**ANALYSIS OF PERFORMANCE VARIATION OF IOT ATTACK DETECTION USING DATASET SIZE**

A Project Report

Submitted for the partial fulfillment for the award of the degree of

### B Tech (Information Technology)

Submitted by

ANSHU SAINI 2016711

BHARTI 2016731

ESHA AGGARWAL 2016744

Under the supervision of

Submitted To

Dr. Aditi Paul Dr. Deepak Kumar

(Assistant Professor CS) (Assistant Professor CS)



**Department of Mathematics & Computing**

**Banasthali Vidyapith**

**Banasthali - 304022**

**Session: 2022-23**

**Certificate**

Certified that Name of student has carried out the project work titled “ANALYSIS OF PERFORMANCE VARIATION OF IOT ATTACK DETECTION USING DATASET SIZE” from January 2023 to April 2023 for the award of the B Tech (IT) from Banasthali Vidyapeeth under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Dr. Aditi Paul

Assistant Professor

Banasthali Vidyapeeth

Date:

**ABSTRACT**

This analysis examines the impact of dataset size on the performance variation of IoT attack detection using machine learning techniques. We are using different categories of attack and non attack datasets . We are analyzing the effect of dataset size on accuracy of ml models, where the ratio of the dataset refers to how imbalance our dataset is i.e for the 1:1 ratio our dataset is balanced, for 1:2 ratio there is a slight imbalance in our dataset and for 1:3 our dataset is highly imbalance which means that the values are highly spread out. Six different classification algorithms were used to train and test the models on each dataset: Decision Tree, Random Forest, Naïve Bayes, Gradient Boosting, K-Nearest Neighbour, Support Vector Machine. The evaluation metrics used were accuracy, precision, recall score , F1-Score. we can clearly analyze the change in accuracy with respect to the dataset sizes. The dataset size is considered a critical property in determining the performance of a machine learning model. The Random Forest algorithm performed the best overall achieving the highest scores in all evaluation metrics.

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Dr. Aditi Paul for your invaluable support and guidance during the preparation of the. Your constant encouragement, feedback, and expertise have been instrumental in the successful completion of this project. Your valuable insights, constructive criticism, and meticulous attention to detail have helped to shape the project in a meaningful way. Your willingness to share your knowledge and expertise has been a great source of inspiration for us, and we have learned a lot from you during this process.

We appreciate the time and effort you have invested in reviewing and providing feedback on my project report. Your contributions have not gone unnoticed, and We are grateful for the support you have given us throughout this process. We are blessed to have had you as my mentor, and We look forward to applying the skills and knowledge we have gained under your guidance in our future endeavors.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Sr no. | Topic | Page No |
| 1 | Introduction | **6** |
| 2 | Literature Review | **7-10** |
| 3 | Methodology | **11** |
| 4 | Implementation | **12-21** |
| 5 | Results | **22-26** |
| 6 | Conclusion | **27** |
| 7 | References | **29** |

**INTRODUCTION**

Machine Learning can be described as an intelligent device’s ability to modify or automate a knowledge-based state or behavior, which is considered a critical part of an IoT solution. ML has the ability to infer helpful knowledge from data generated by devices or humans, and ML algorithms are used in tasks such as regression, and classification. With the increase of IoT devices connected through the internet, the capacity of web traffic increased. Due to this change, attack detection through common methods and old data processing techniques is now obsolete. Detection and Analysis of attacks in IoT  in the early stages is a very challenging problem due to the increase in the size of network traffic. In this paper, a framework is recommended for the analysis of malicious network traffic.

We are analyzing the effect of dataset size on accuracy of ml models, where the ratio of the dataset refers to how imbalance our dataset is i.e for the 1:1 ratio our dataset is balanced, for 1:2 ratio there is a slight imbalance in our dataset and for 1:3 our dataset is highly imbalance which means that the values are highly spread out. With these differences we can clearly analyze the change in accuracy with respect to the dataset sizes. The dataset size is considered a critical property in determining the performance of a machine learning model.

Establishing a method to find the trend in small datasets is not only of scientific interest but also of practical importance and requires special care when developing machine learning models.

Small datasets typically contain less details, hence the classification model cannot generalize patterns in training data.

Without adding any new data or knowledge, one can still increase the accuracy of the model by adding an element of randomness to the dataset. The main idea or the objective is to train an ensemble of models, where each one is trained on a subset of the data, and then to aggregate the predictions of each model. Because each model is trained on a randomly selected majority subset of the data  there will obviously be significant overlap in data.

**LITERATURE REVIEW**

**Datasets**

The dataset size is considered a critical property in determining the performance of a machine learning model. Typically, large datasets lead to better classification performance and small datasets may trigger over-fitting. Moreover, in the case of available large datasets, training a model using such data requires further time and computing resources, which may not be available.

Establishing a method to find the trend in small datasets is not only of scientific interest but also of practical importance and requires a special care when developing machine learning models. Unfortunately, classification algorithms may perform worse when trained with limited size datasets. This is because small datasets typically contain less details, hence the classification model cannot generalize patterns in training data. In addition, over-fitting becomes much harder to avoid as it sometimes goes beyond training data to affect the validation set as well. Classification is a challenging task by itself. It becomes more challenging when dealing with small datasets. The central cause behind this challenge relates to the limited size of training data, which leads to unreliable and biased classification model [3]. While previous studies are focusing on increasing the accuracy of the classification algorithms on limited size datasets, less effort was made to study the impact of the size property of the dataset on the performance of the classification algorithms, which makes it an open problem in the area that needs more investigation.

Research on the subject has been mostly restricted on increasing the accuracy of the classification algorithms on limited size datasets, little attention has been paid to study the impact of the dataset size on the performance of the classification algorithms.

Small Datasets: The results derived from small datasets reveal that, depending on the problem domain, dataset size is not necessarily an obstacle to a high performing model n a small dataset, classifiers perform relatively similarly, while each classifier has varying performance across the small datasets.

Large Datasets: Statistical tests revealed that DT is the most sensitive model to the size of the dataset since its performance decreases significantly in the majority of the scenarios. RF and NN showed a relatively similar response to the decrease of dataset size as they show significant performance degradation. For K-NN, the model learns by adjusting a large number of weights using back propagation. Thus, more data allows further adjustment, and hence better performance. The next model is SVM, where its performance decreases significantly of the scenarios.

**Classification Methods**

Different ML based widely-used classifiers are there which included probabilistic classification using naïve Bayes (NB), decision function classification using support vector machine (SVM), K-Nearest Neighbour (KNN), decision tree (DT), tree ensemble random forest (RF), Below, we shed light on these classification models:

* **SVM**:

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

The objective of the SVM algorithm is to find the hyperplane in the data that gives the largest separation margin between data instances and classifies them into two classes. It can be explained based on four basic concepts, the separating hyperplane, the maximum margin hyperplane, the soft margin, and finally the kernel function. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

* **NB:**

It is a supervised learning method based on the Bayesian theorem. Therefore, it is considered as a statistical method for classification. It works by calculating explicit probabilities for hypotheses. It is mainly used in *text classification* that includes a high-dimensional training dataset.NB models use the method of maximum likelihood for parameter estimation. Literature showed that it often performs better in many complex real-world applications. One of the features of this method is that it is robust to noise in data, and it can estimate the parameters using a small training set .Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

* + **DT:**

A decision tree which is also known as prediction tree refers a tree structure to mention the sequences of decisions as well as consequences. In a decision tree, all the test points exhibit testing specific input variables (or attributes), and the developed decision tree is represented by the branches. Because of flexibility as well as simple visualization, decision trees are mostly probably deployed in data mining applications for the purpose of classification. In the decision tree, the input values are considered as categorical or continuous. A structure of test points (known as nodes) and branches is established by the decision tree by which the decision being made will be represented. Leaf node is the one which do not have further branches. The returning value of leaf nodes is class labels while in some cases they return the probability scores.

A Decision Tree is constructed as a binary classification tree, based on the training data. In the tree structure, class labels are represented by leaf nodes, while the internal nodes represent the conjunction of features that assess class.

* + **KNN:**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. The algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN. The algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

* + **RF:**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

To achieve good predictions, those models require a certain amount of data to train on, whereas this amount is generally limited and difficult to obtain; and increases with the complexity of the interactions between the outcome and the model variables. In authors compared the ways training dataset size and interactions affect the performance of those prediction models.

* **GB**

Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.

To compare the two influences, several datasets were simulated that differed in the number of observations and the complexity of the interactions between the variables and the outcome. A few logistic regressions and neural networks were trained on the simulated datasets and their performance evaluated by cross-validation and compared using accuracy, F1 score, and AUC metrics. This work aims to investigate the impact of dataset size on the performance of six widely-used supervised machine learning models in the medical domain.

