Privacy Preserving and Secure Machine Learning

Now-a-days machine learning and deep learning algorithms are employing in almost all fields to predict future events such as Heart disease prediction, road side traffic prediction, self-driving vehicle state prediction and many more. This algorithms get trained on past dataset and then by analysing new test data can predict future events. This algorithm accuracy can be enhanced by increasing dataset size and by using accurate data. This dataset can be obtained from public repository or can generate own dataset.

ML trained models can be vulnerable to attacks where attackers can inject poison (tamper or noisy data) data to ML models and based on prediction attacker can know the model behaviour and can extract sensitive data from train models. To overcome from this paper author of this paper suggesting to apply privacy to model while training models. In propose paper author experimenting with Differential Privacy and Homomorphic Encryption.

Differential Privacy algorithm will add noise to training features and ML model will get trained on noise data and if attacker extract sensitive data from the model then cannot separate original data from noise values and security will be provided to model sensitive training features data.

Homomorphic Encryption is an encryption technique wherein encrypted training data is used to perform computations and the final result, which is also in an encrypted form, can be decrypted by using cryptography techniques. Homomorphic Encryption techniques generally convert cipher texts of message which could be either a computation or a function applied on the message. This makes Homomorphic encryption a good way to perform computational operations on sensitive data as the said data is no longer in clear text and not directly apparent to the individuals who are not supposed to have direct access to the data such as system administrators who may have access to the servers for maintenance purposes. This algorithm can encrypt data partially or fully and by using this algorithm we can secure ML models sensitive training features from the attackers.

In propose work author just suggested to use differential privacy and Homomorphic algorithms to secure model but not used any dataset or ML models for experiments. So we have Heart Disease dataset whose values get perturbed using Differential Privacy and Homomorphic Encryption and then trained with Decision Tree algorithm and after encryption decision tree can able to get accuracy between 95 to 100% which proof that privacy to training featured cannot degrade ML performance as well as provide security to them.

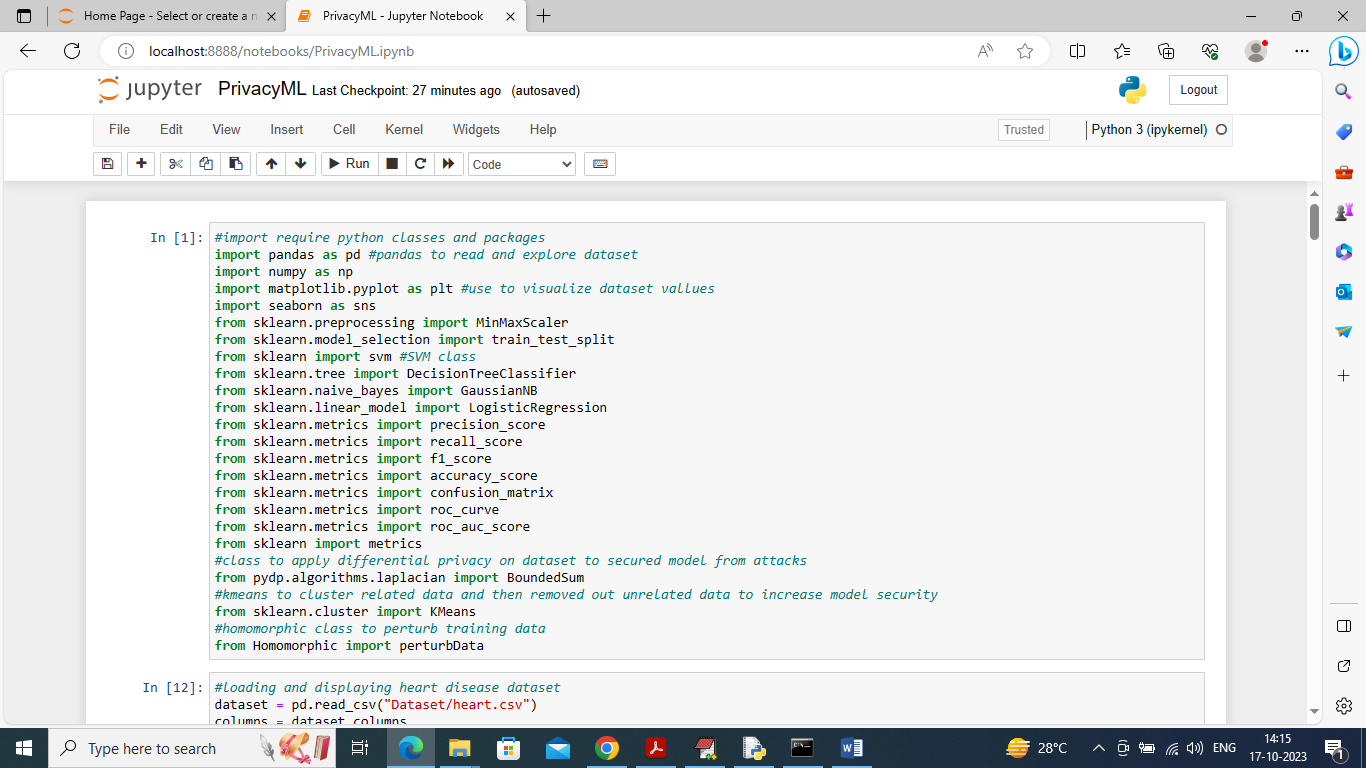
**Extension Concept**

In propose work author said that dataset can be collected from public repository and some malicious dataset repository may include unrelated data inside dataset which may affect ML model performance and to overcome from this as extension we have decided to filter dataset to find and remove out unrelated data before training secured model. As extension we have applied KMEANS algorithm on training features which will group similar data into same cluster and unrelated data to another cluster, so the cluster with least number of records can be consider as unrelated and then those cluster data will be removed out. After removing unrelated data then will apply Differential Privacy and Homomorphic encryption on training features to train and test ML models performance.

