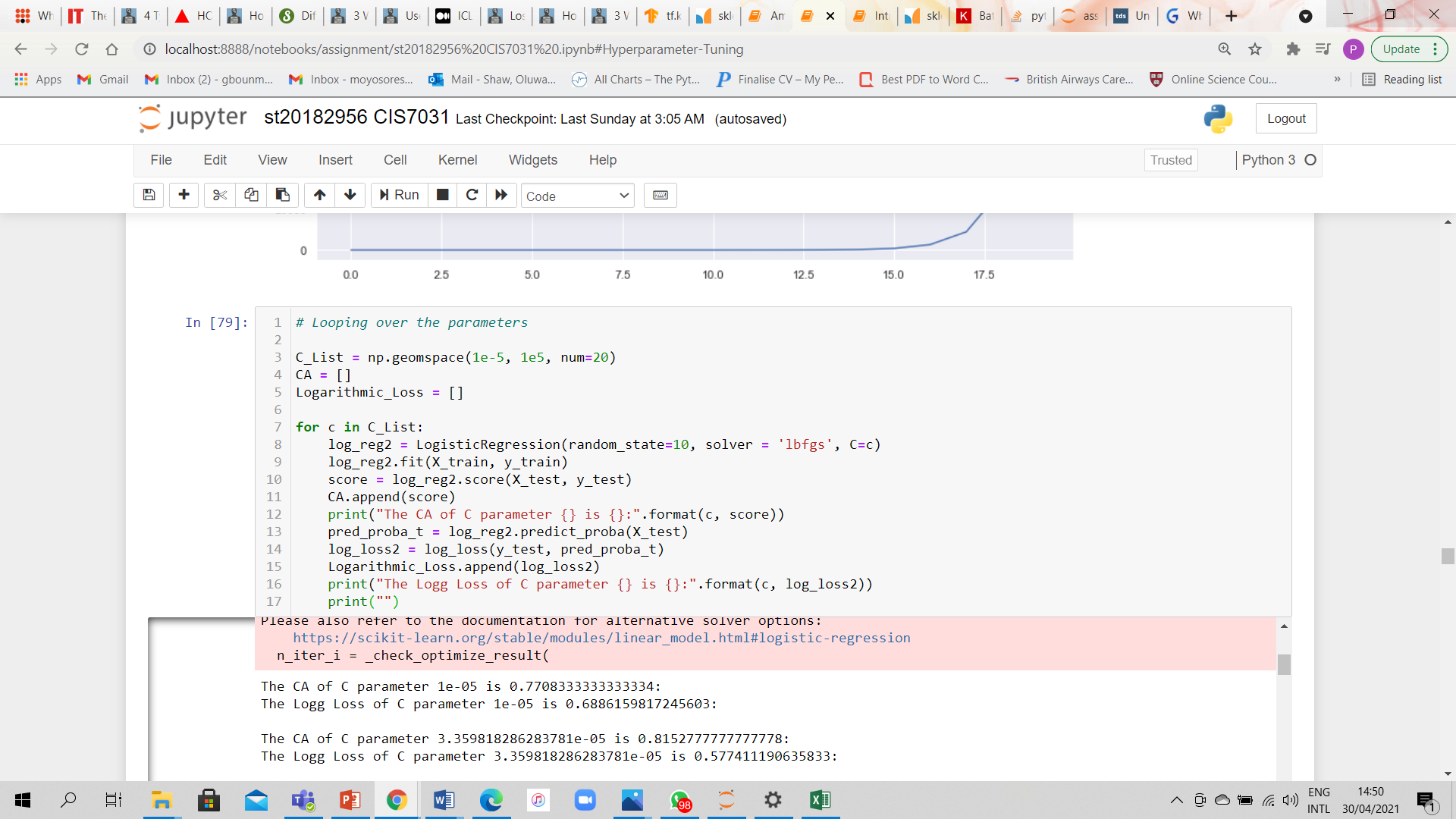
The True positive rate: this is the percentage of true values the model predicted as true

