**ML Project :**

**Bala Sai Praneeth Vankayala**

**AM.EN.U4CSE17508**

**Problem Statement:**

To design an efficient machine learning algorithm that will be able to identify and classify the genre of the music.

The model is said to learn and differentiate the type of songs based on their genre.

● Task(T): Classify the songs based on its genre

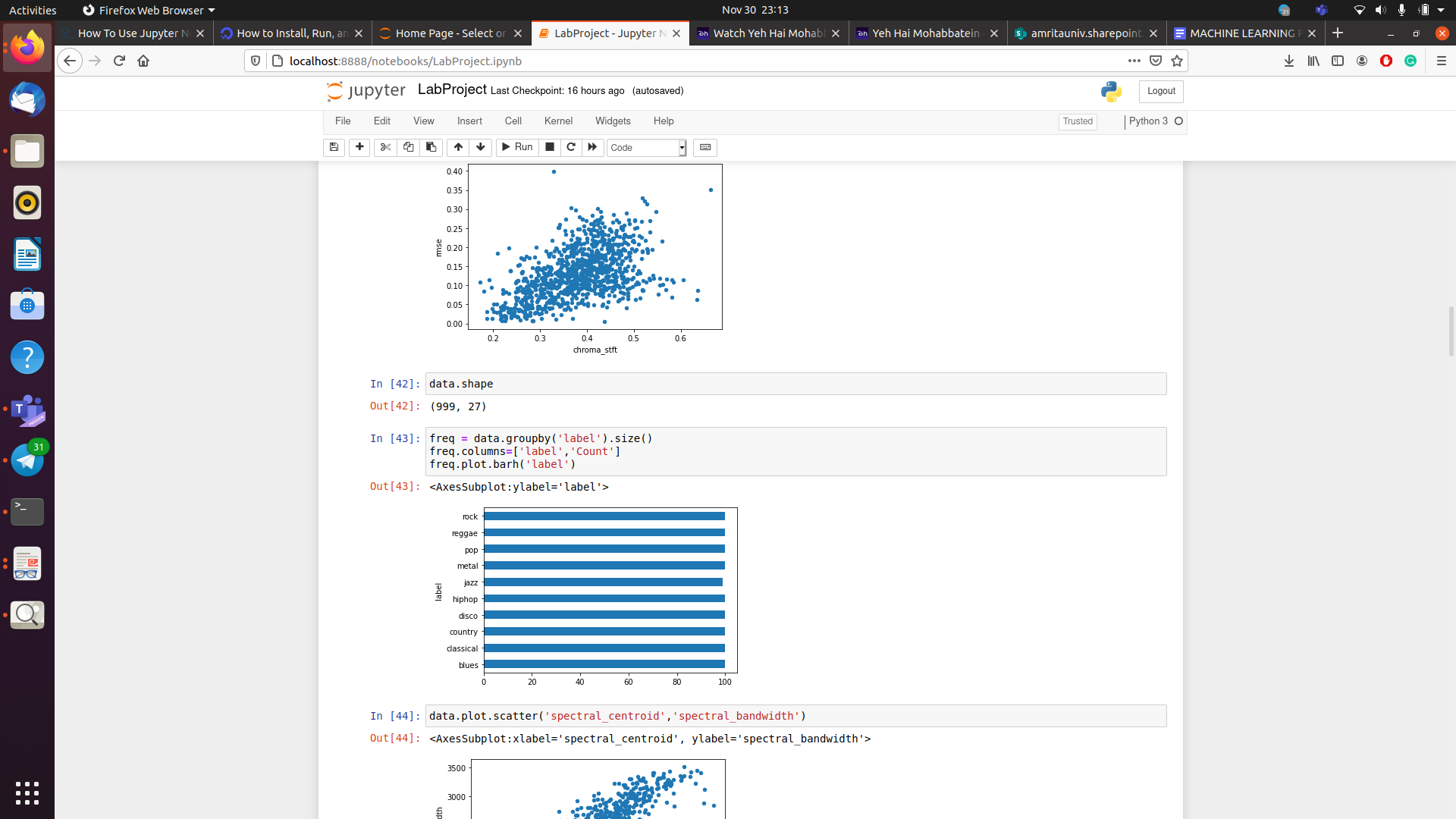
● Experience(E): A data set of songs with its genre

● Performance(P): Classification accuracy, the number of songs predicted correctly out of all songs.