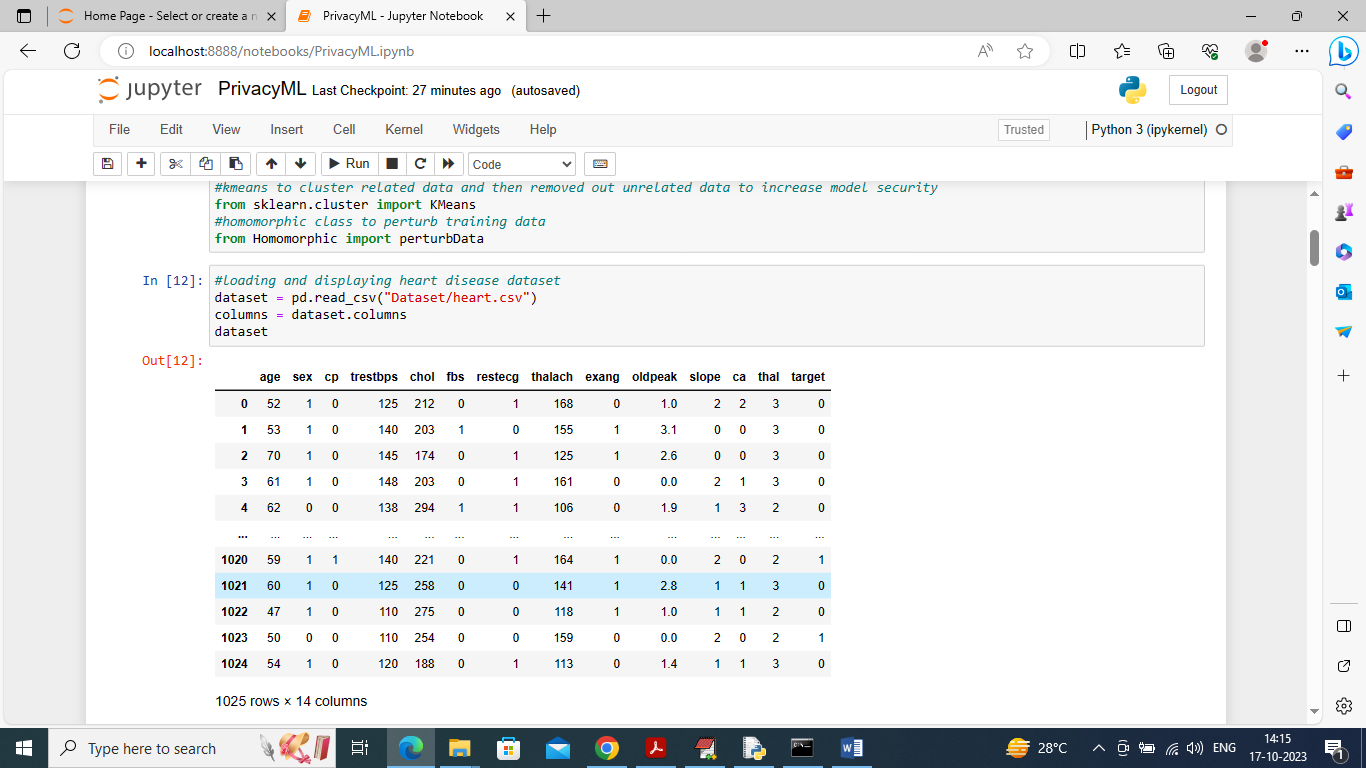
Algorithm performance is evaluated in terms of accuracy, precision, recall, FSCORE, confusion matrix and ROC curve.

SCREEN SHOTS

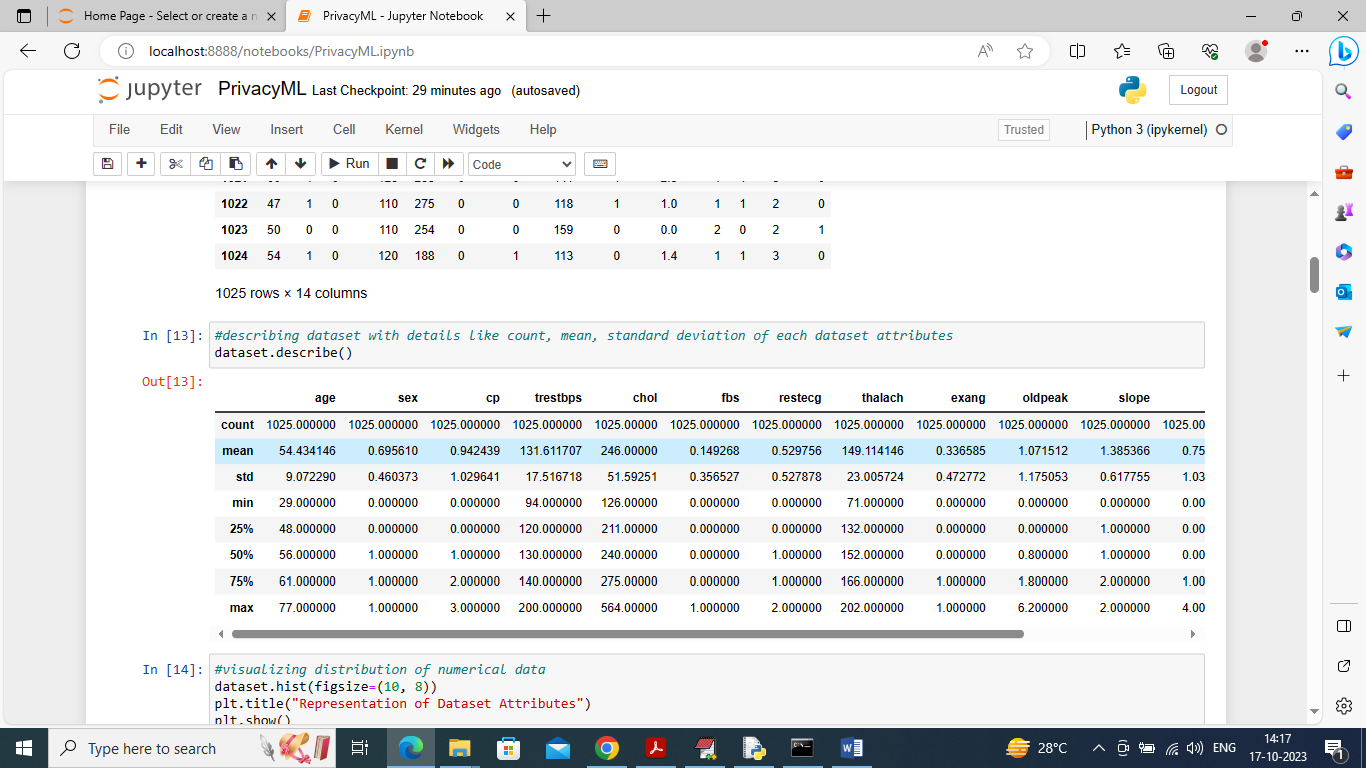
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