The precision: this is how correct the values the predicted are

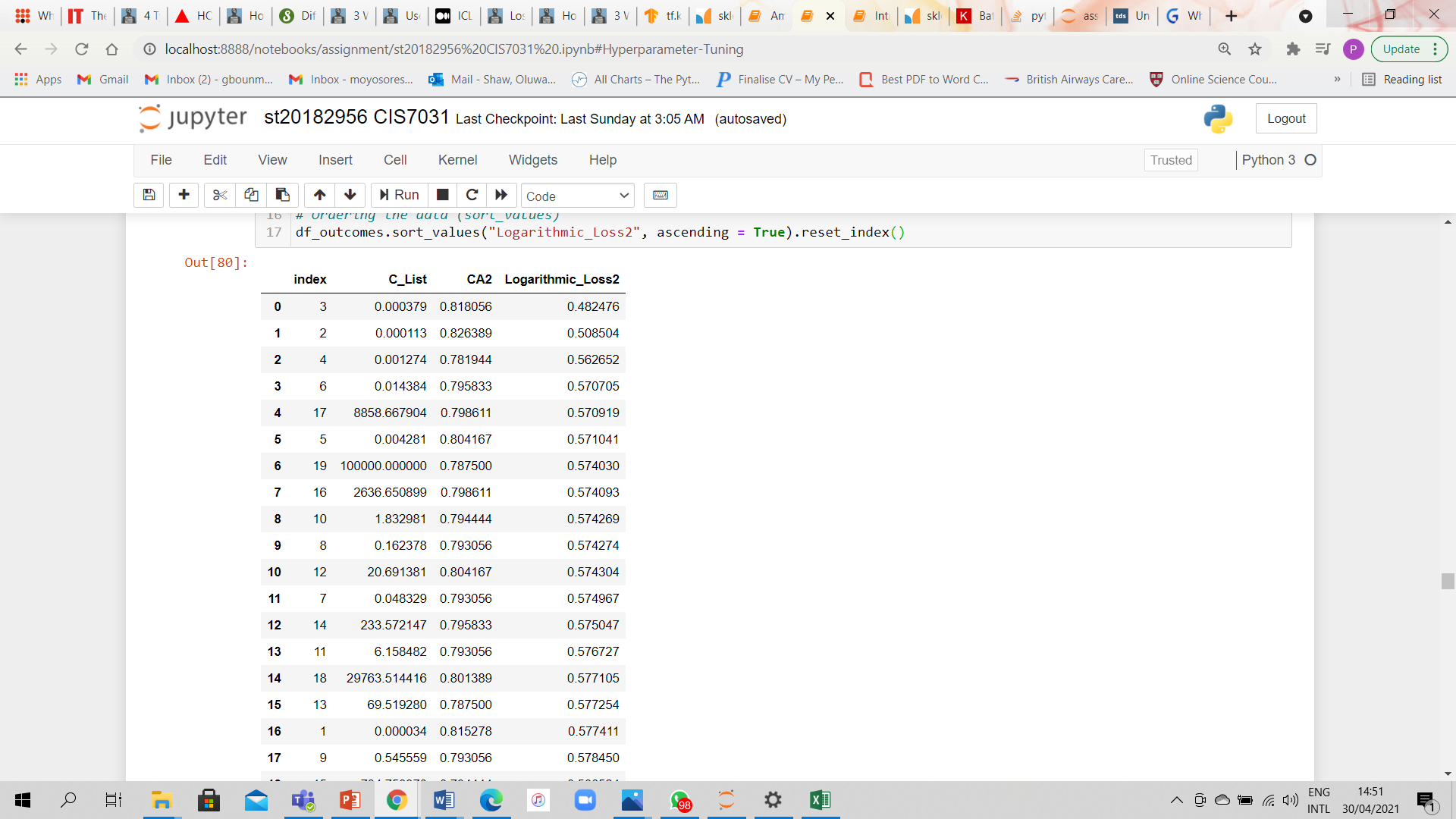
The false positive rate: this is percentage of false values that was predicted to be true

The false negative rate: this is percentage of true values that were predicted false.