Dataset: https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-ge

nre-classification

Data is distributed equally in the dataset. This dataset contains different genres like jass, pop.



**Data Preprocessing:**

So we are given the data in wav format and I used the librosa package in python to convert the data given in wav format to CSV.

**librosa** is a **python** package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems

I had loaded the data which is in wav format into librosa. load which returns us the time series and the default rate is 22050, then this time series is used to extract multiple features like

**chroma short-time Fourier transform:** It is a variant of Fourier transform which splits the audio signal into frames and then takes the Fourier transform of each frame.

**root-mean-square** (RMS) energy for each frame, using the time series or audio samples generated above.

**Spectral - centroid:** Each frame of a magnitude spectrogram is normalized and treated as a distribution over frequency bins, from which the mean (centroid) is extracted per frame or indicates at which frequency the energy of a spectrum is centered upon.

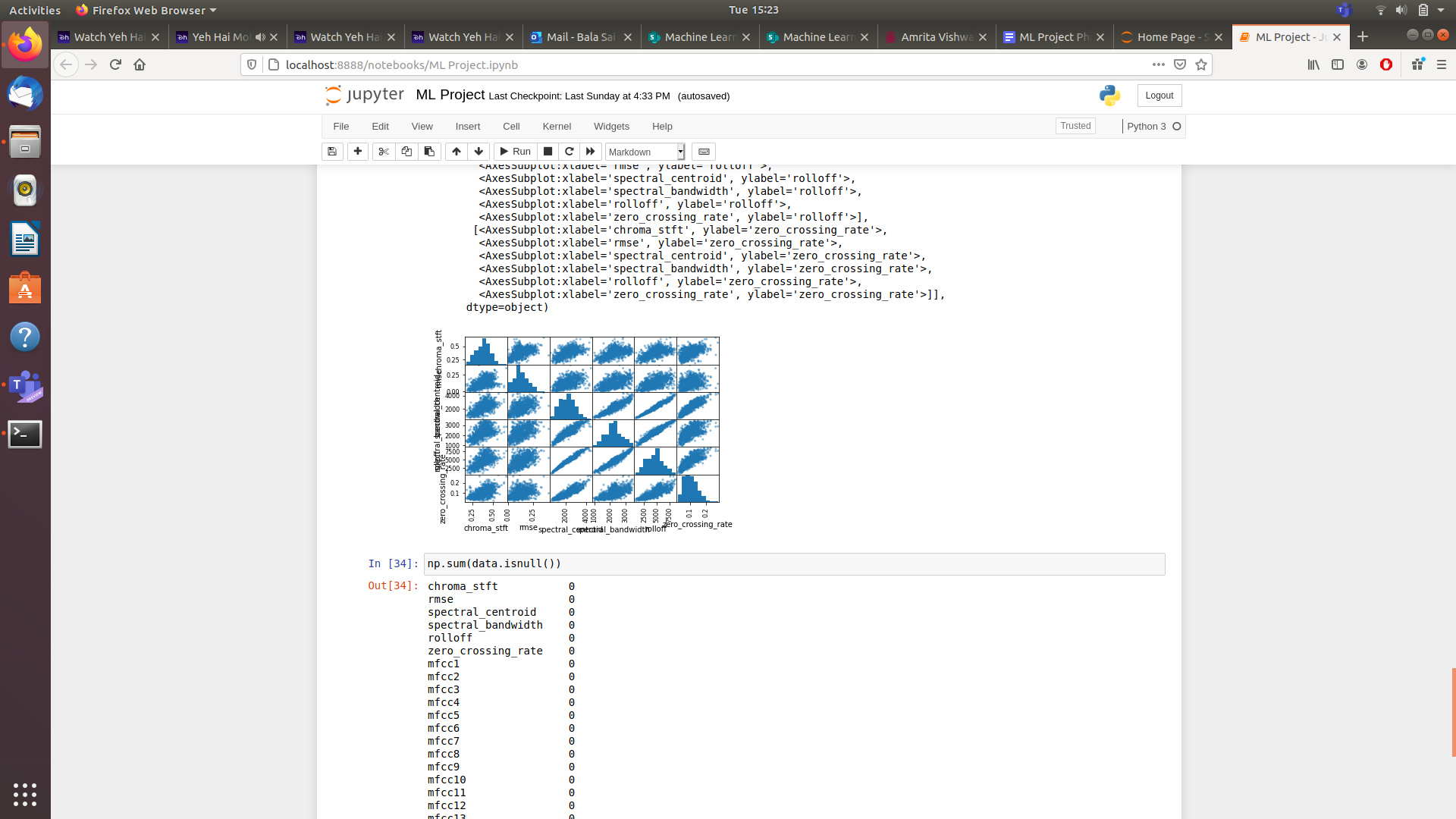
**Spectral roll-off** is the frequency below which a specified percentage of the total spectral energy, e.g. 85%, lies.