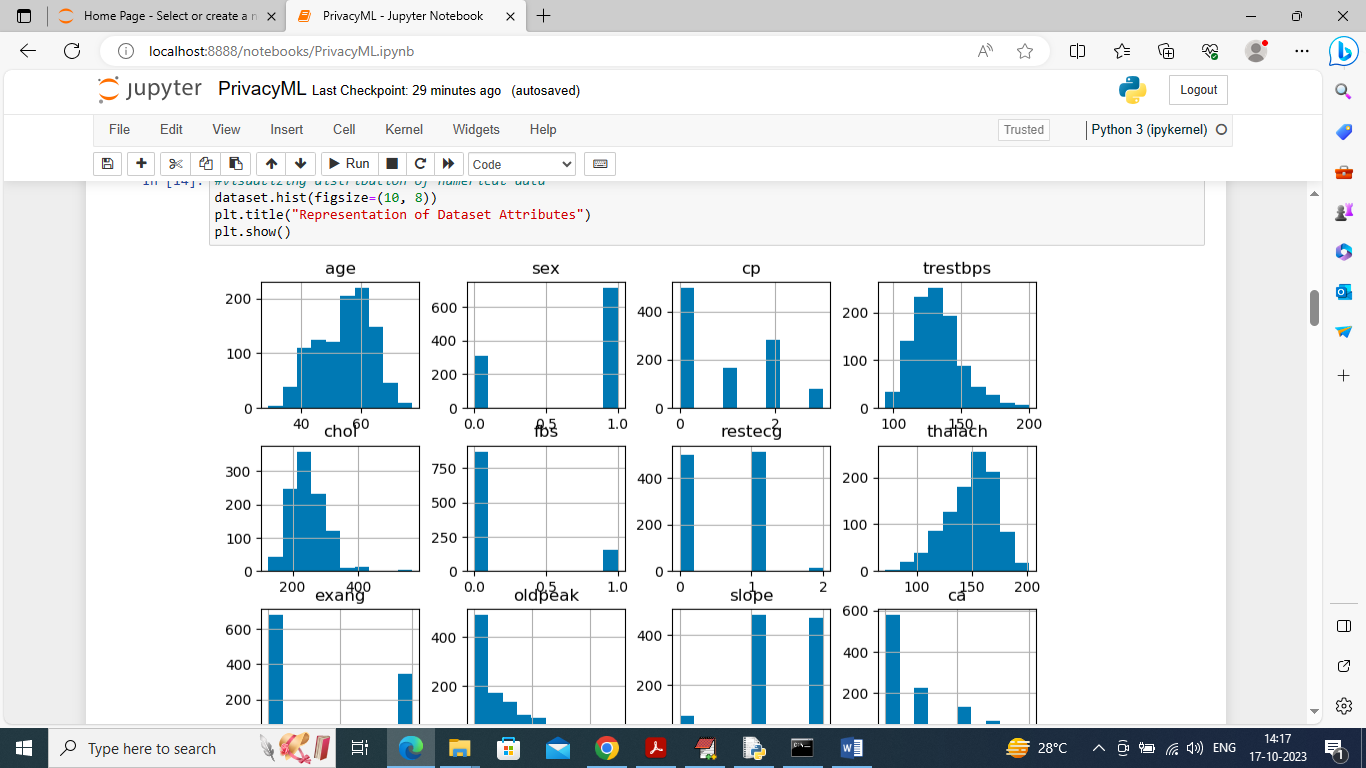
In above screen importing required python classes and packages and we are importing class of Homomorphic and Differential Privacy



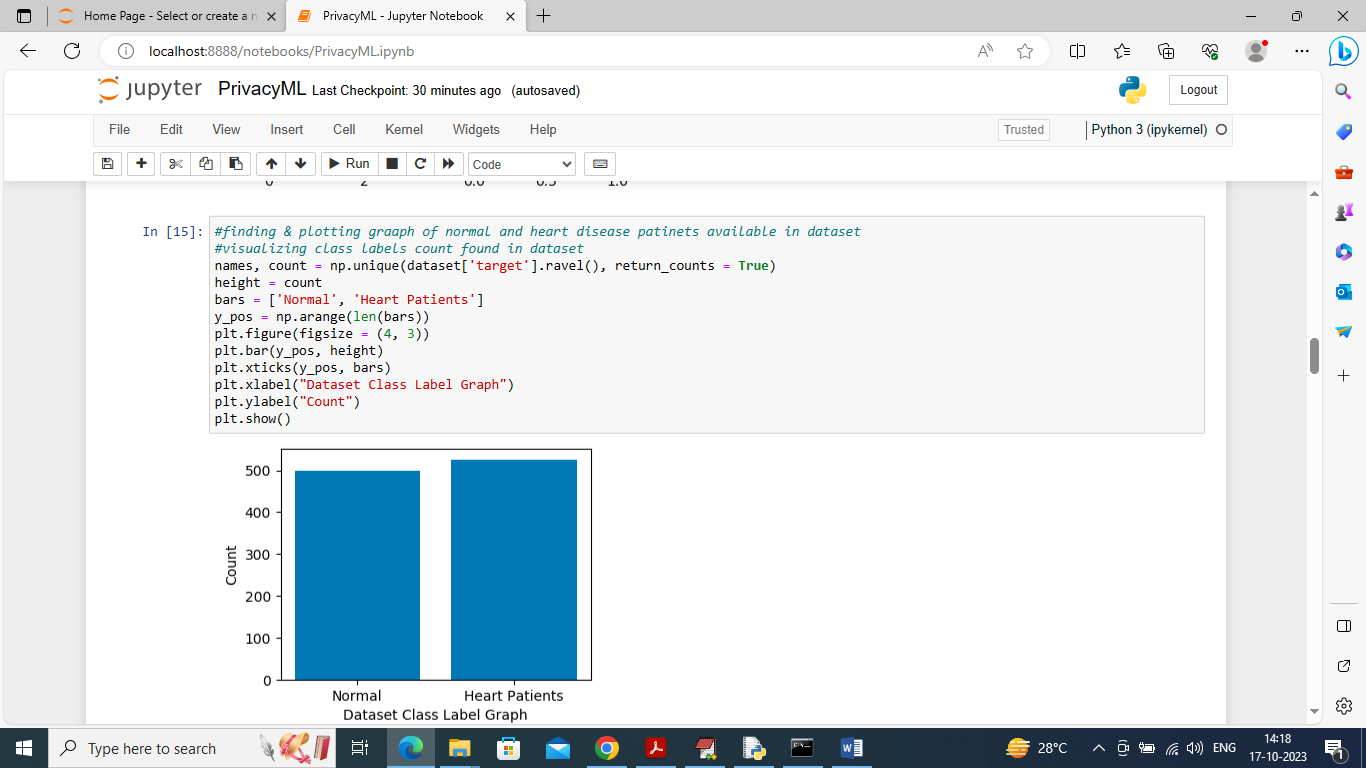
In above screen we are loading Heart dataset and this values are plain values and after applying Differential and Homomorphic encryption then above values will changed completely and still ML models can train and perform prediction on those changed values