**METHODOLOGY**

**EVALUATION**

**USING DIFFERENT MACHINE LEARNING TECHNIQUES**

**EXTRACTING THE FEATURES FROM THE DATASET**

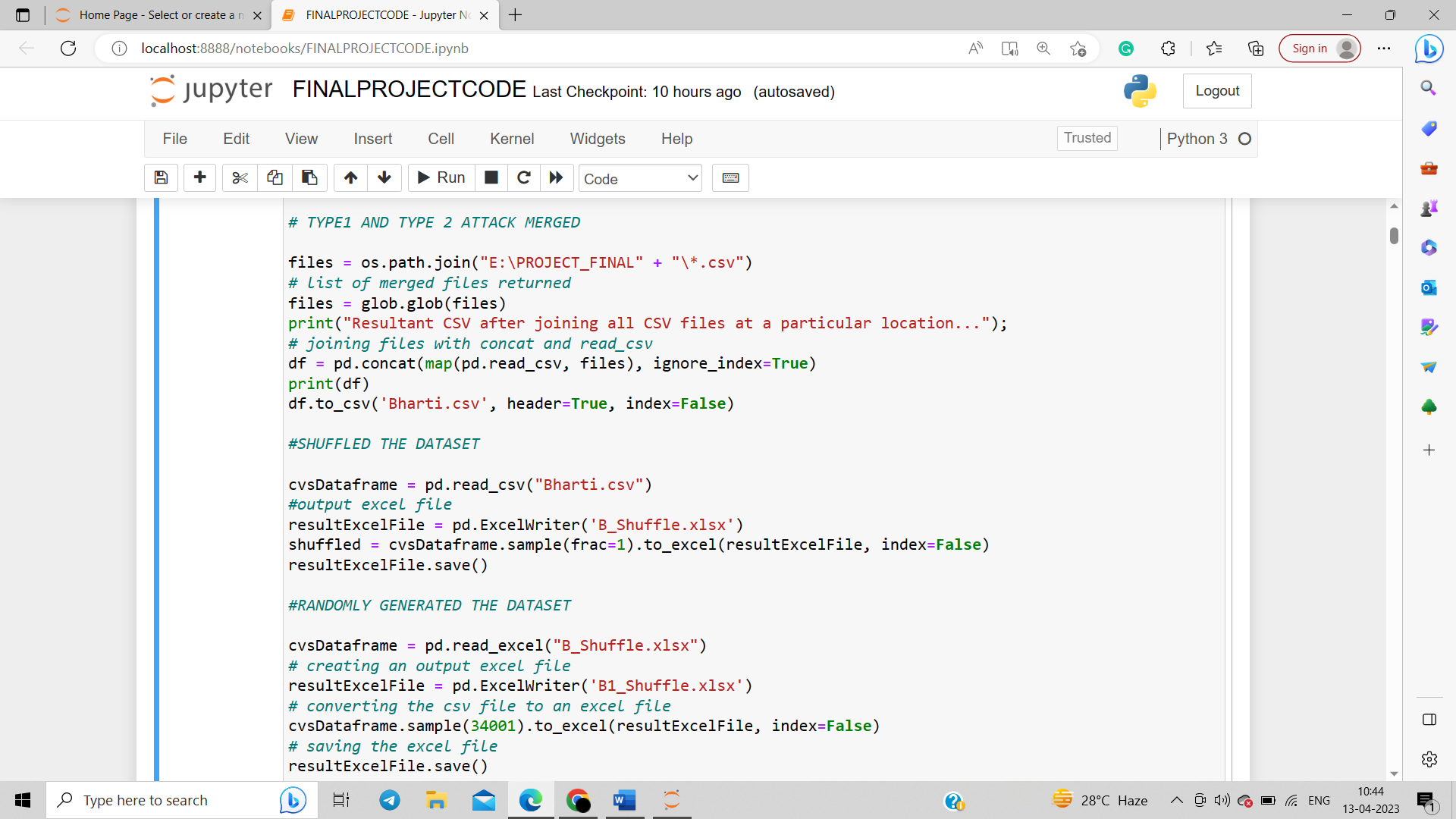
**FEATURE SELECTION**

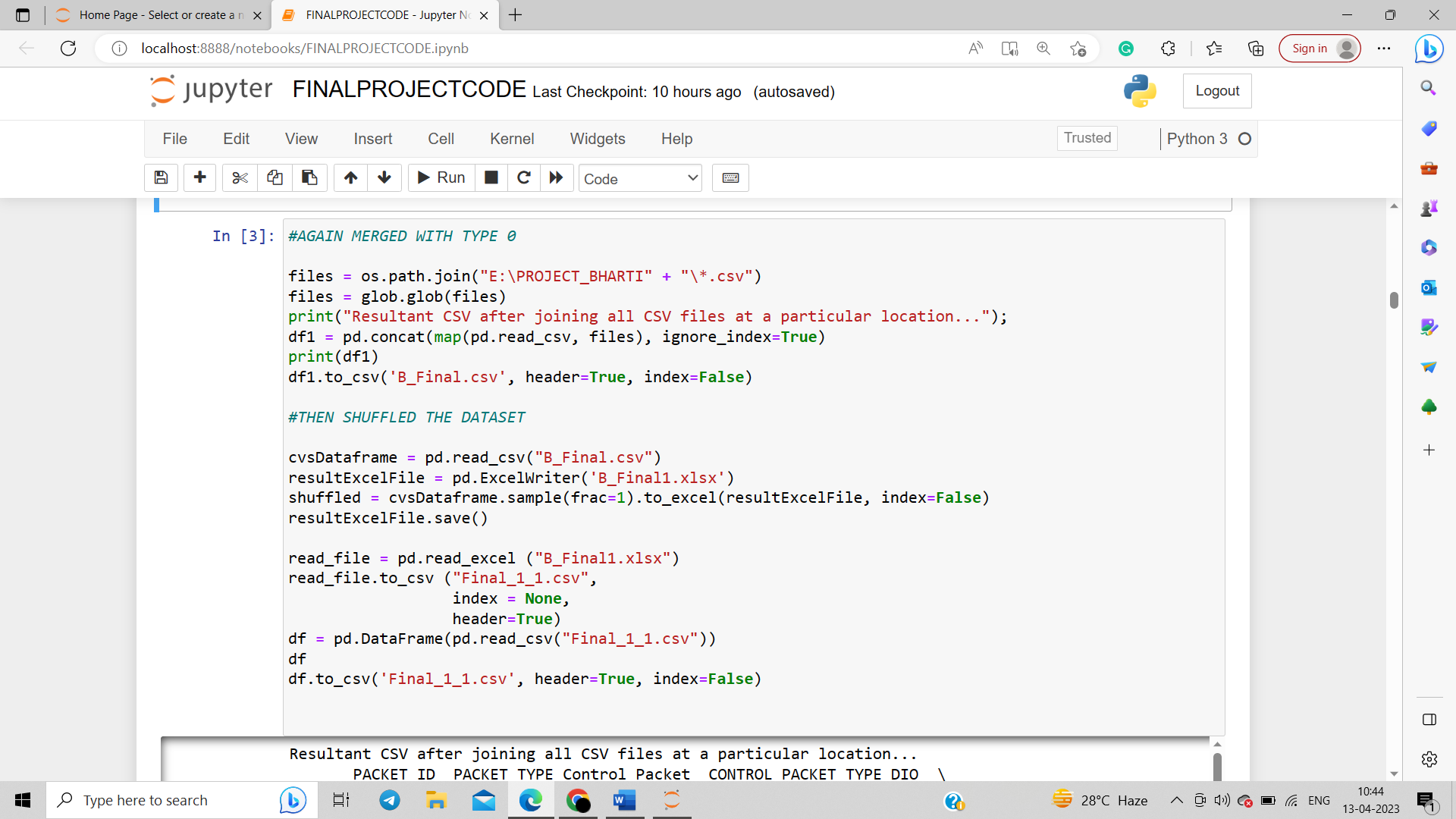
**DATA PREPROCESSING**

**COLLECTION OF RAW DATA**

**For Randomness and Biasness of data**

To get the maximum randomness in data we have merged the Type 1 and Type 2 attack dataset and then shuffled it and then merged it with no attack and then extracted the ratios 1:1, 1:2, 1:3





**DATA PREPROCESSING:**

**Steps Involved in Data Preprocessing:**

**1. Data Cleaning:**   
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.   
**(a). Missing Data:**   
This situation arises when some data is missing in the data. It can be handled in various ways.   
Some of them are:

* **Ignore the tuples:**   
  This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
* **Fill the Missing values:**   
  There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

**(b). Noisy Data:**   
Noisy data is a meaningless data that can’t be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways

* **Binning Method:**   
  This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
* **Regression:**   
  Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables.

**2. Data Transformation:**   
This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

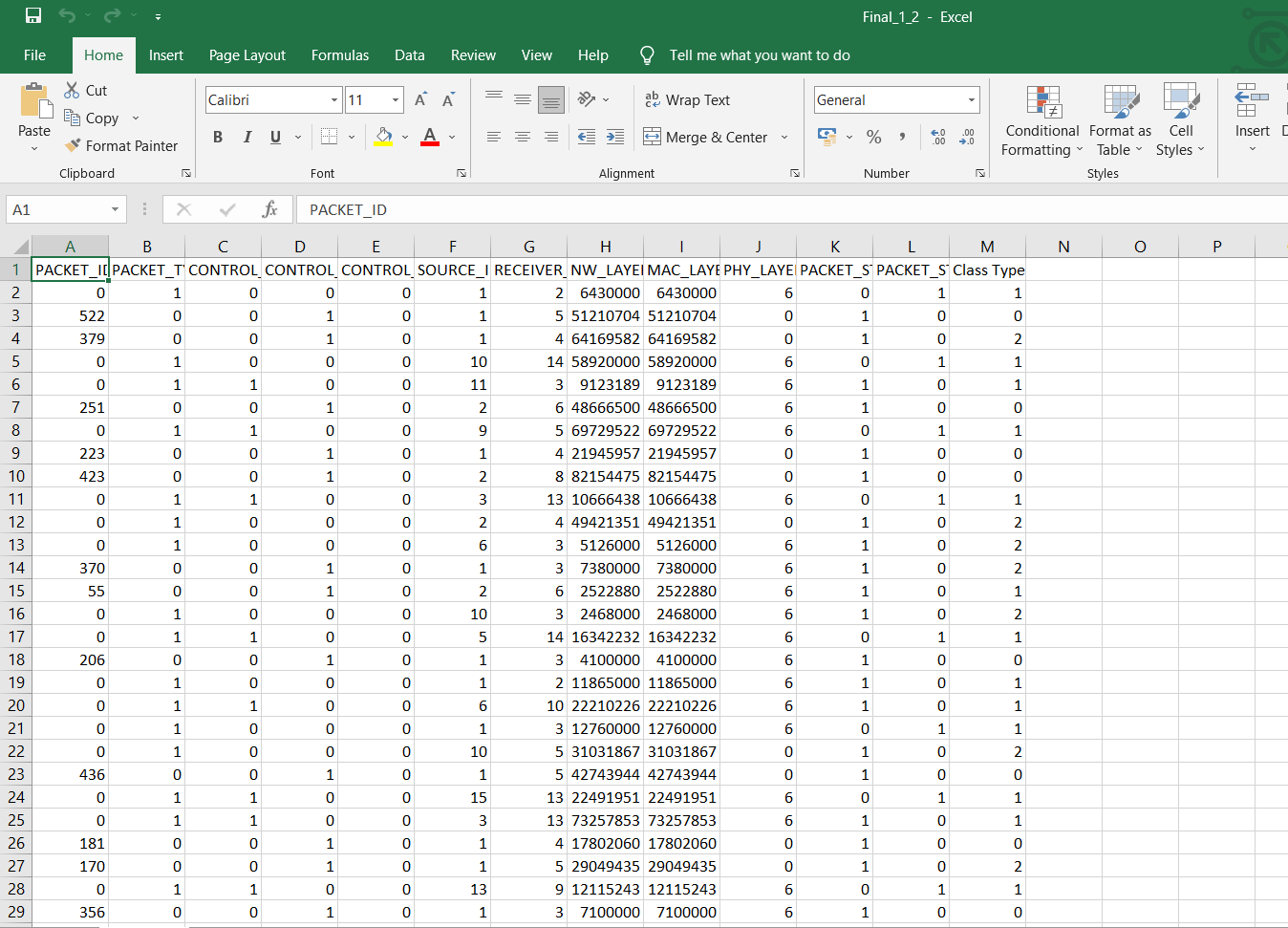
* **Normalization:**   
  It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
* **Attribute Selection:**   
  In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
* **Discretization:**   
  This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

**3. Data Reduction:**   
Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we used data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

* **Data Cube Aggregation:**   
  Aggregation operation is applied to data for the construction of the data cube.
* **Attribute Subset Selection:**   
  The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.
* **Numerosity Reduction:**   
  This enables to store the model of data instead of whole data, for example: Regression Models.
* **Dimensionality Reduction:**   
  This reduces the size of data by encoding mechanisms. It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are: Wavelet transforms and PCA (Principal Component Analysis).