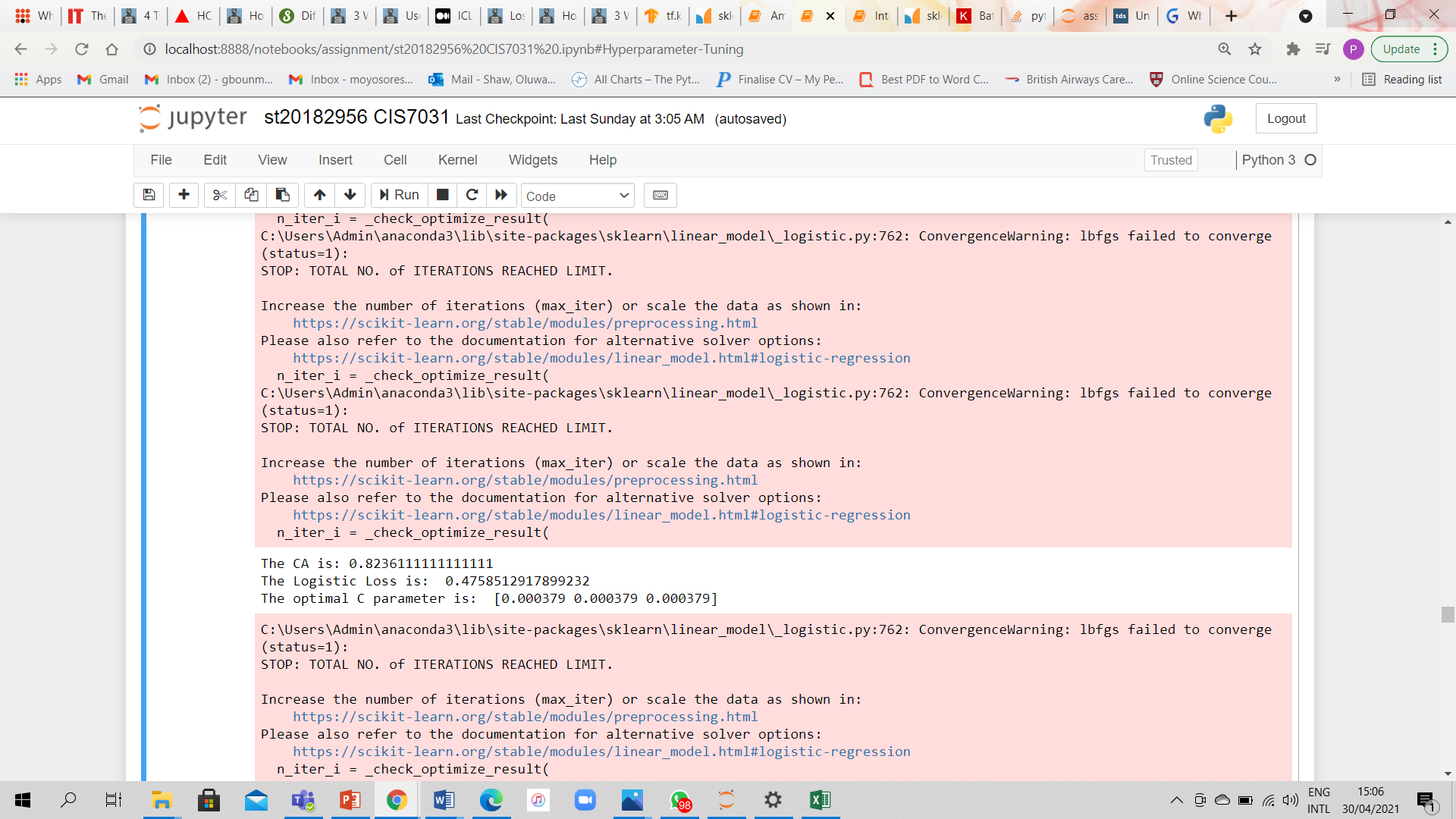
We know that are model is good because the TPR is high and FPR is low.