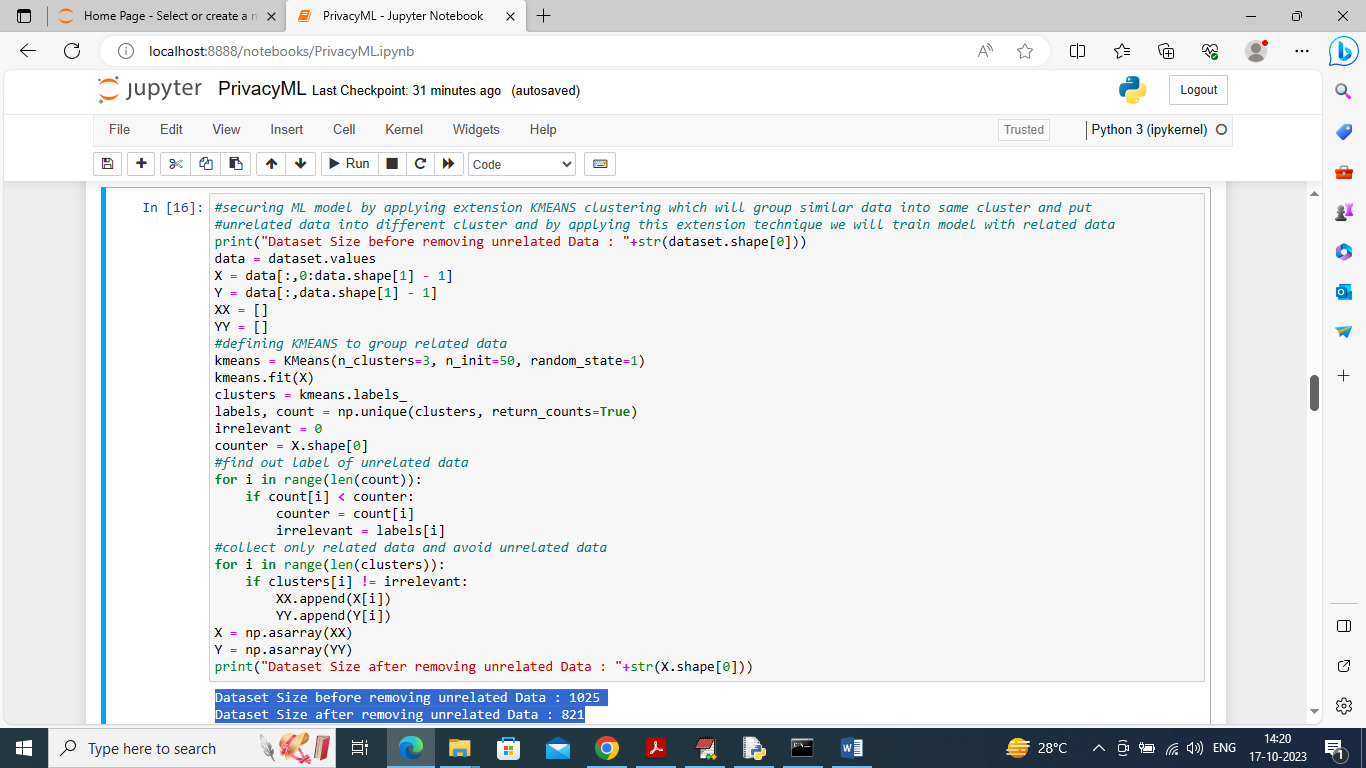
In above screen we are describing dataset values for each column in the form of count, mean, standard deviation etc.



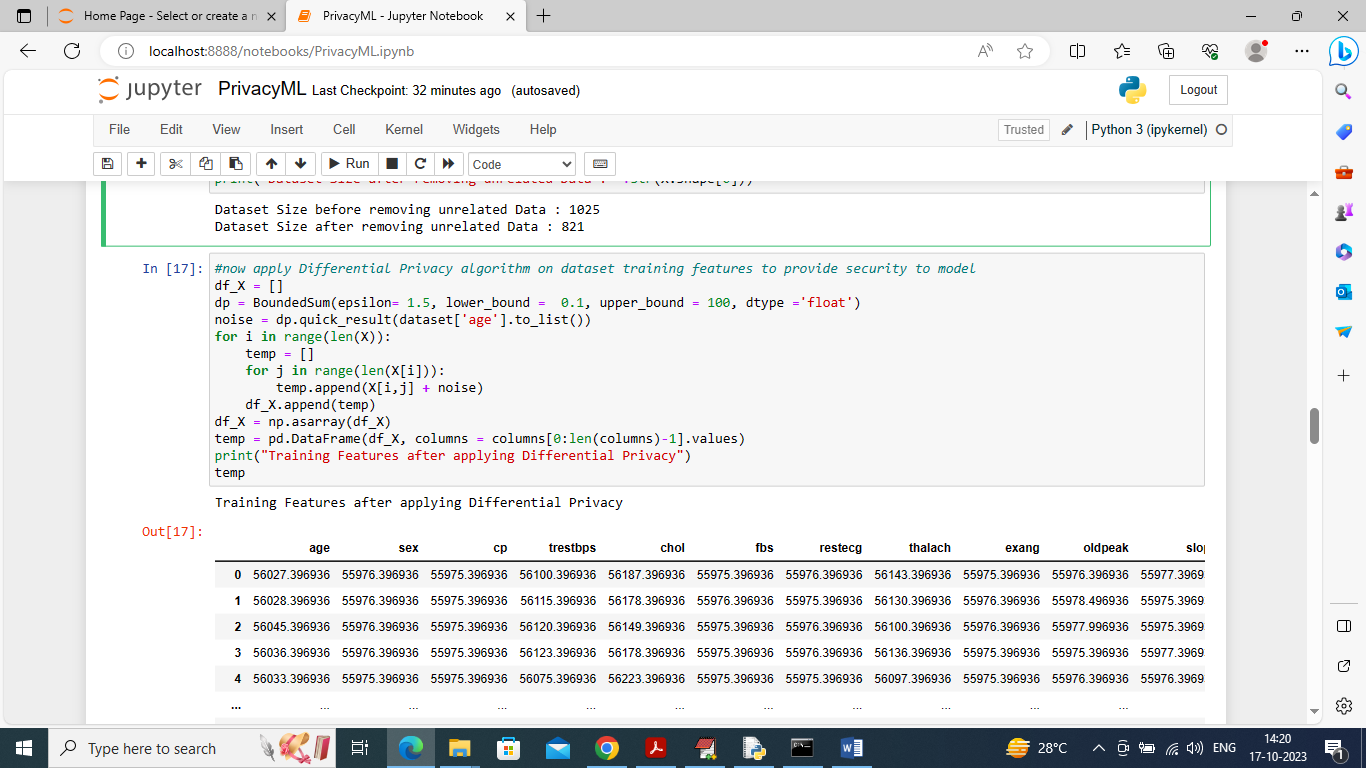
In above screen we are plotting distribution values for each dataset column in graph and this values will show how values are varied at different points of time