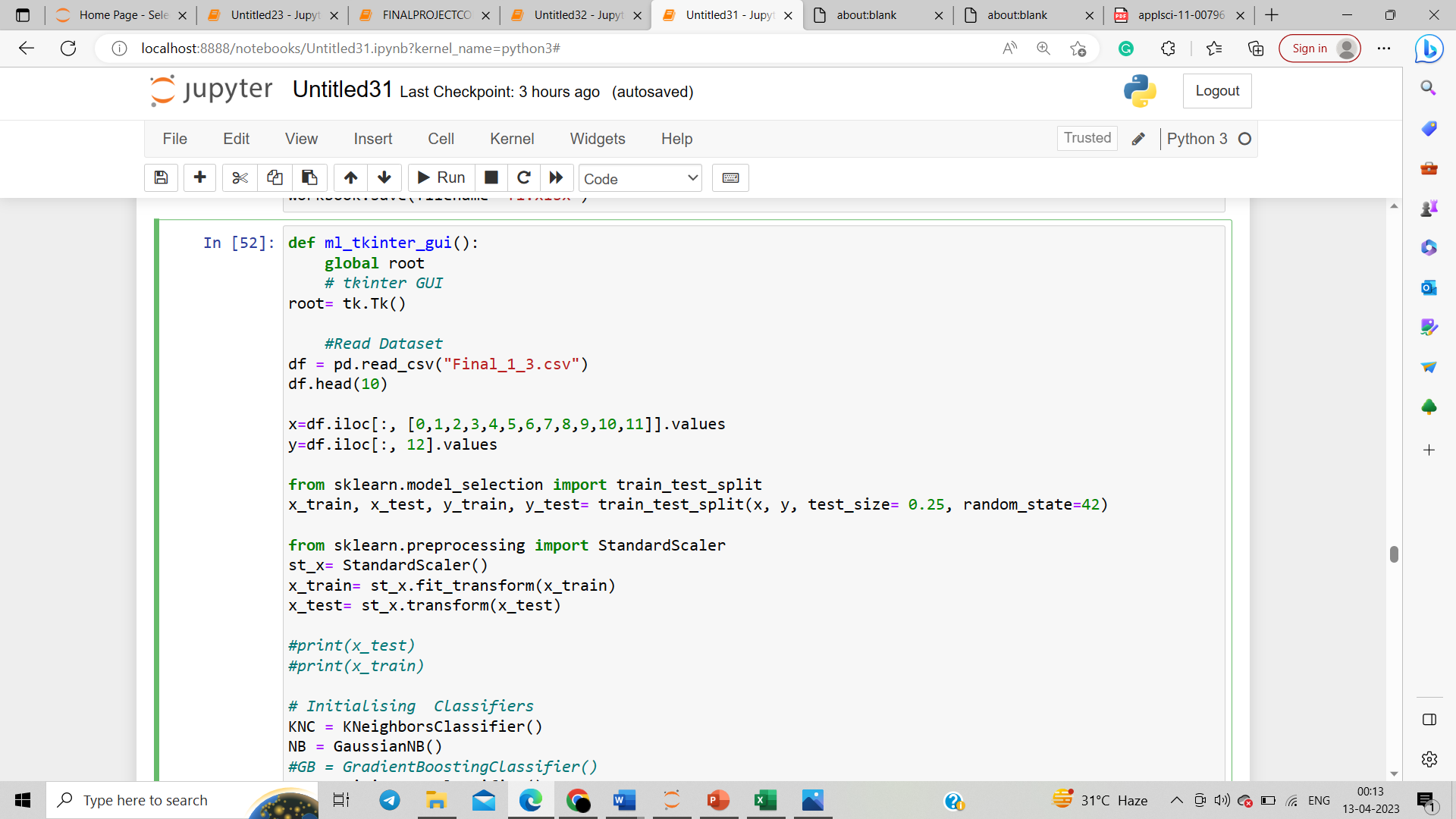
Dataset has following features as shown in the figure:



**Reading the data:**

Reading the data using pandas library

import pandas as pd



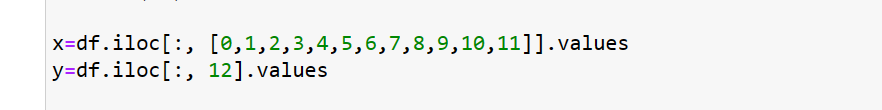
**Feature Selection and Extraction**

**A** feature is an attribute that has an impact on a problem or is useful for the problem, and choosing the important features for the model is known as feature selection. Each machine learning process depends on feature engineering, which mainly contains two processes; which are Feature Selection and Feature Extraction.

It is necessary to provide a pre-processed and good input dataset in order to get better outcomes. We collect a huge amount of data to train our model and help it to learn better. Generally, the dataset consists of noisy data, irrelevant data, and some part of useful data. Moreover, the huge amount of data also slows down the training process of the model, and with noise and irrelevant data, the model may not predict and perform well. So, it is very necessary to remove such noises and less-important data from the dataset and to do this, and Feature selection techniques are used. Selecting the best features helps the model to perform well.

**Some Feature Selection Techniques are:**

* **Filter Method**: In this method, features are dropped based on their relation to the output, or how they are **correlating** to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly. Eg: Information Gain
* **Wrapper Method**: We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features. Eg: Forward Selection
* **Intrinsic Method**: This method combines the qualities of both the Filter and Wrapper method to create the best subset. This method takes care of the machine training iterative process while maintaining the computation cost to be minimum.



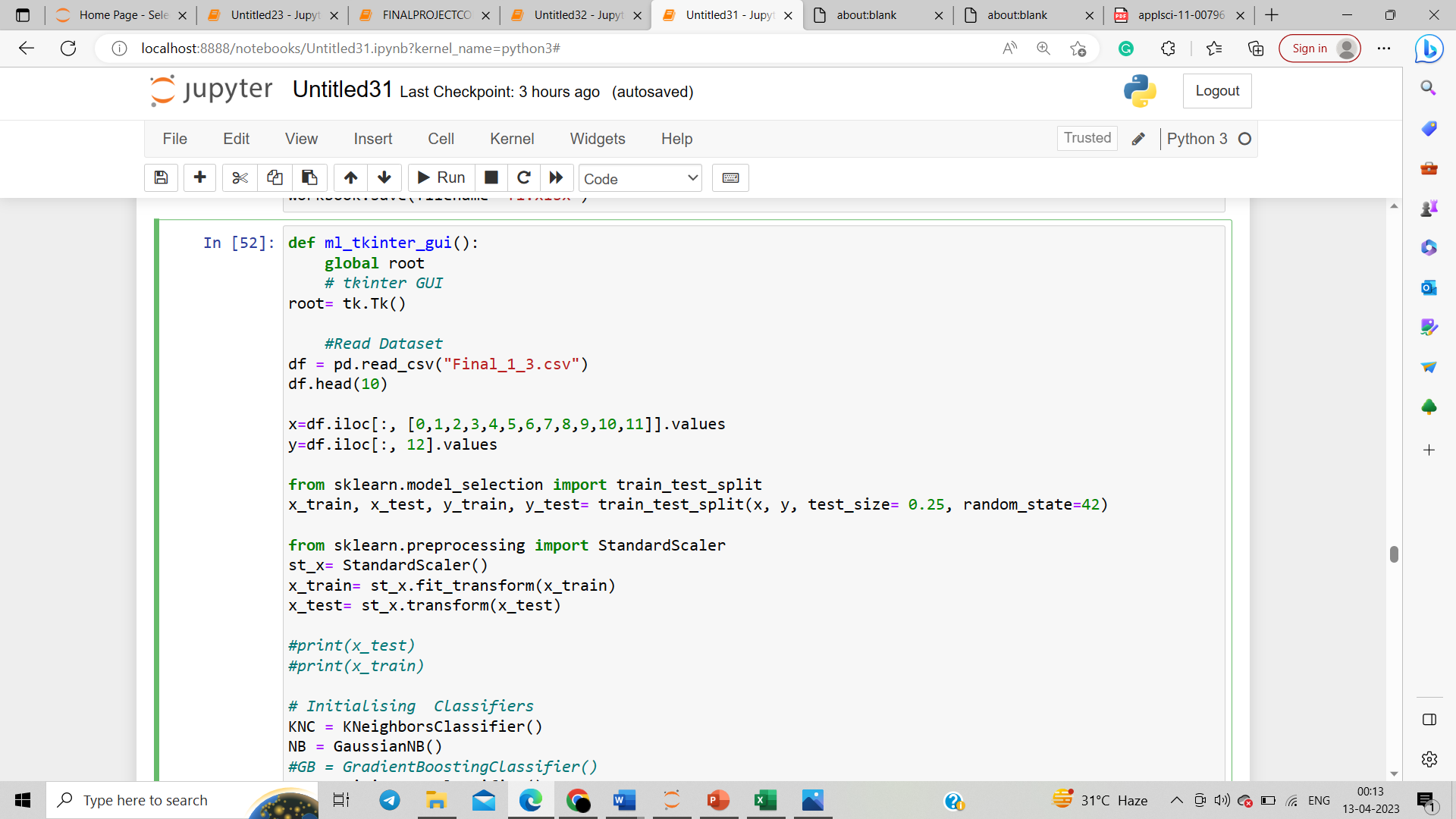
**Splitting the data into training and testing**

Splitting data into training and testing sets is a common practice in machine learning to evaluate the performance of a predictive model. The basic idea is to use a portion of the available data to train the model, and then use another portion of the data to test the model's accuracy on new, unseen data.