We also used the zero-crossing rate and mfcc in our feature set.

**Data Visualization and summarization:**

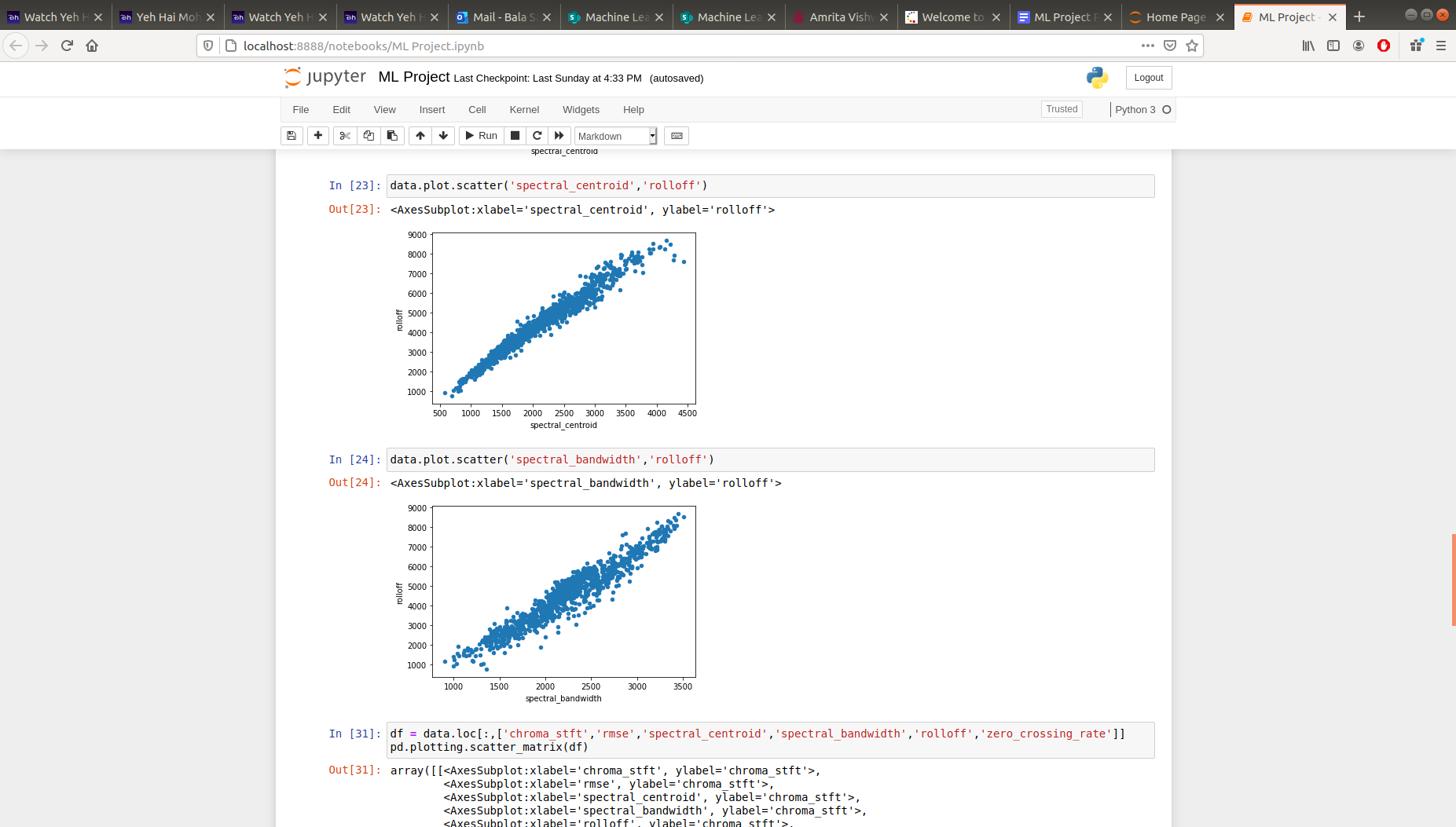
In this, we visualize the data and find the relation between the attributes present in the data.