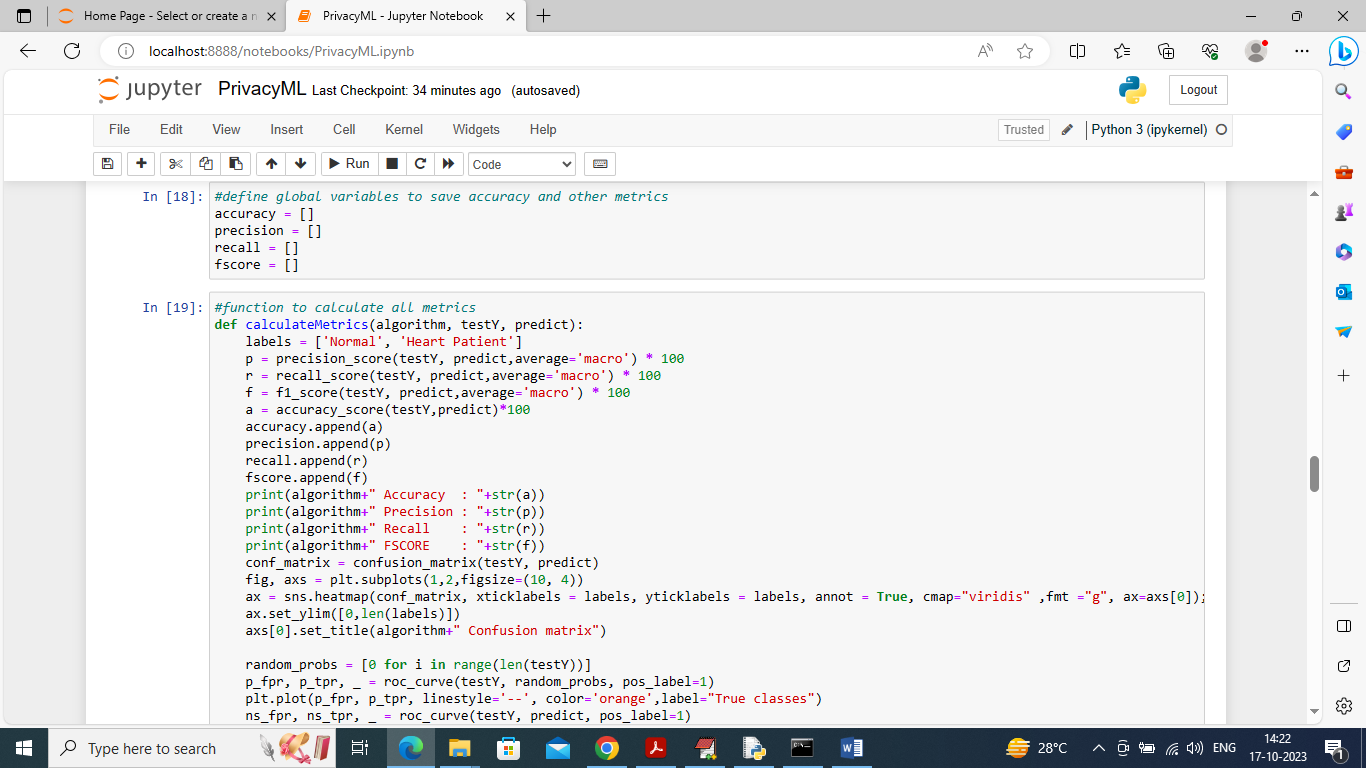
In above screen we are finding and plotting graph of normal and heart disease patients where x-axis represents patient’s type and y-axis represents counts



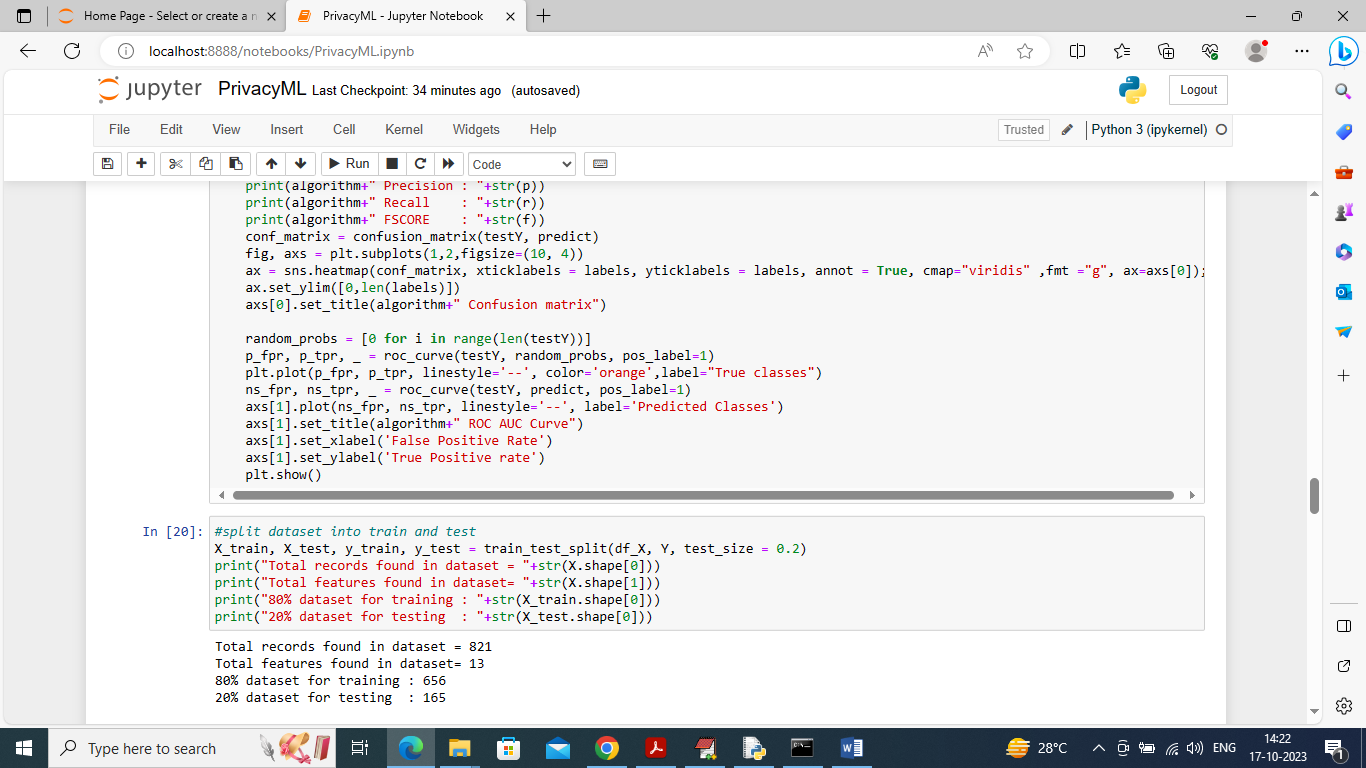
In above screen we are applying extension KMEANS clustering on dataset to removed out irrelevant data and then in blue colour text we can see dataset size before and after removal of irrelevant data