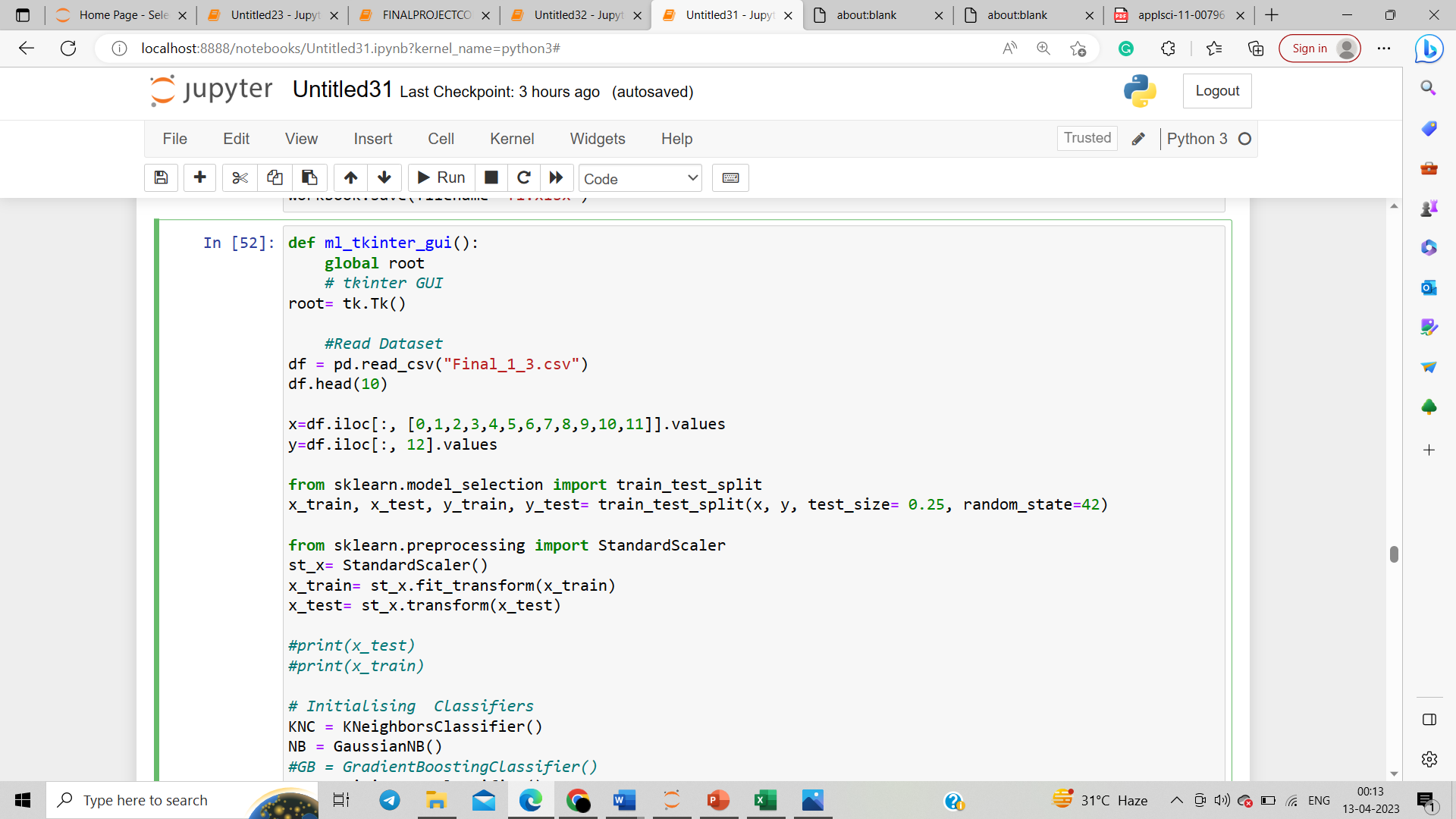
Training the Algorithm: The machine learning algorithm is trained using the training set. During training, the algorithm learns the patterns and relationships in the data that will enable it to make accurate predictions on new, unseen data.

Testing the Algorithm: Once the algorithm has been trained, it is evaluated on the testing set. The algorithm's performance on the testing set is used to estimate how well it will perform on new, unseen data.

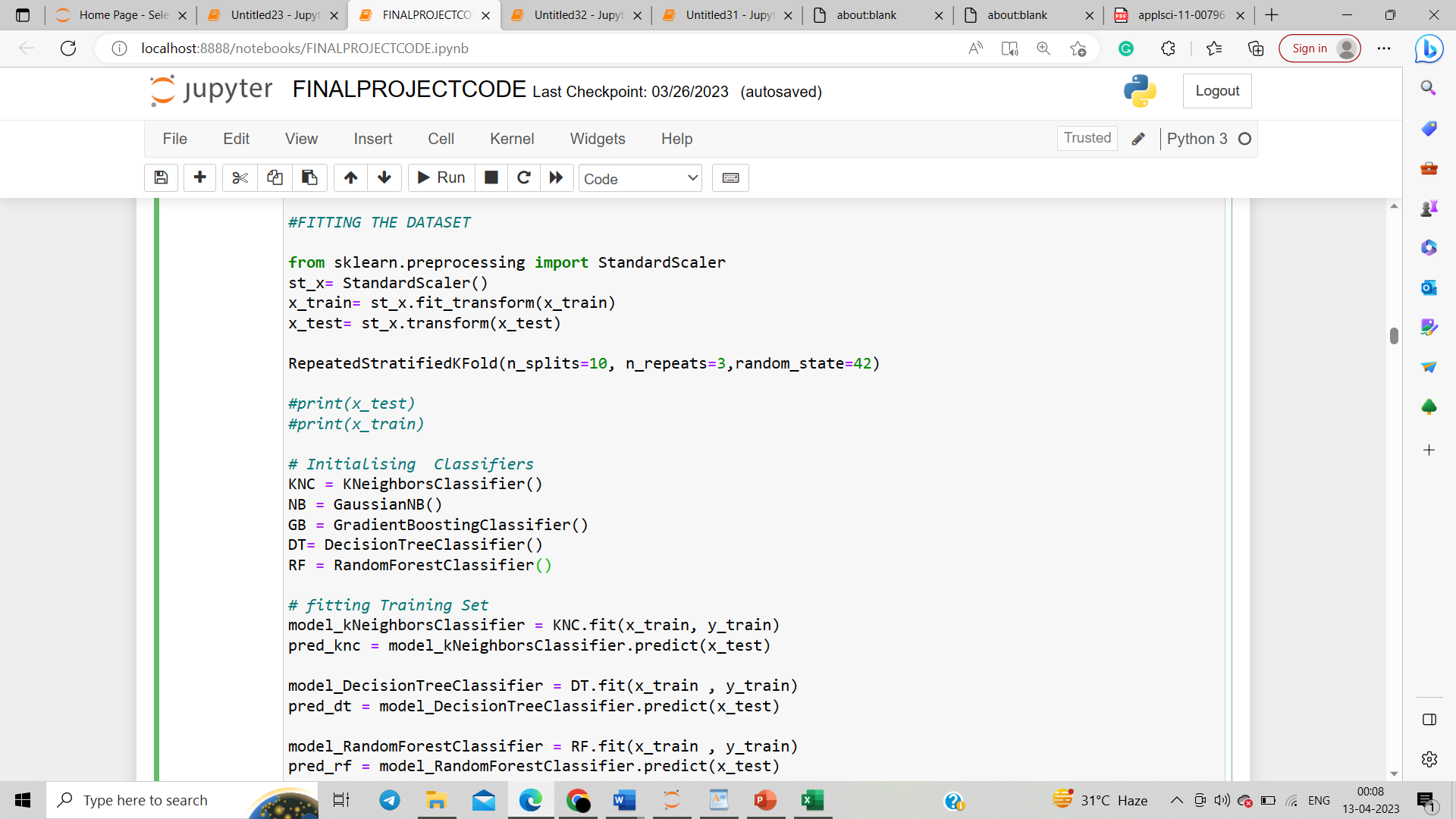
By Importing train\_test\_split, we are spliting the data into training and testing

****

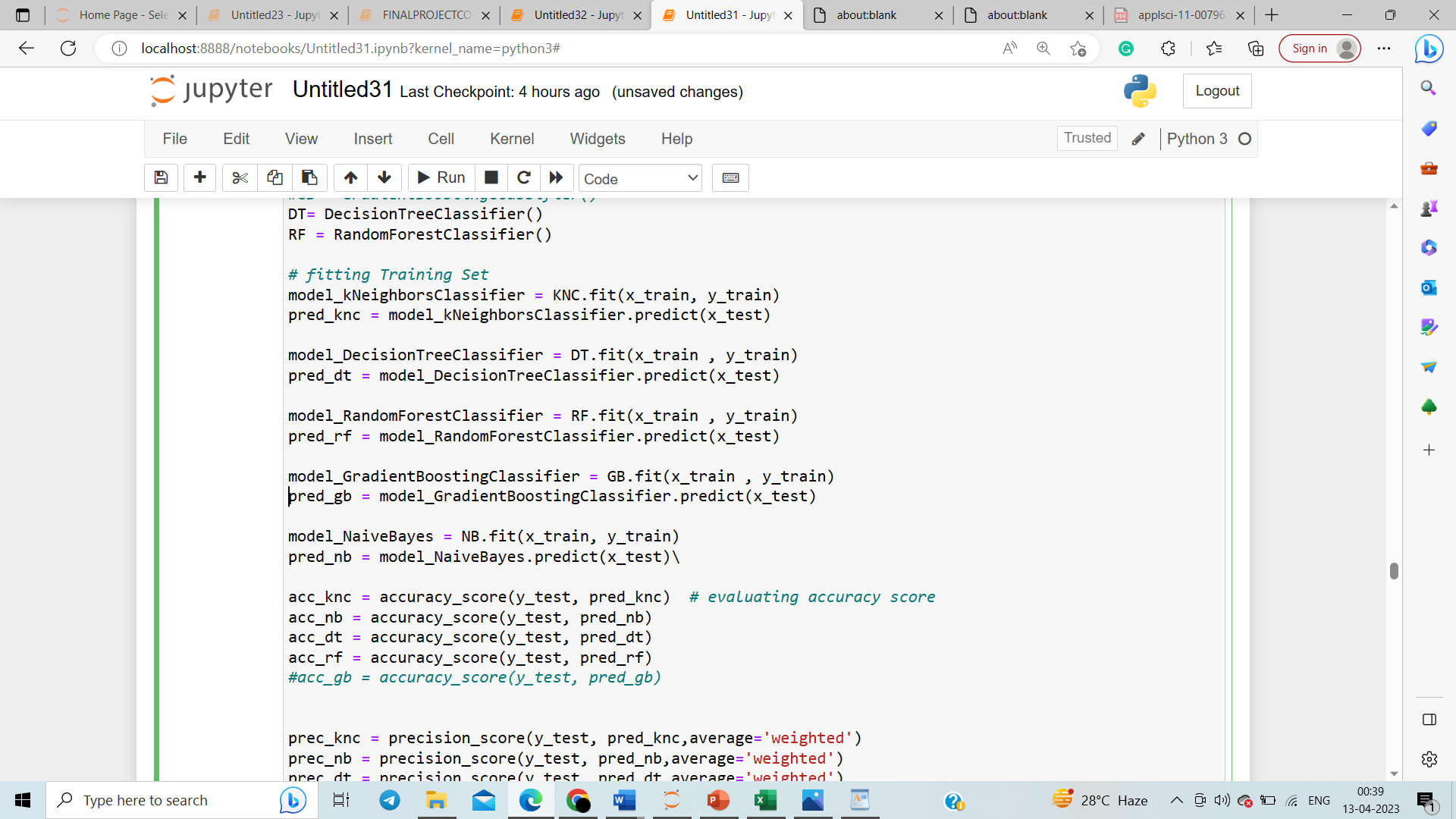
**Using StandardScaler for fitting the dataset**