We use scatter plot and scatter matrix to find the correlation between necessary attributes and then we can remove some attributes which are similarly related.



We can also use heat maps to find the correlation between the attributes.

I have also used standardization and also searched for the null values if any and then fill them using the mean and mode values.



**Data Interpretation:**

* So from the above graphs, I came to the conclusion that there is a clear positive correlation between Roloff and spectral centroid so we can eliminate the roll-of feature from our dataset.
* Converted the music(wav format) to a CSV format dataset.
* Checked for null values and used standardization to make our dataset uniform.

**Python Packages:**

Numpy: *NumPy* is a *python* library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. This is used for numpy arrays and functions.

Pandas: **Pandas** is a high-level data manipulation tool. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables. This is used for storing data in the data frame and manipulating data for data visualization and preprocessing.

Matplotlib: **Matplotlib** is a plotting library for the **Python** programming language and its numerical mathematics extension NumPy. This is used for plotting graphs.

Librosa: L**ibrosa** is a **python** package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. This is used for getting information from audio files.

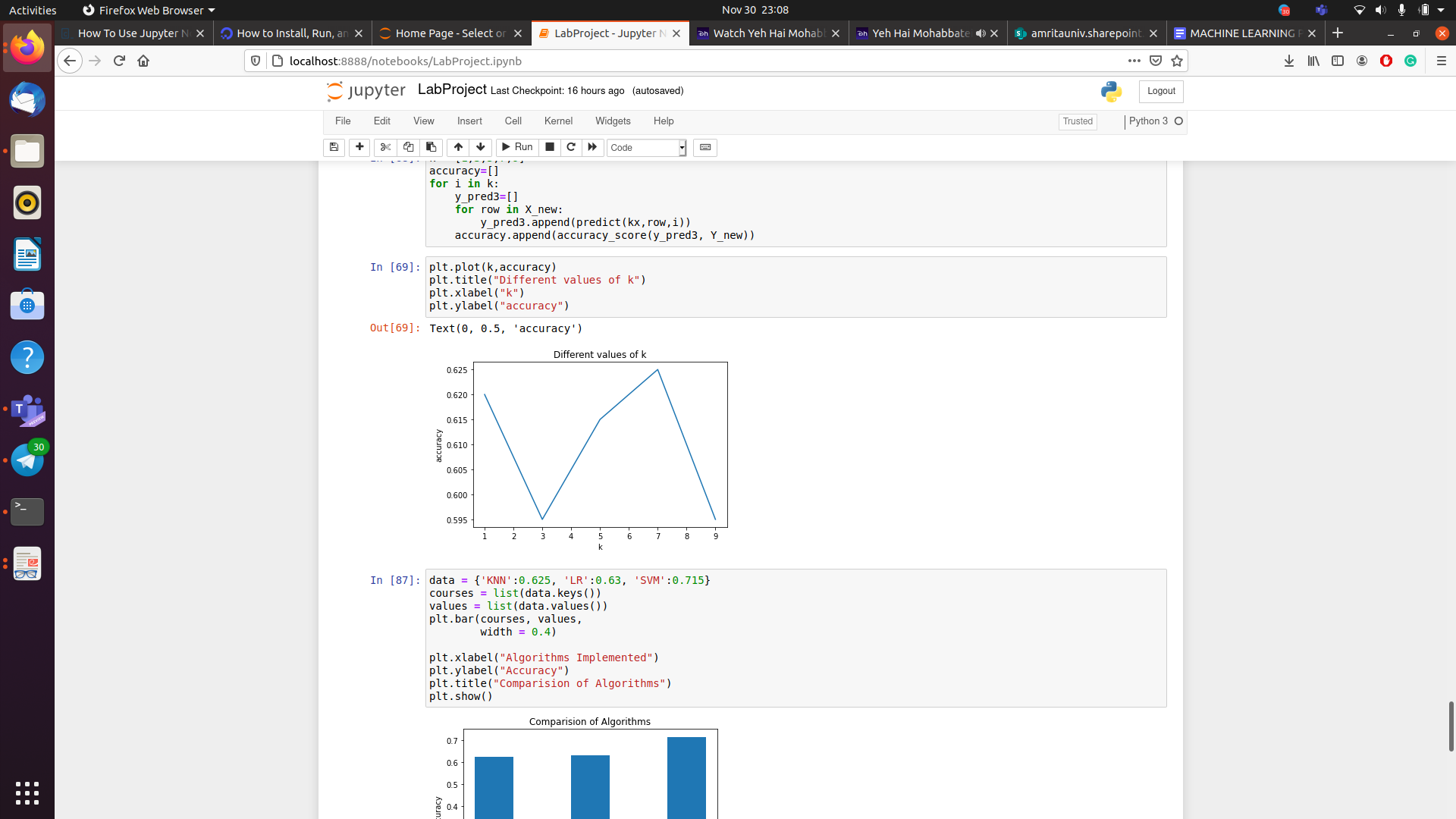
Sklearn: **Scikit-learn** is a free machine learning library for **Python**. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports **Python** numerical and scientific libraries like NumPy and SciPy. This is used to implement machine learning algorithms.

**Supervised Learning Algorithms:**

As our problem is a neural-network-based idea and we can also use some supervised learning algorithms but the accuracy given by these algorithms will be comparatively less than neural network algorithms. But I have done my best to try and match the accuracy of the neural network algorithms.