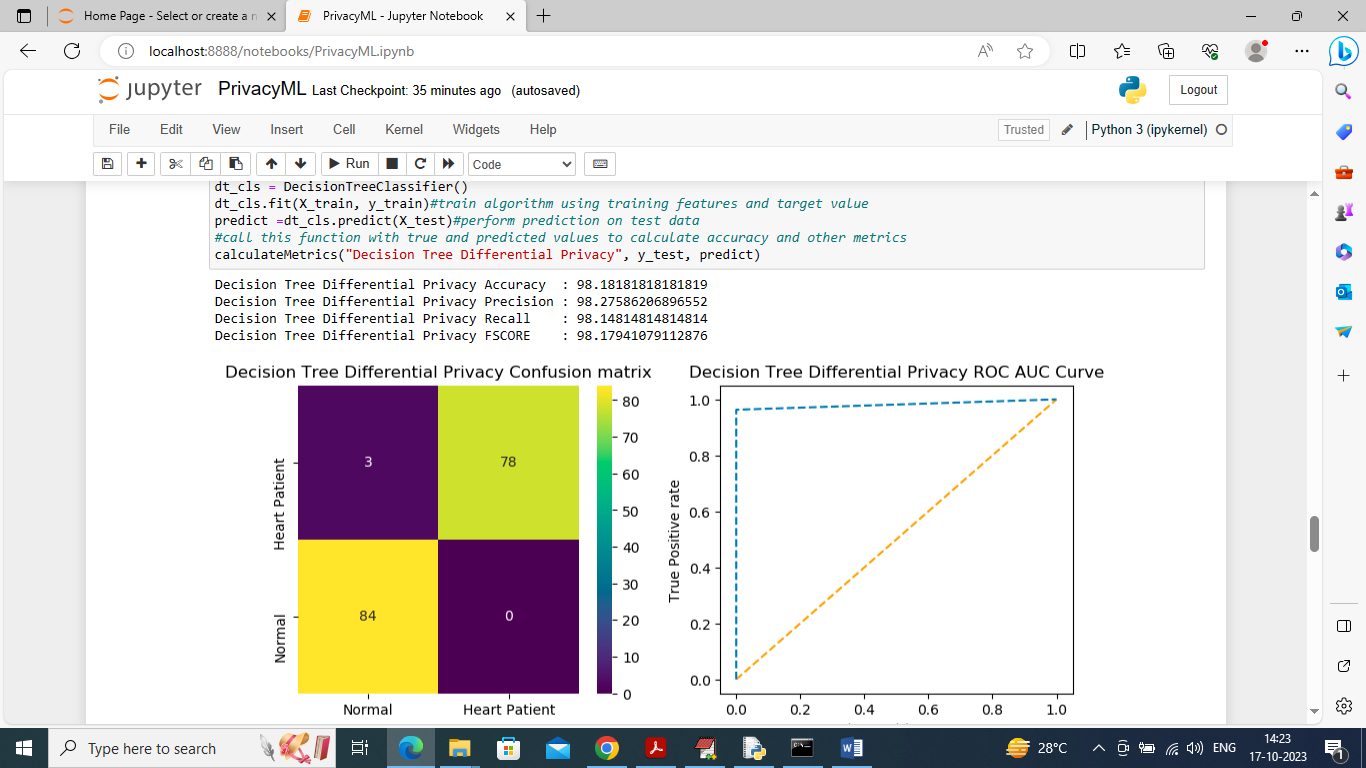
In above screen we are applying Differential Privacy algorithm on RELEVANT features dataset and after applying we can see entire dataset values get changed with noise data and this changed values you can see in above table



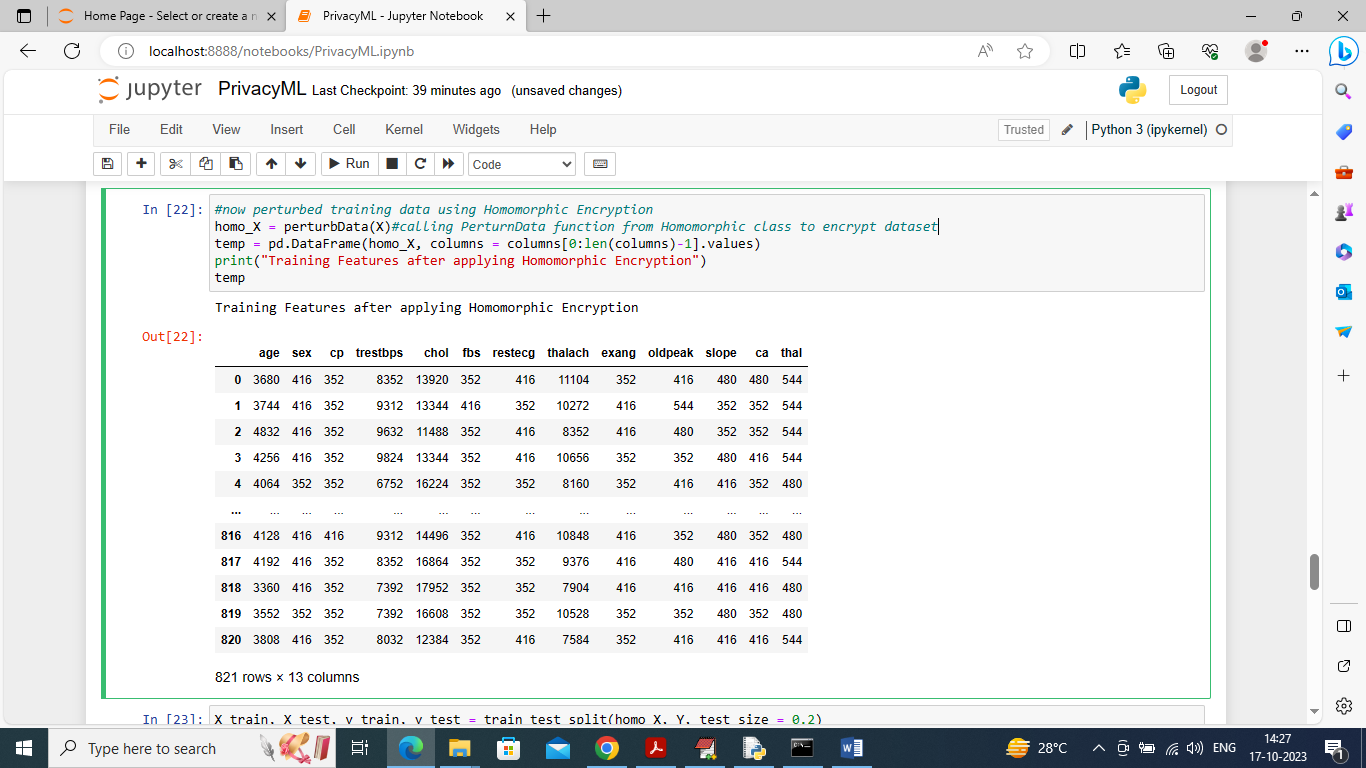
In above screen defining function to calculate accuracy and other metrics