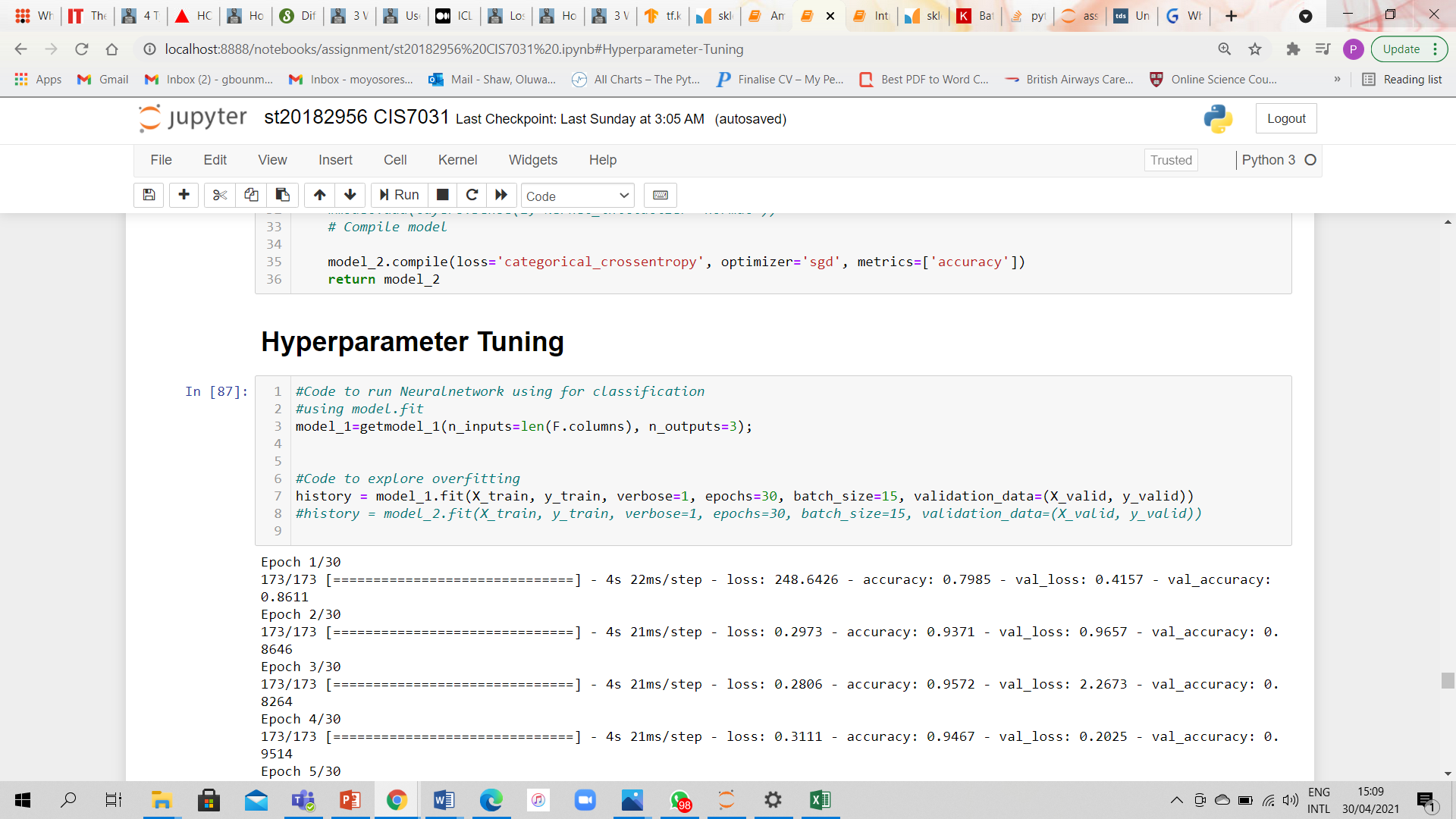
In the above picture, we manually computed the hyperparameter tuning and the picture below, we passed it to a data frame arranging the log loss in ascending order. We want the log loss to be as low as possible and classification accuracy as high as possible. Though the lowest log loss may not give us the highest CA, we will select the best fit. In our case, we have two possible fits: log loss = 0.482476 CA = 0.818056 and log loss = 0.508504 CA = 0.826389