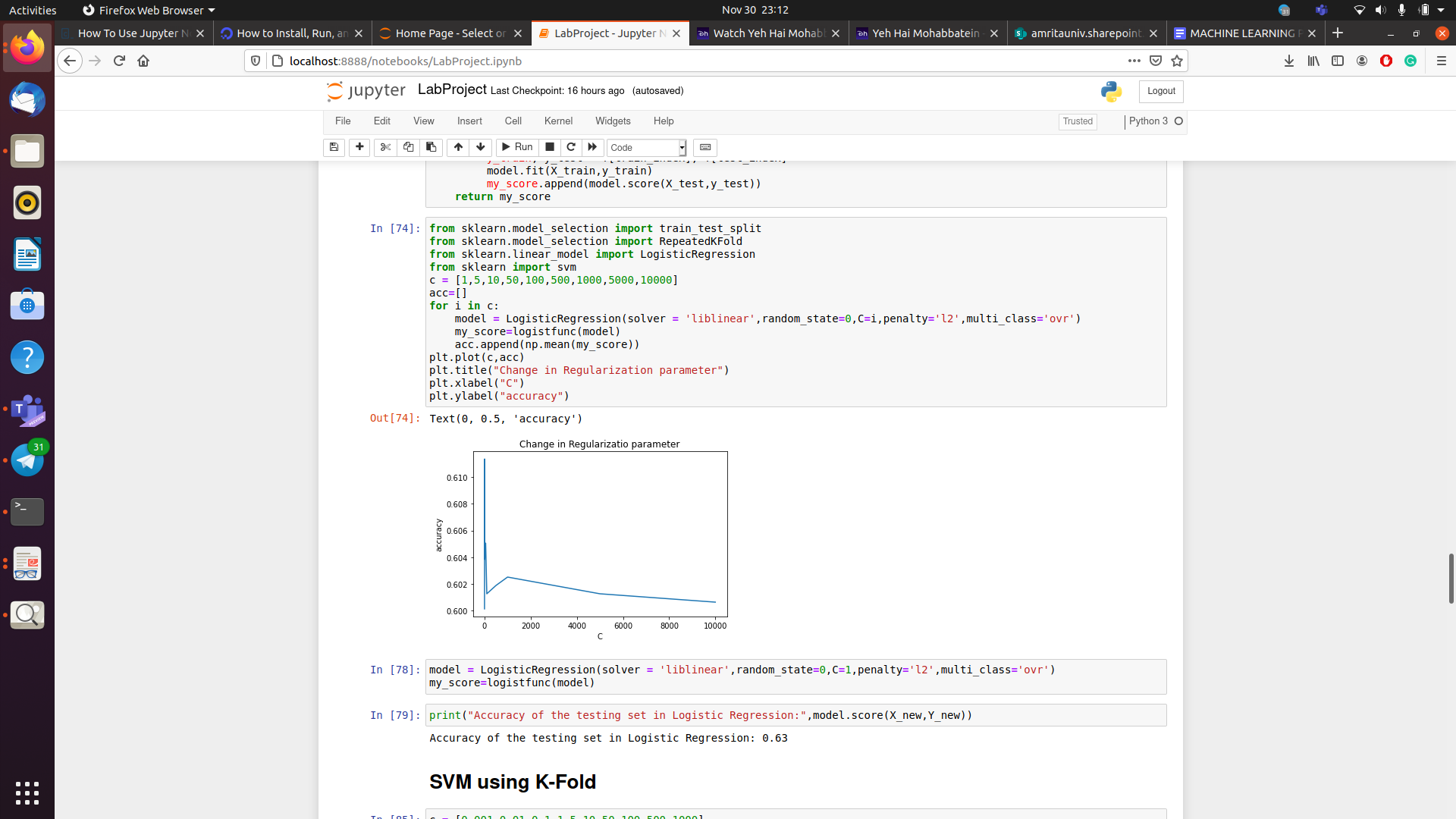
KNN: K-Nearest Neighbors, or *KNN* for short, is one of the simplest machine learning algorithms and is used in a wide array of institutions. *KNN* is a non-parametric, lazy learning algorithm.