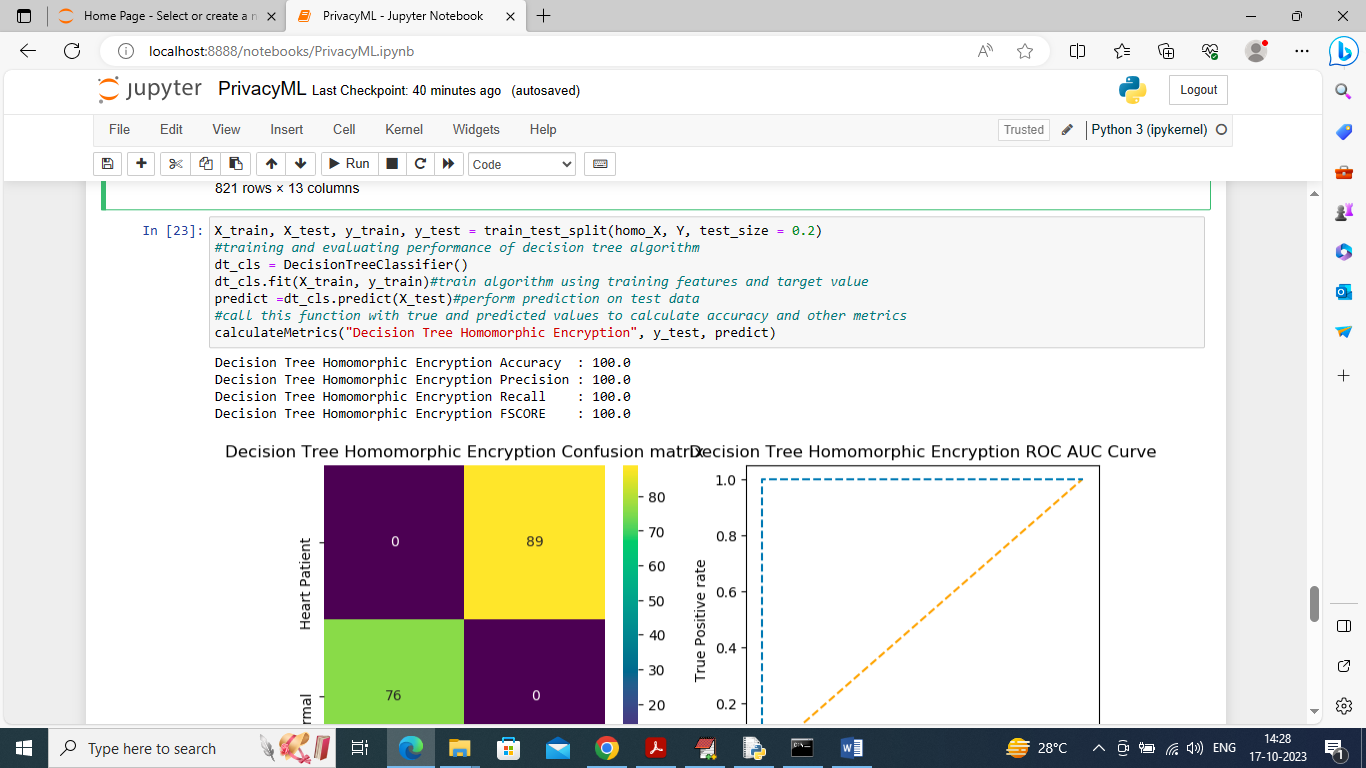
In above screen splitting dataset into train and test where application using 80% dataset for training and 20% for testing