****

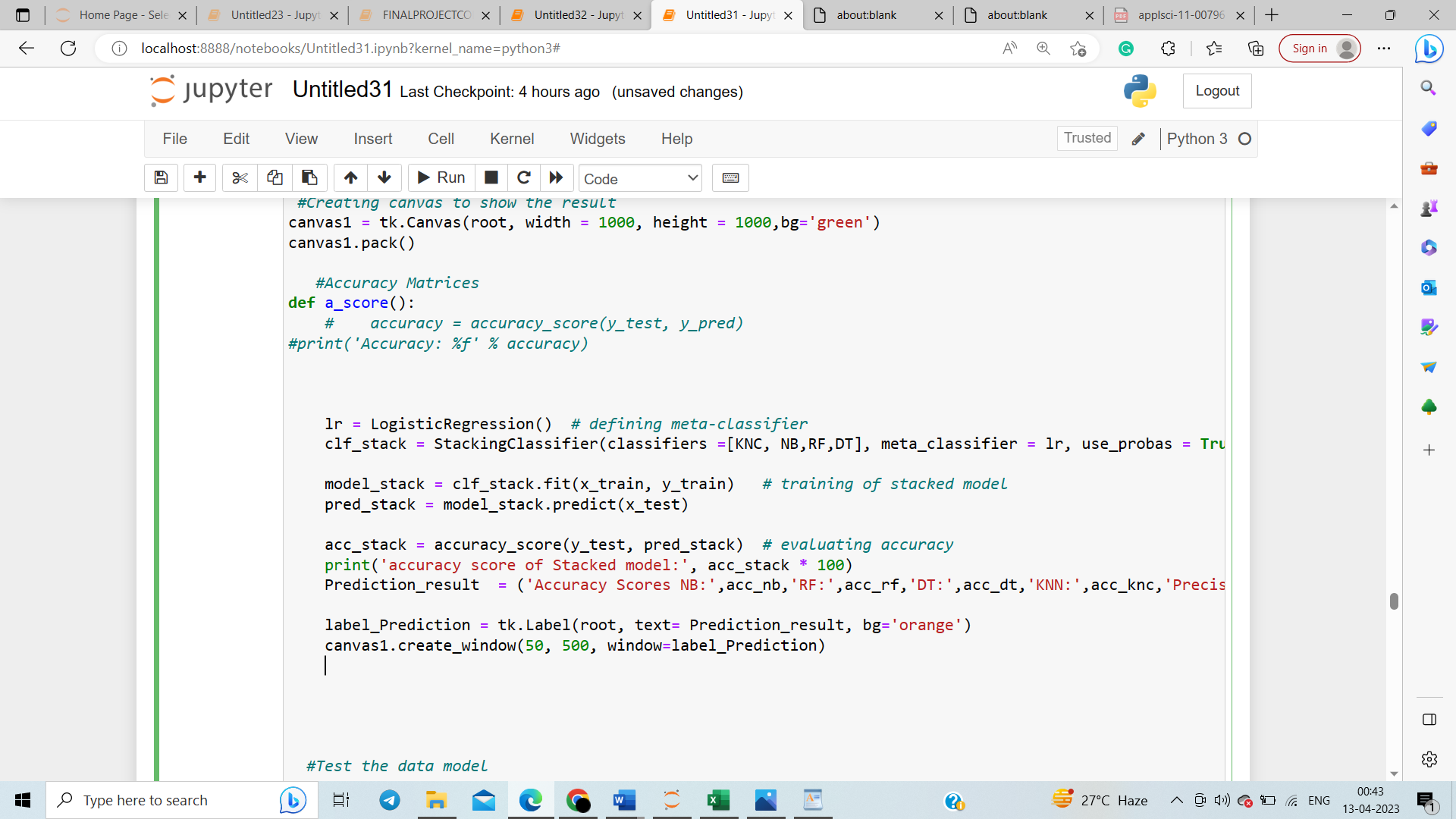
**Initialising Different Machine Learning Classifiers**

****

**Fitting the training set into Classifiers**

****

**Using LR(Logistic Regression) as a base model :**

****

**Calculating the accuracy, precision, recall score and f1 score**

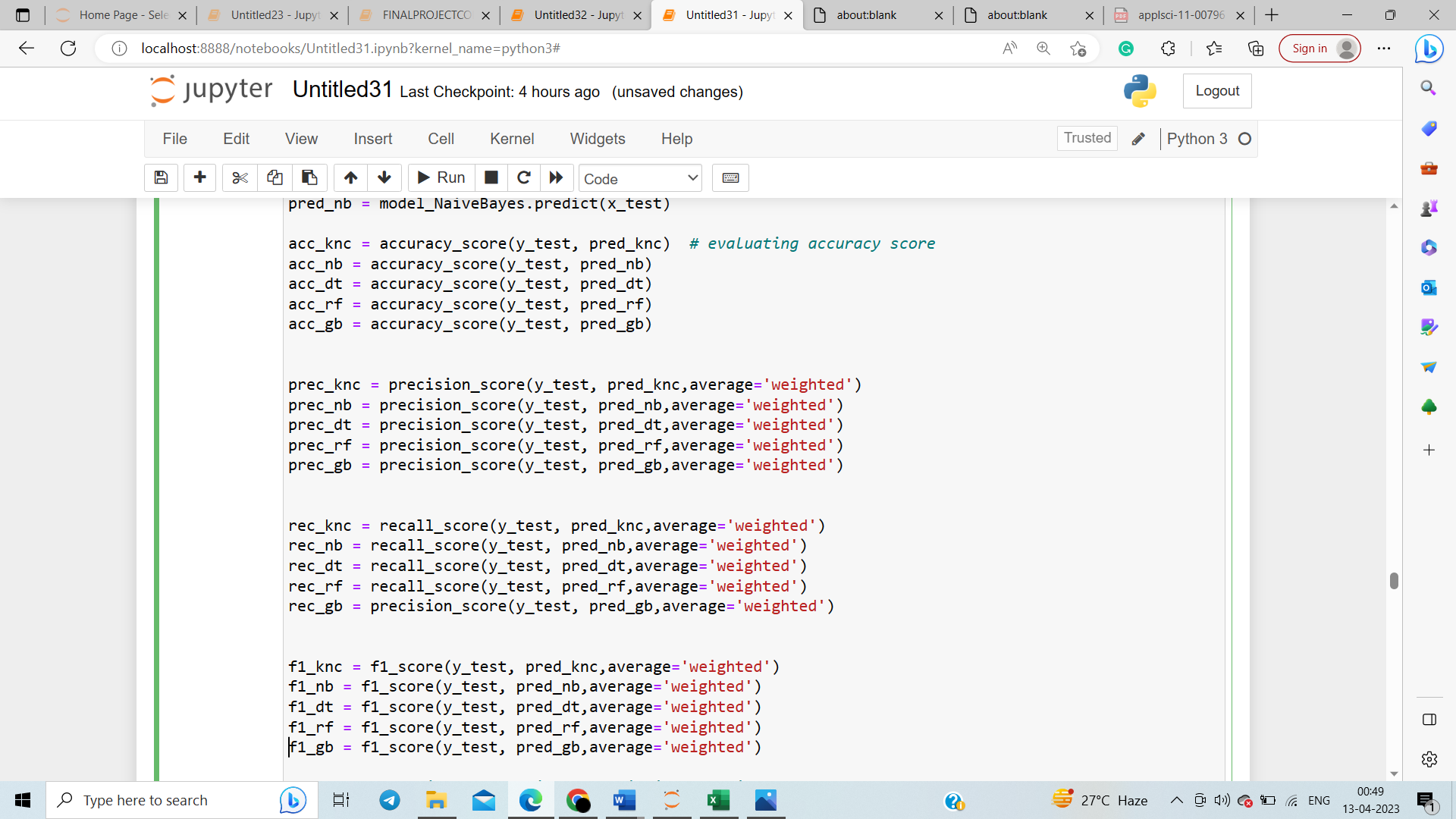
By importing accuracy\_score, f1\_score, recall\_score as follows:

from sklearn.metrics import f1\_score

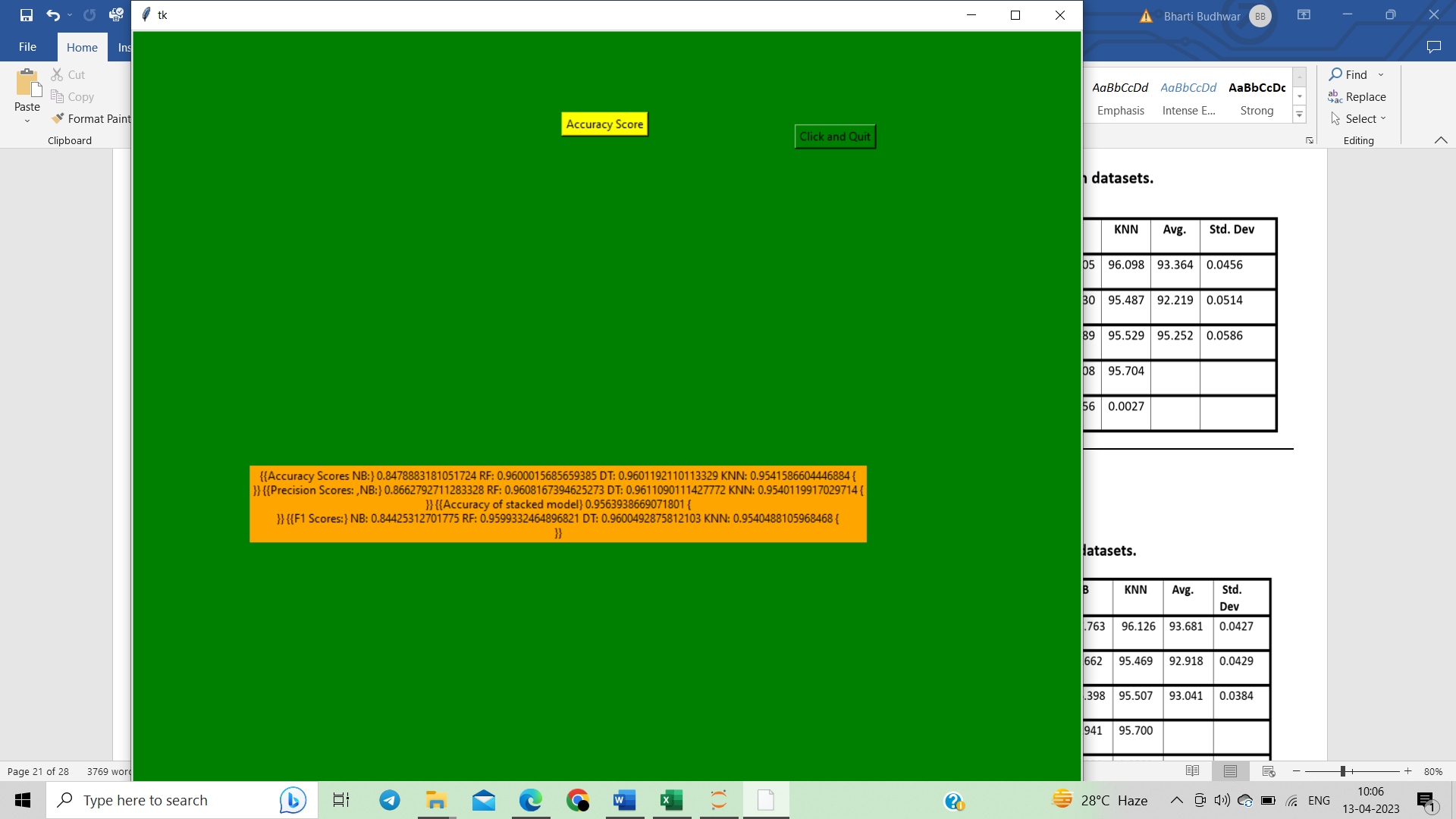
from sklearn.metrics import accuracy\_score