The change in accuracy when varying K will be shown in the graph below



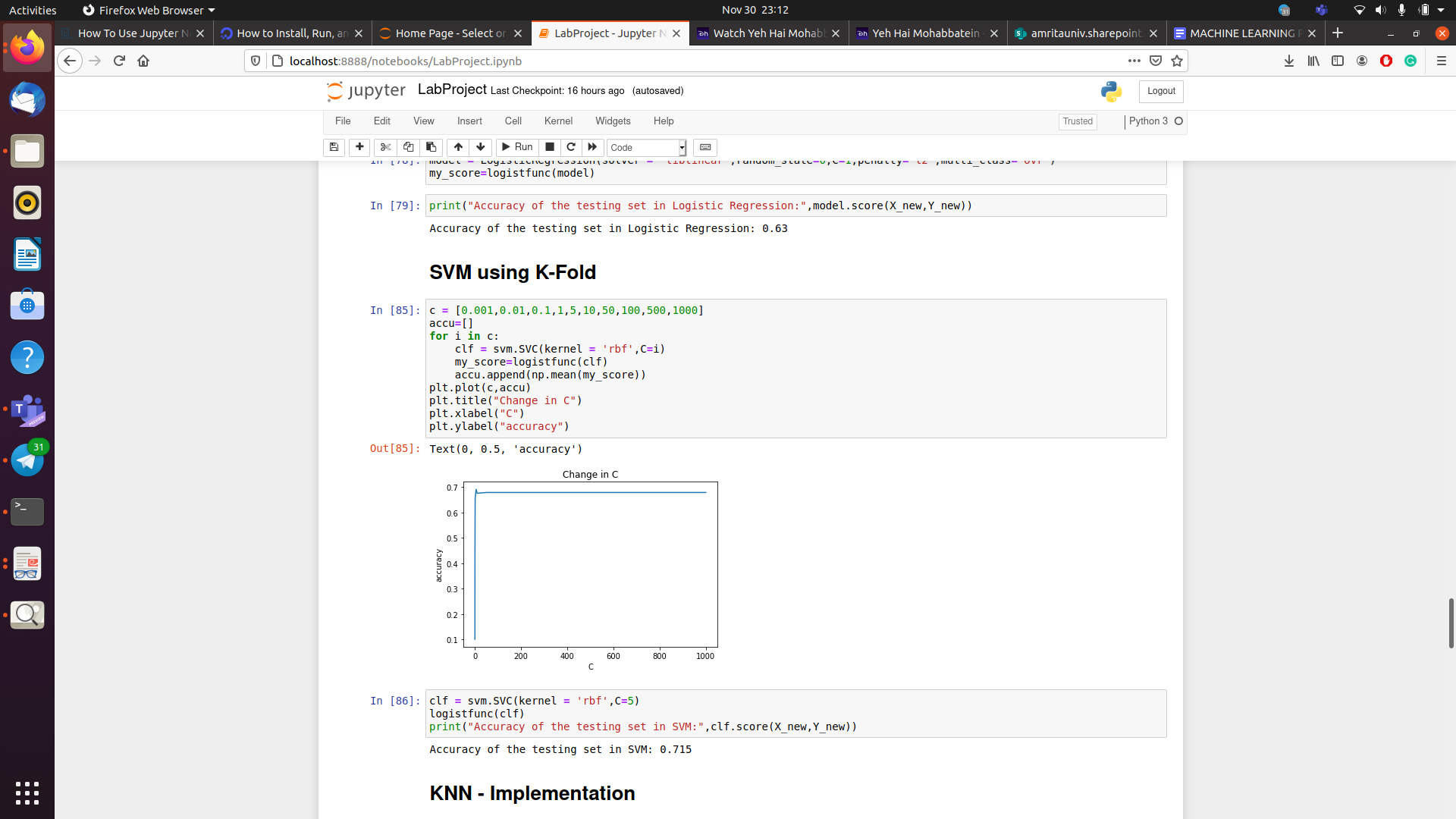
Logistic Regression: This is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables

There is a change in accuracy while varying the regularization parameter.