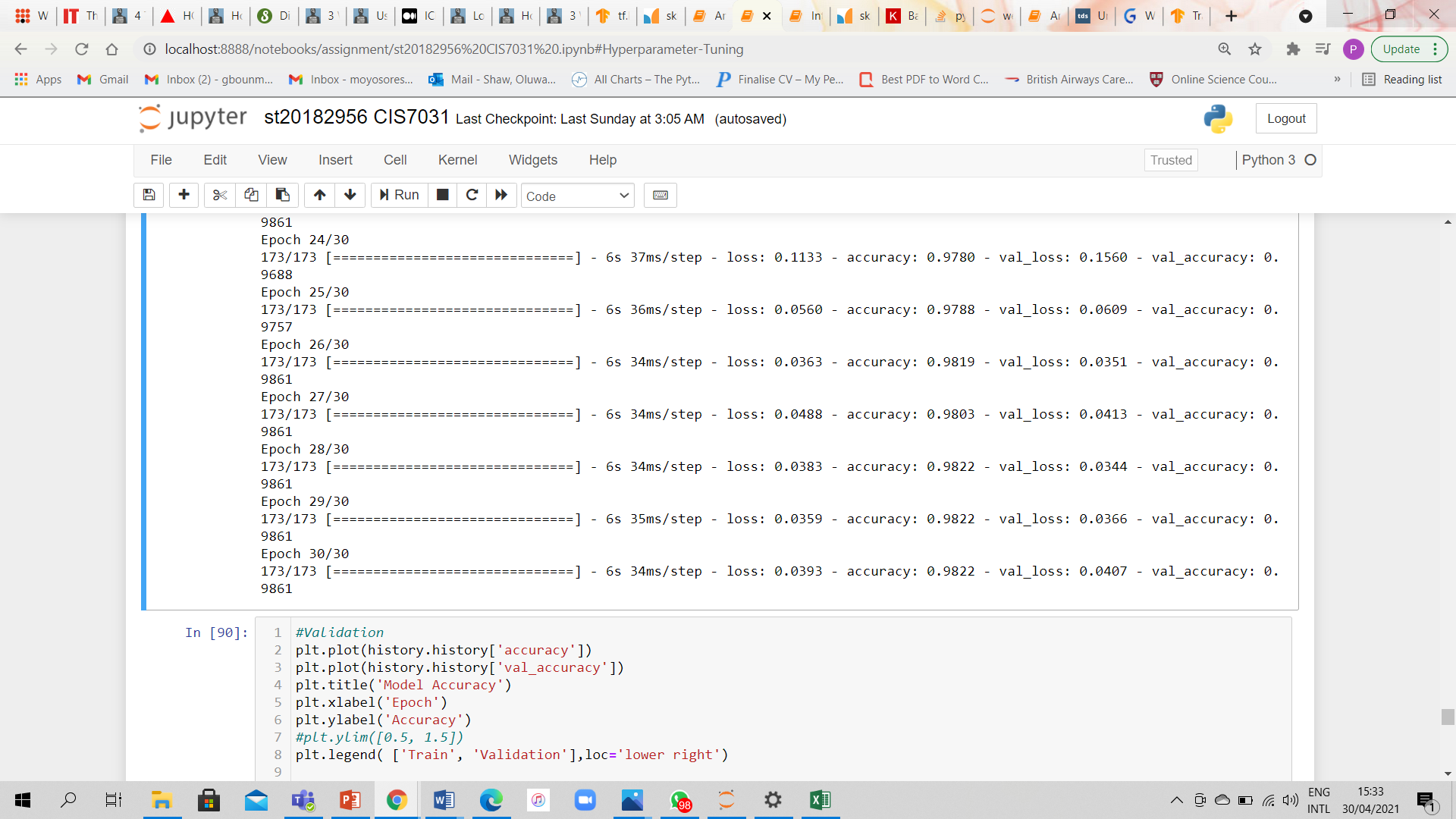
In the diagram below, we computed the hyperparameter tuning automatically and it gave us a fit of log loss = 0.8236111111111111 CA = 0.4758512917899232



We got a lower log loss and higher CA from computing manually although the difference in values in little. In addition, I recommend hyperparameter tuning be computed both ways and compare the results.



The diagram above displays the exploration of overfitting neural networks and the diagram below displays the validation loss = 0.0407 accuracy = 0.9822 loss= 0.0393



Training loss >> validation loss so it is suffice to say that our model might have been over feed. On the other hand, the difference between the two losses is 0.0014 which also good enough to say our model is not overfitting.