from sklearn.metrics import precision\_score

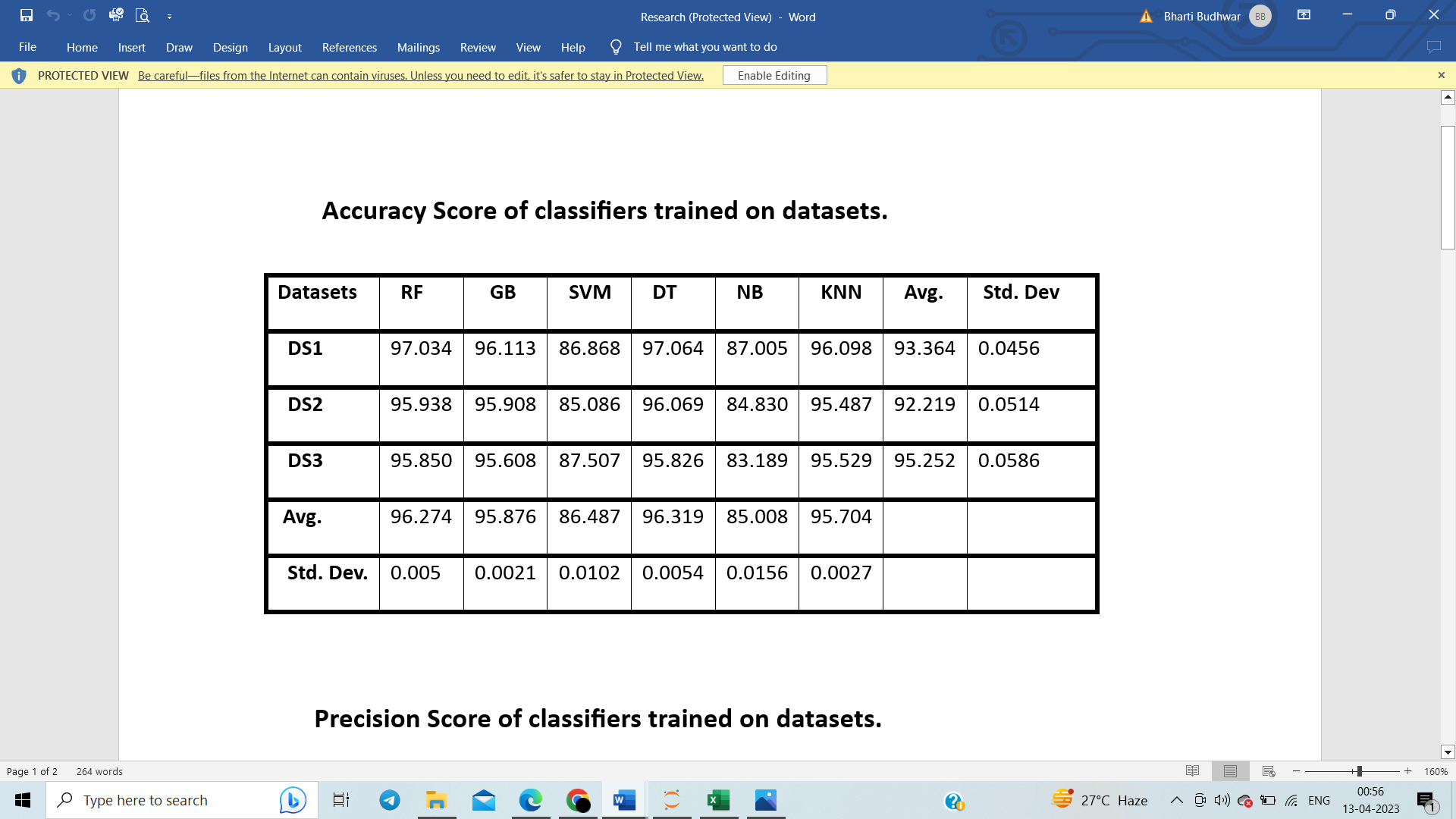
from sklearn.metrics import recall\_score

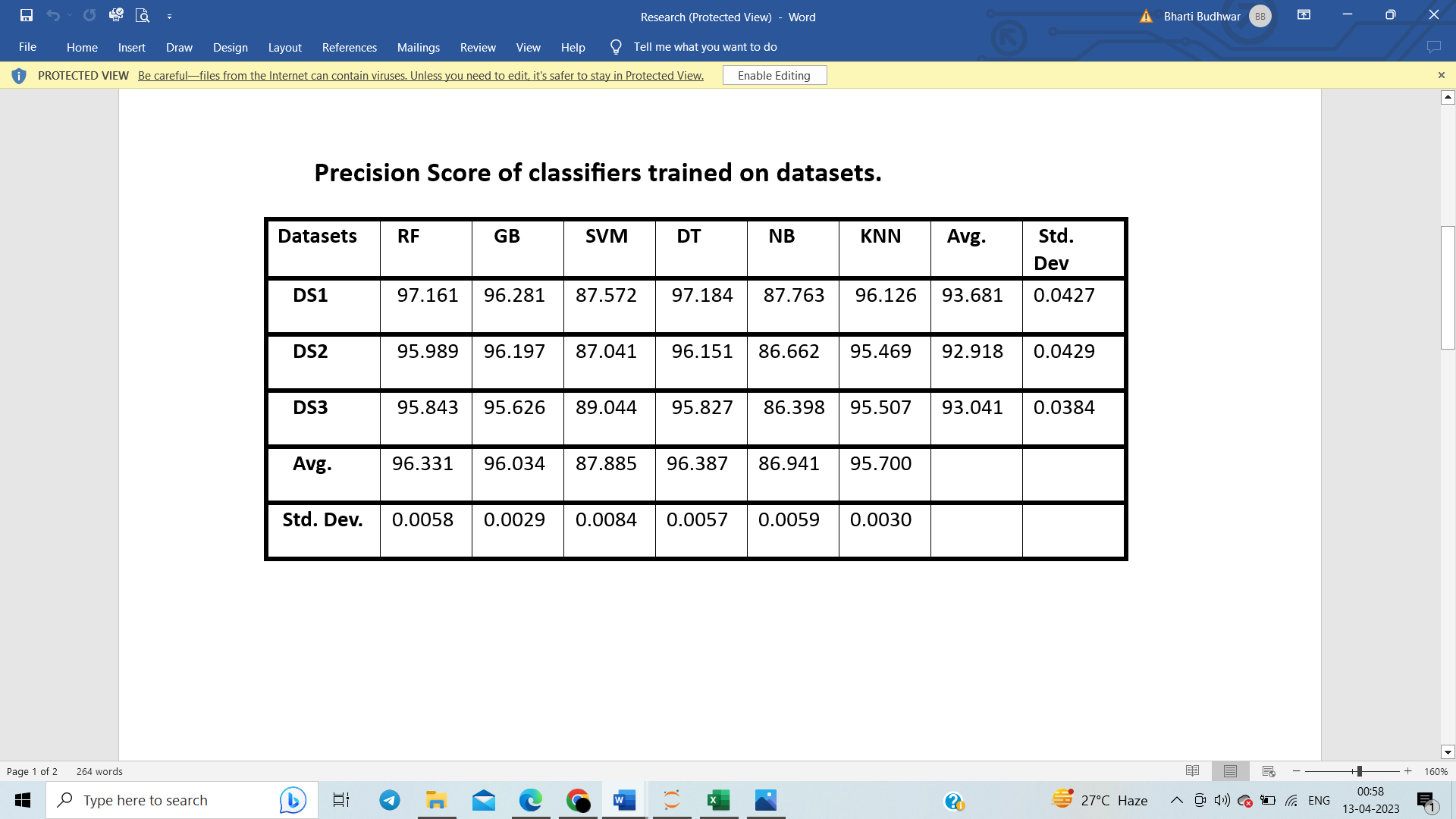


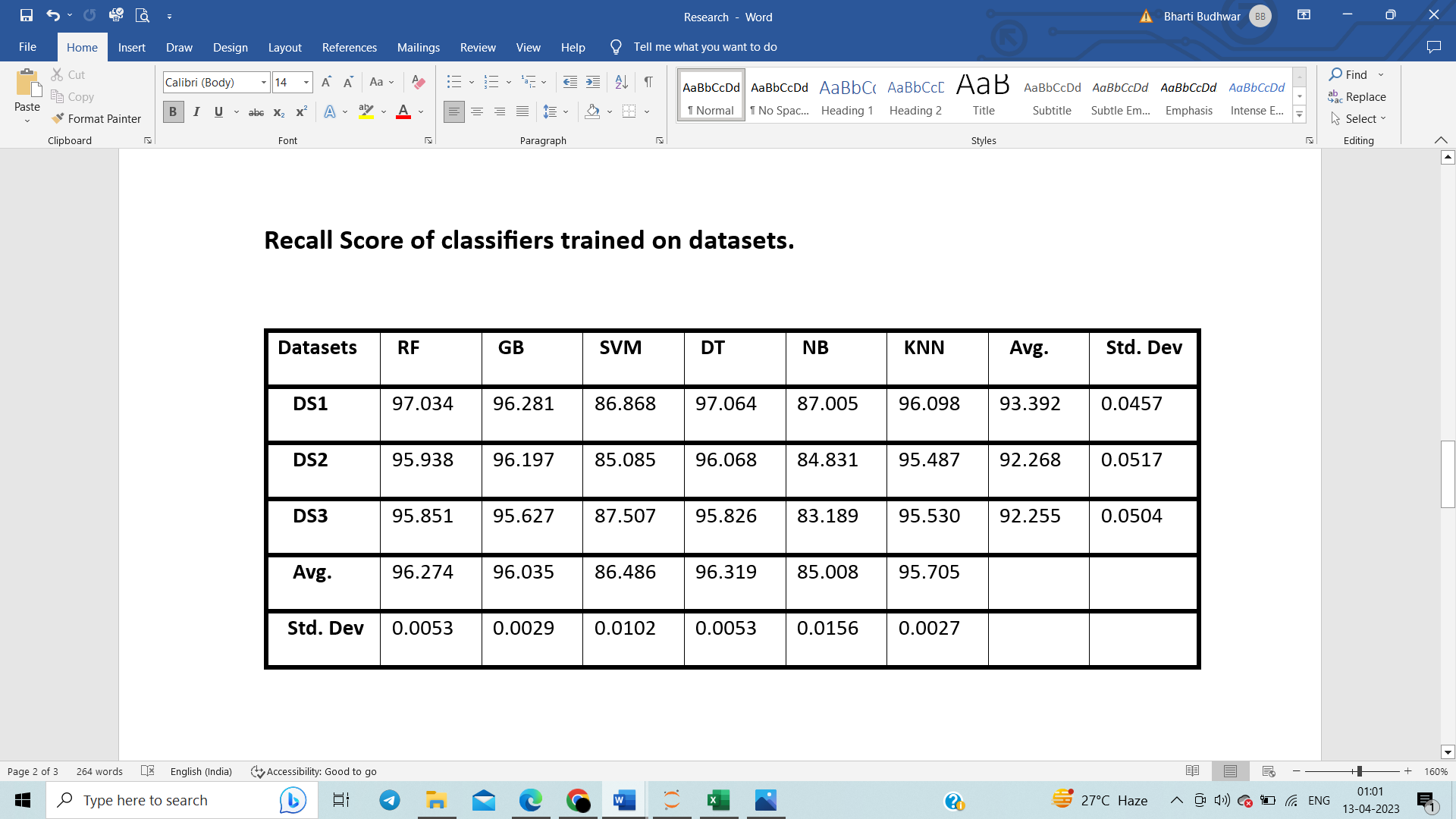
The output of the following will be as shown:

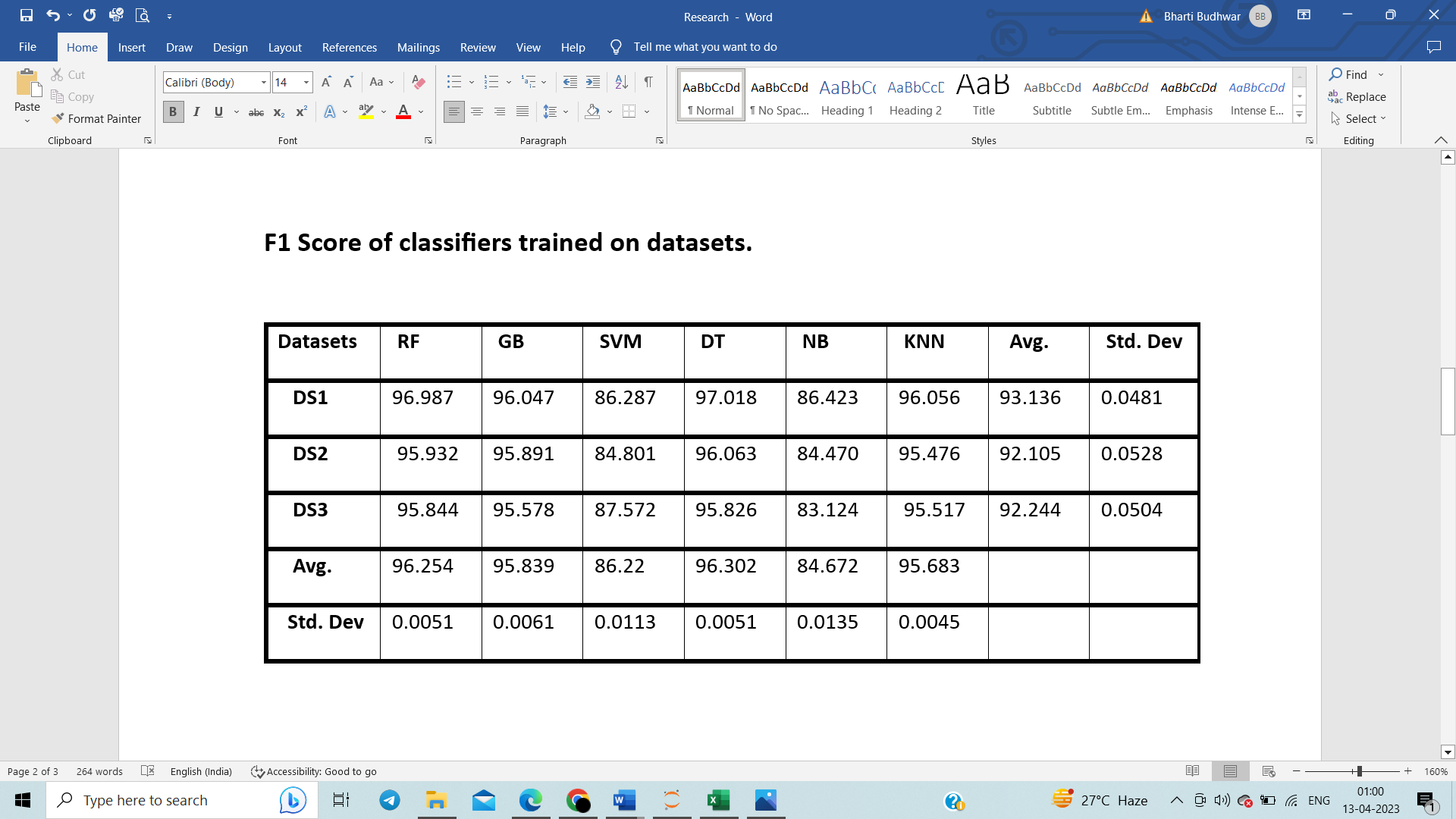


**RESULTS:**

****

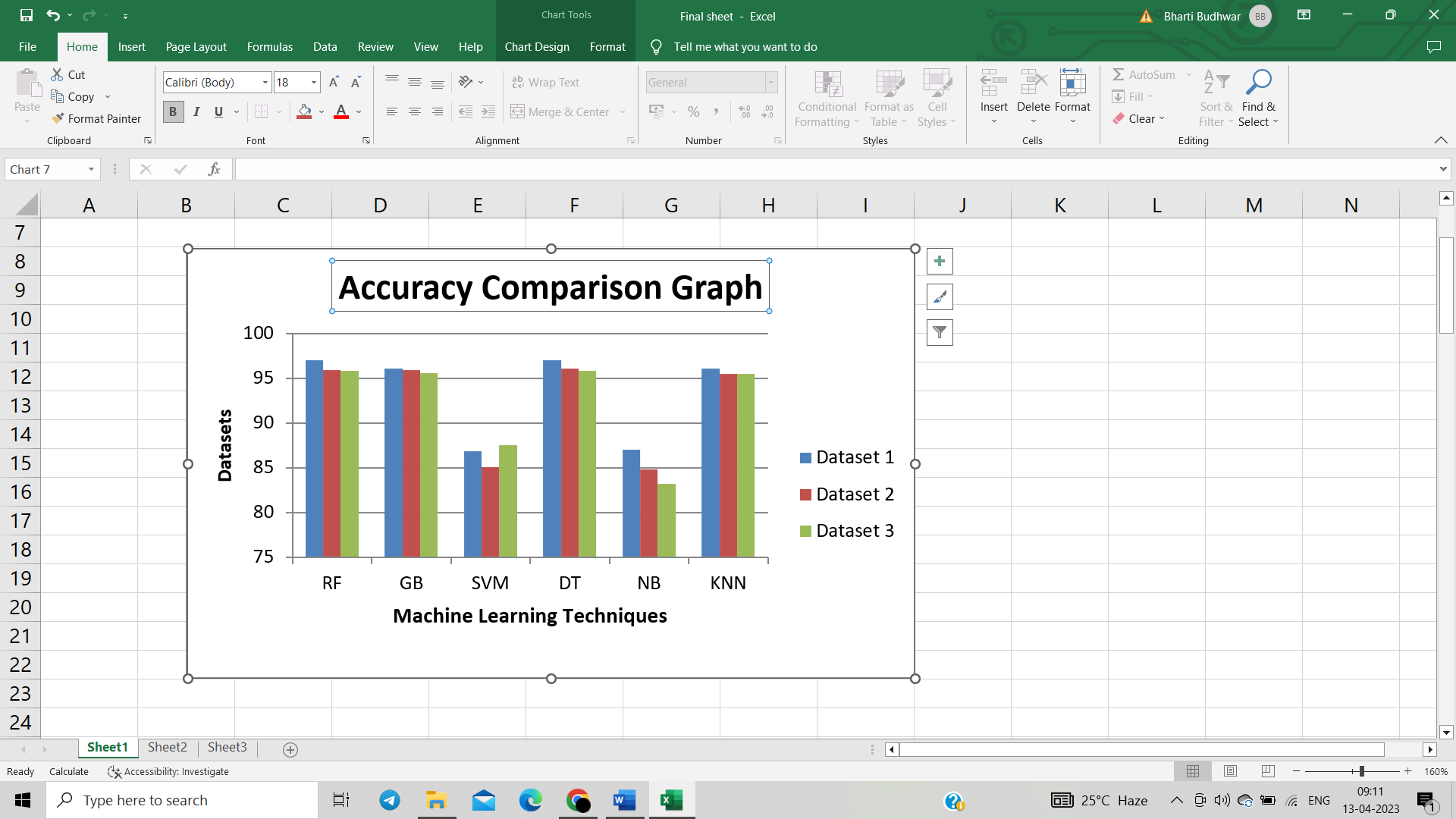
****

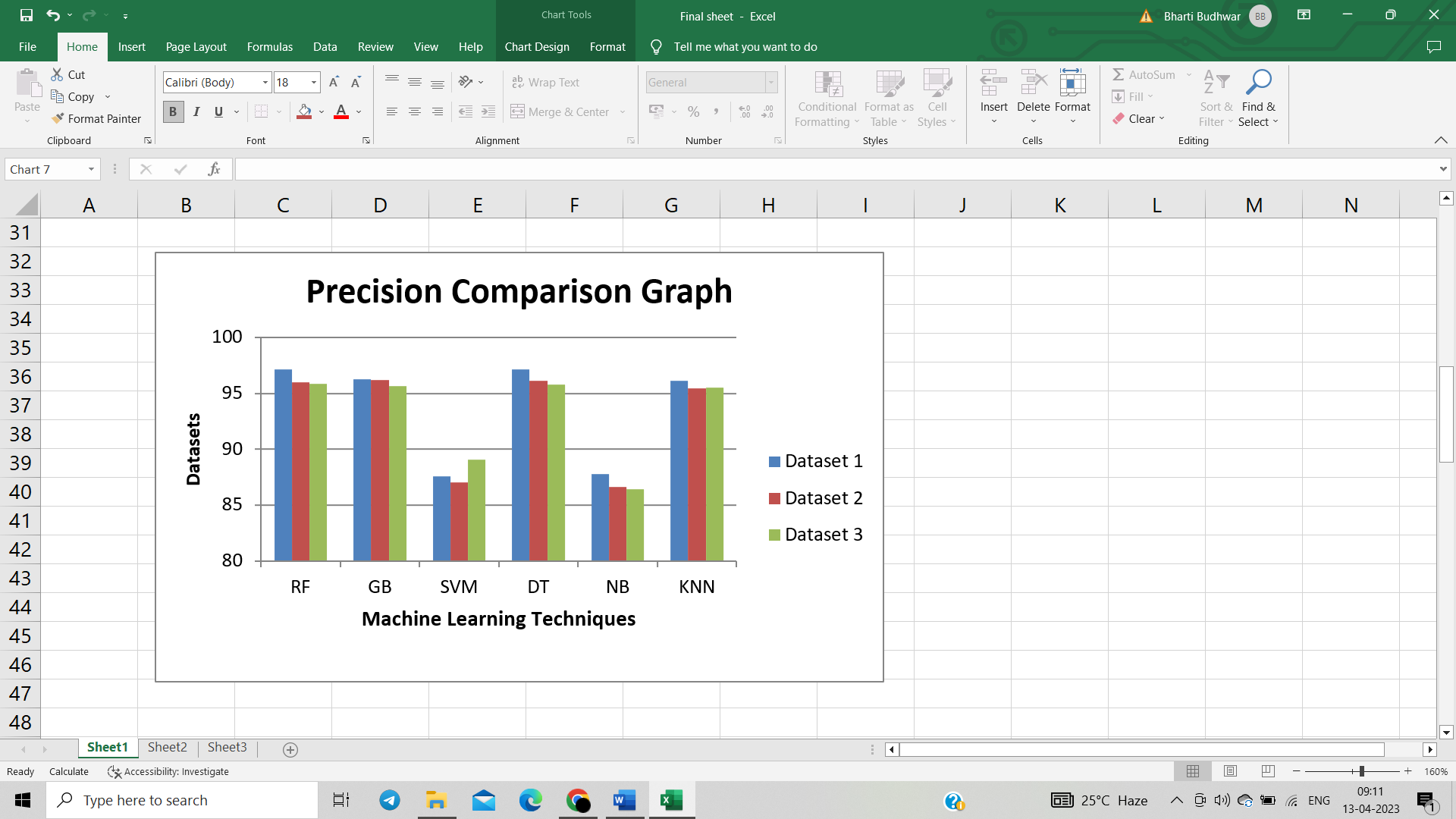
****

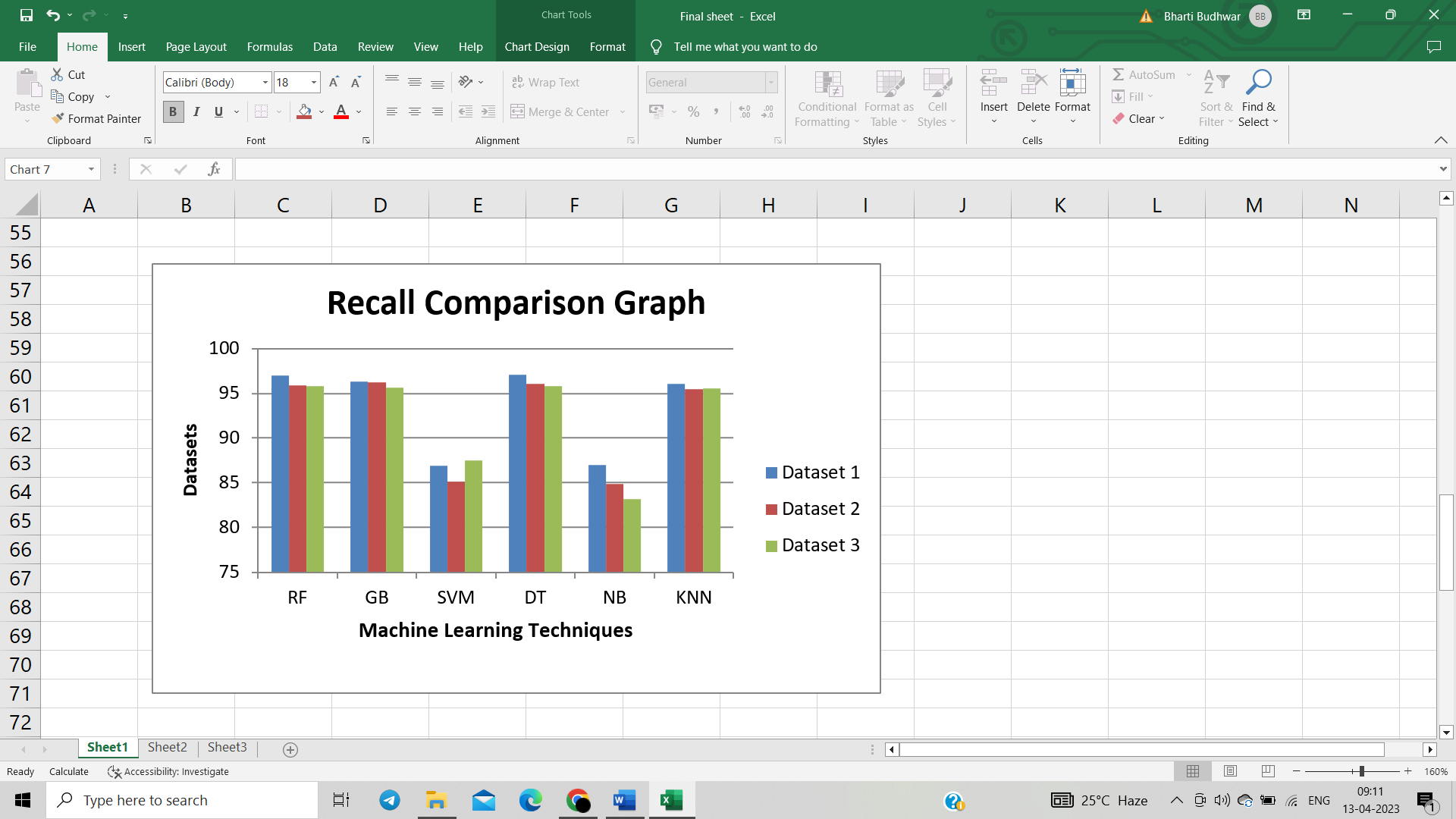
****

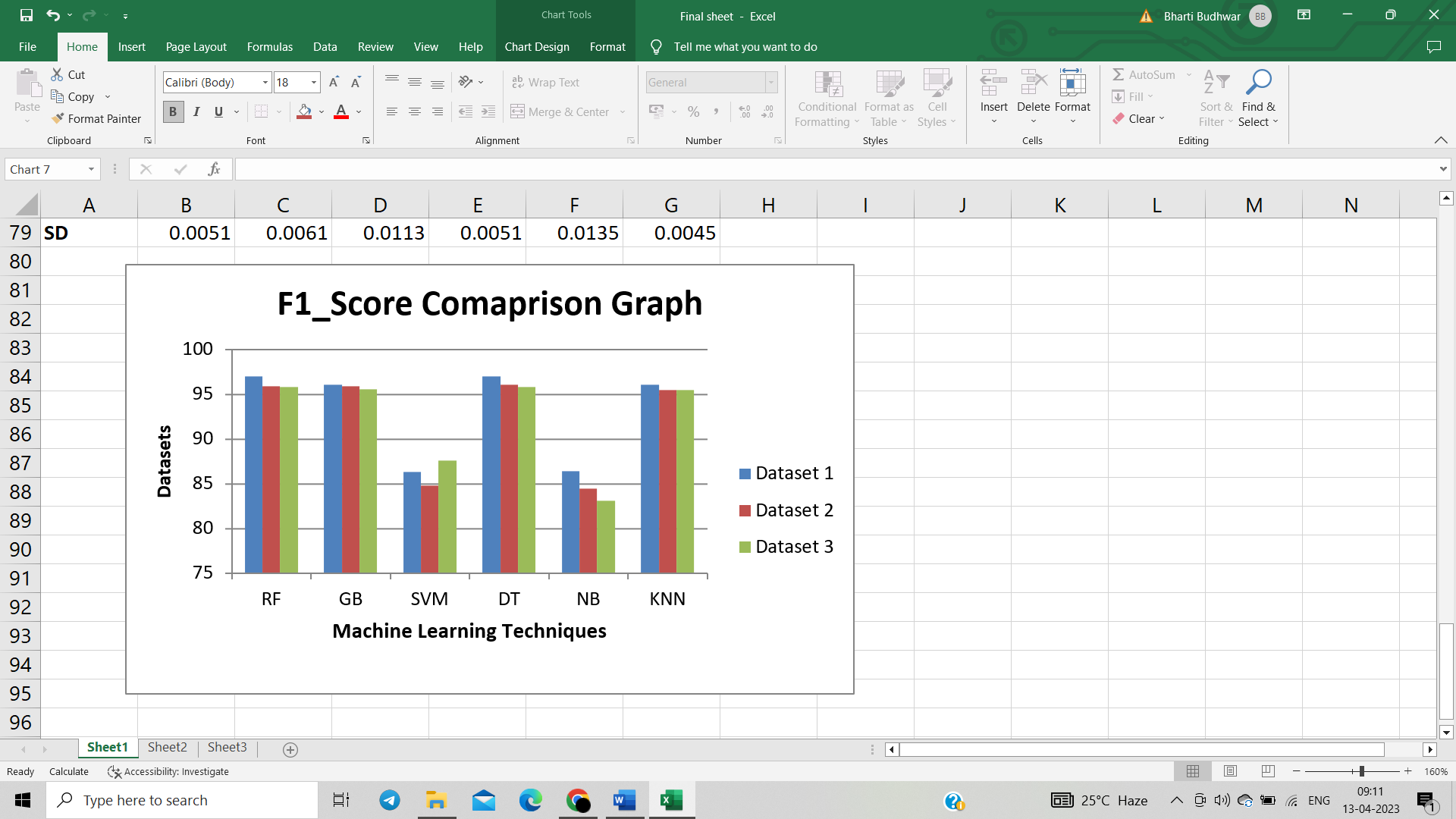
We were analyzing the effect of dataset size on accuracy of machine learning models, where the ratio of the dataset refers to how imbalance our dataset is i.e for the 1:1 ratio our dataset is balanced, for 1:2 ratio there is a slight imbalance in our dataset and for 1:3 our dataset is highly imbalance which means that the values are highly spread out. With these differences we can clearly analyze the change in accuracy with respect to the dataset sizes. We can see that overall Random Forest and k-Nearest Neighbour perform better with respect to all datasets.

**GRAPHS:**

****

****

****

****

**CONCLUSION:**

The overall performance of the classifiers depends on the extent to which a dataset represents the original distribution rather than its size, as size may add some difference as the randomness and biasness increase or the dataset is large it means it is more scattered and have given better result in some techniques.

The most robust machine learning models are Random Forest and k-Nearest Neighbour as it is giving overall best performance.

A robust machine learning model to limited dataset does not necessarily imply that it provides the best performance compared to other models. Our results are in agreement with previous studies which we have taken in consideration while doing this project.