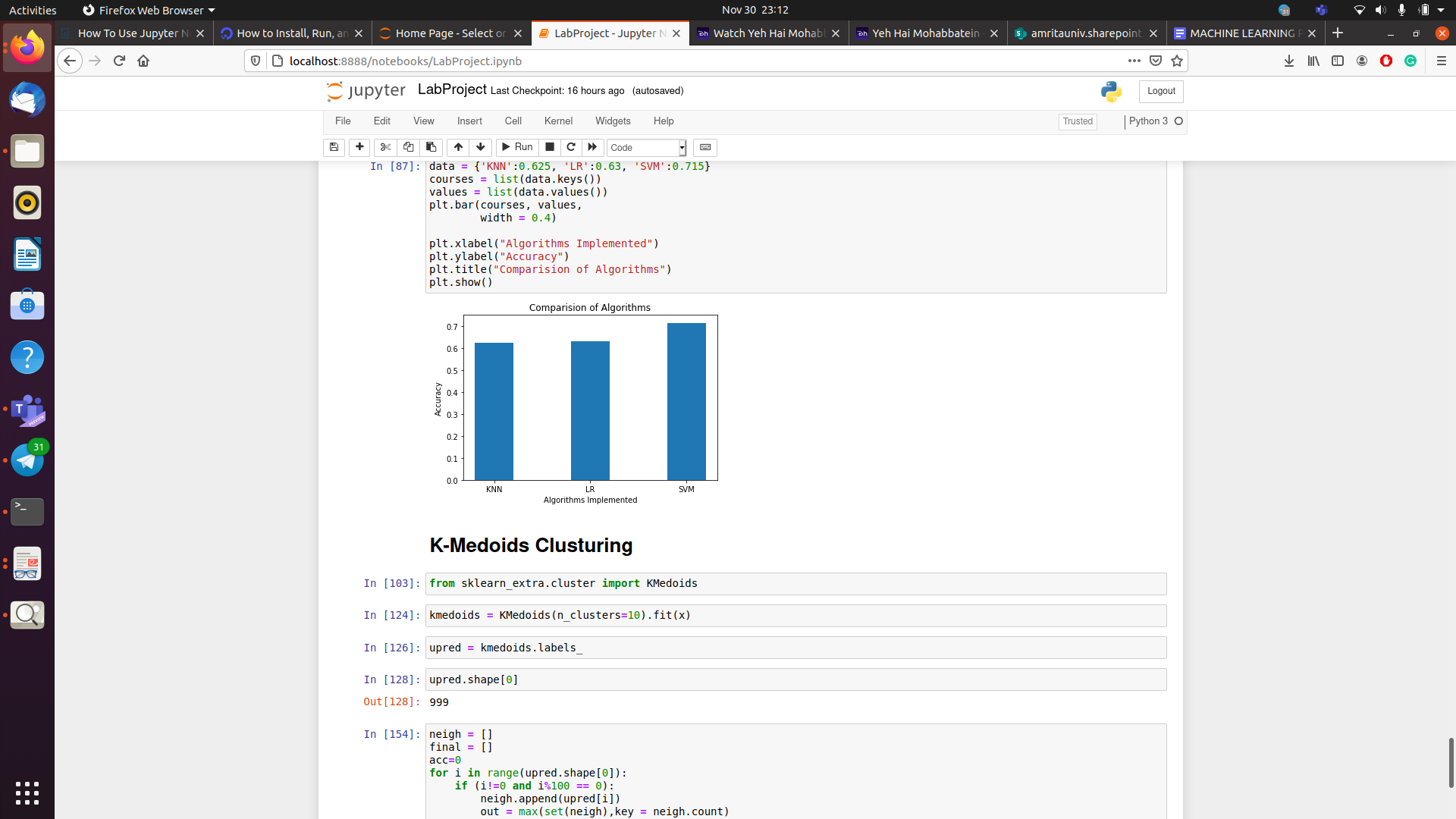


SVM: A support vector machine (**SVM**) is a supervised machine learning model that uses classification algorithms for n group classification problems.

The change in accuracy while varying the parameter C is given below



Comparison between algorithms:



On comparing the algorithms the conclusion is that performance of SVM is good on both validation and testing data.

**Unsupervised Learning:**

K-Mediods clustering: the **k-medoids** or **partitioning around medoids** (**PAM**) algorithm is a clustering algorithm reminiscent of the k-means algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster.

Tried to implement this algorithm but the accuracy got from this algorithm is 0.23. So from this, I got to the conclusion that since our algorithm is based on supervised learning it may not give better outcomes when it is treated with unsupervised learning.