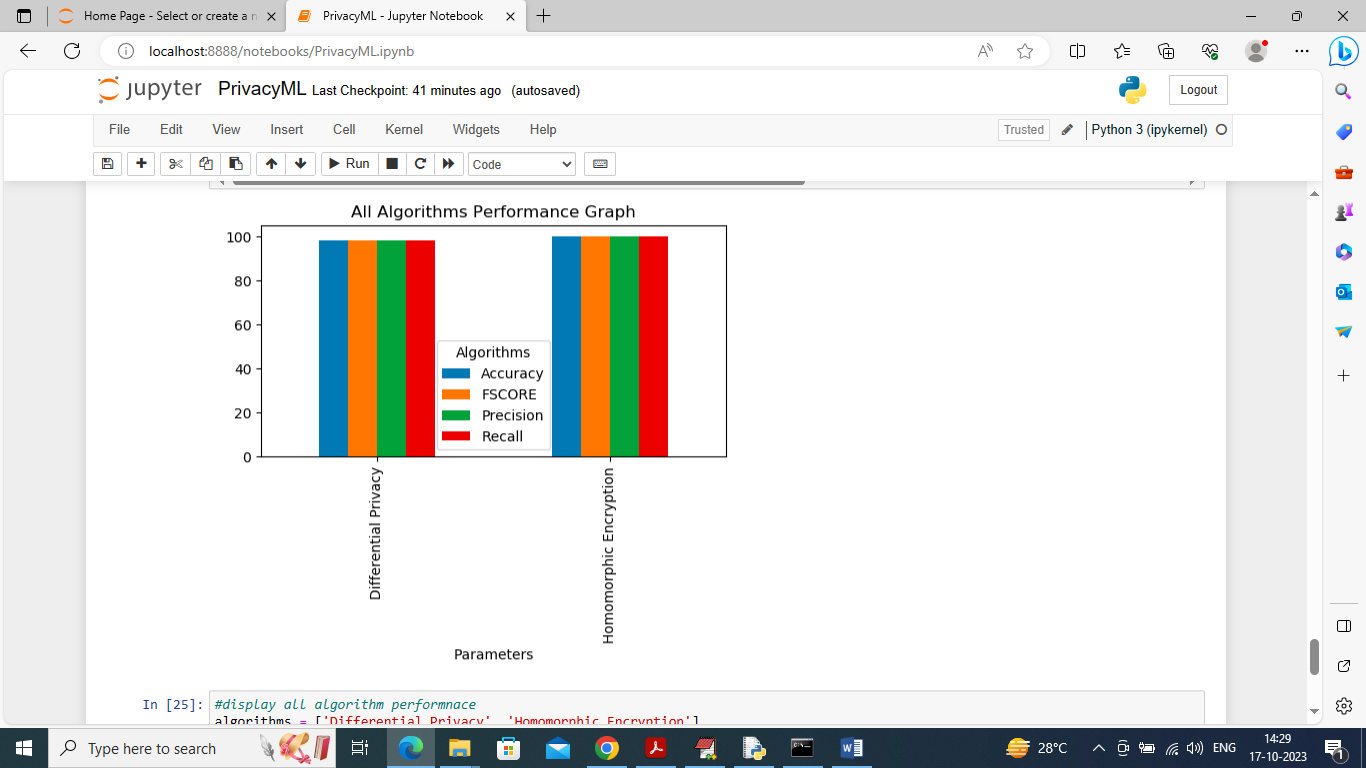
In above screen training Decision Tree algorithm on Differential Privacy values and after training we perform prediction on test data and then Decision tree got 98% accuracy on Differential privacy values which proves there is no effect on ML model after applying privacy. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels where all blue boxes represents incorrect prediction count and yellow, green represents correct prediction count. In ROC curve graph x-axis represents False positive Rate and y-axis represents True Positive rate and if blue line comes below orange line then all predictions are false and if goes above orange line then all predictions are correct



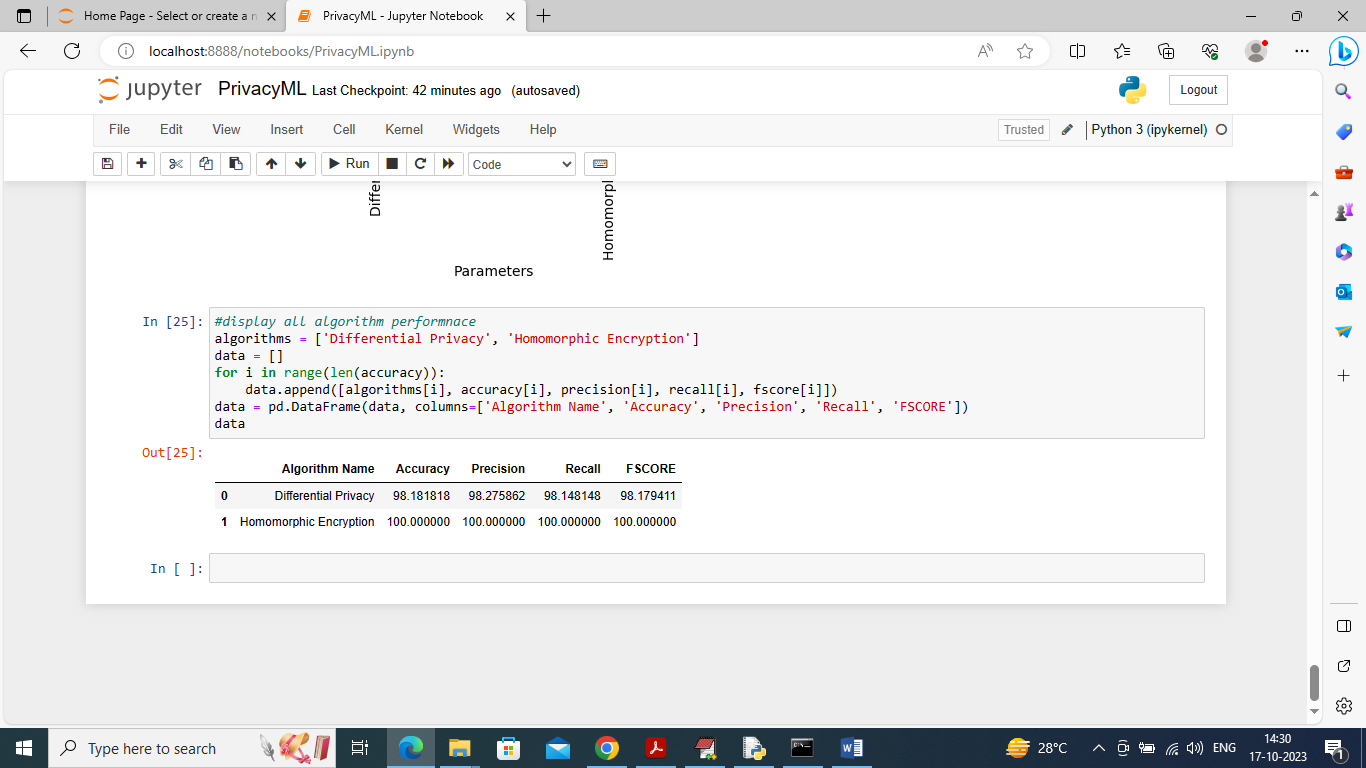
In above screen we are applying Homomorphic encryption on X training features for encryption and then encrypted values will be assigned to homo\_X and then we are displaying encrypted homo\_X values which you can see in above table



In above screen training decision tree on Homomorphic features and then decision tree got 100% accuracy and can see other metrics graph of trained model performance



In above graph displaying Decision tree performance on both Differential Privacy and Homomorphic features where x-axis represents technique name and y-axis represents accuracy and other metrics in different colour bars and from above graph we can say both techniques manages to give ML model accuracy more than 95%



In above screen displaying both algorithm performance in tabular format. So from above experiments we can see ML shows no change in performance even after model get privacy so by using this privacy we can secure model features from attackers