**REFERENCES:**

1. K.J. Cios, I. Shin, Image recognition neural network: IRNN, Neurocomputing 7 (1995) 159–185, doi:10.1016/0925-2312(93)E0062- I.
2. D. Li, L. Yang, Deep Learning in Natural Language Processing, 1st Edition, Springer Publishing Company, 2018 Incorporated.
3. S.I. Ayon, M.M. Islam, Diabetes prediction: a deep learning approach, Int. J. Inf. Eng. Electr. Bus. 11 (2019) 21–27, doi:10.5815/IJIEEB.2019.02.03.
4. Sordo, M.; Zeng, Q. On sample size and classification accuracy: A performance comparison. In Biological and Medical Data Analysis; Springer: Berlin/Heidelberg, Germany, 2005; pp. 193–201.
5. Prusa, J.; Khoshgoftaar, T.M.; Seliya, N. The effect of dataset size on training tweet sentiment classifiers. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 9–11 December 2015; pp. 96–102.
6. Rahman, M.S.; Sultana, M. Performance of Firth-and logF-type penalized methods in risk prediction for small or sparse binary data. BMC Med. Res. Methodol. 2017, 17, 33. [CrossRef] [PubMed]
7. Marcoulides, G.A. Discovering Knowledge in Data: An Introduction to Data Mining, Daniel T. Larose. J. Am. Stat. Assoc. 2005, 100, 1465. [CrossRef]
8. Wieczorek, G.; Antoniuk, I.; Kurek, J.; Swiderski, B.; Kruk, M.; Pach, J.; Orłowski, A. BCT Boost Sementation with U-net in ´ TensorFlow.

**EVALUATION**

**USING DIFFERENT MACHINE LEARNING TECHNIQUES**