**Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems**

**Introduction:**

Today, all higher education institutions, especially computer and engineering colleges, face challenges in the admissions process. Each university should strive for an admissions system based on valid and reliable admissions criteria that select candidates likely to succeed in its programs. In addition, each university should use the best possible techniques for predicting applicants’ future academic performance before admitting them. This would support university decision makers as they set efficient admissions criteria. However, most higher education institutions face challenges when they analyze their large educational databases to predict students’ performance. This is because they use only conventional statistical methods rather than new and efficient predictive techniques such as Educational Data Mining (EDM), which is the most popular technique to evaluate and predict students’ performance . EDM is the process of extracting useful information and patterns from a huge educational database, which can then be used to predict students’ performance. As a result of better information, student performance can be more effectively improved through more effective strategic programs.

Data Description:

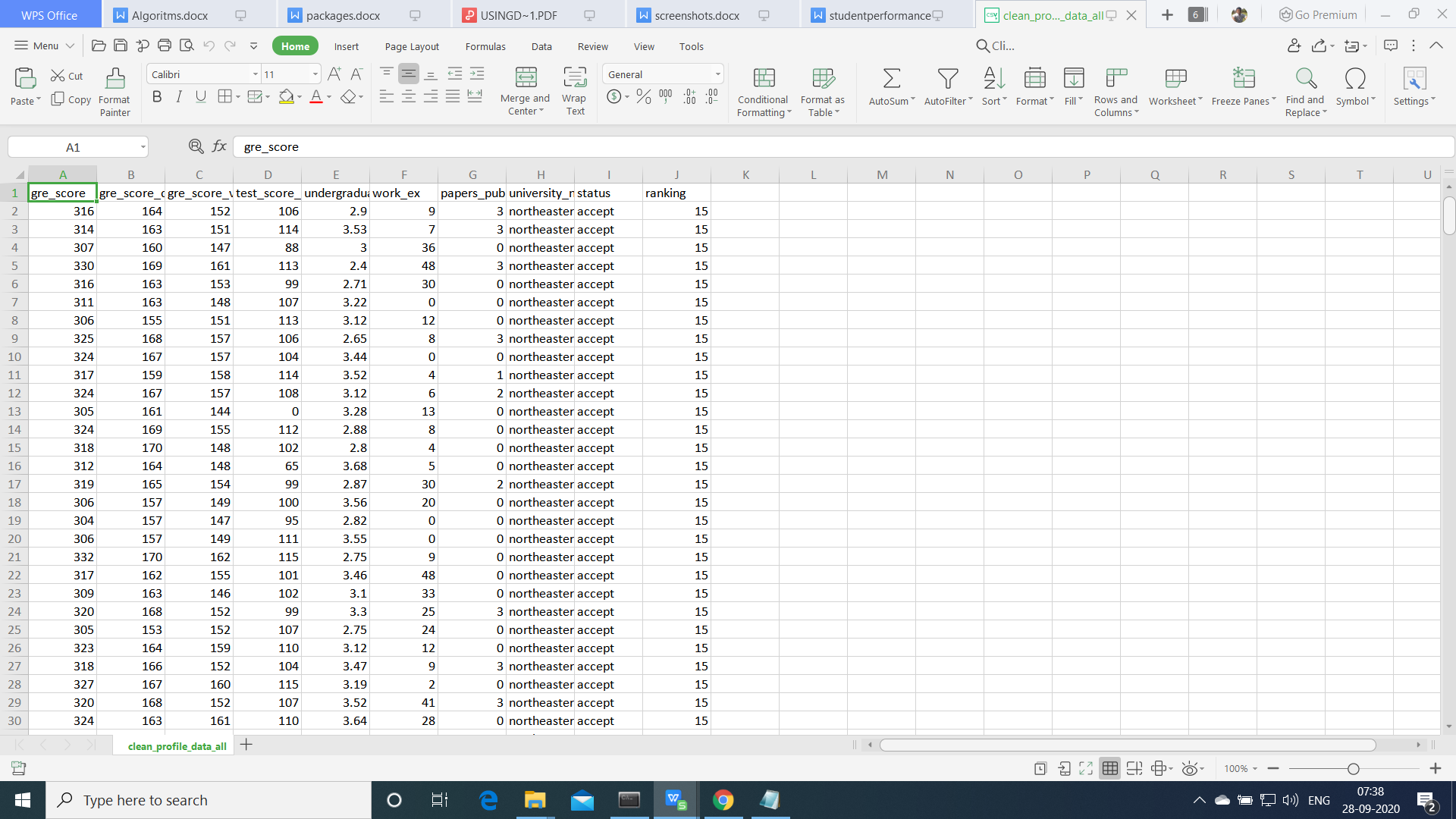
**Existing Methods:**In Existing methods , we are using Decision Trees, Support Vector Machines, and Naïve Bayes will help classifying admitted applicant who will accepted and not accepted. Alogrithms has less performance and acuracy.

**Proposed Method:**

In this paper, we proposing ANN algorithms which will increase the model performance and over and undefitting

**Dataset:Location:**<https://www.kaggle.com/c/datasciencebowl/data>

The dataset consists of 480 student records and 16 features. The features are classified into three major categories: (1) Demographic features such as gender and nationality. (2) Academic background features such as educational stage, grade Level and section. (3) Behavioral features such as raised hand on class, opening resources, answering survey by parents, and school satisfaction.



**Algorithms:**

**Support Vector Machine**

“Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle" \t "https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/_blank) which can be used for both classification or regression challenges. However,  it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_1.png)

Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

**Logistic regression**

Logistic regression is named for the function used at the core of the method, the logistic function.

The [logistic function](https://en.wikipedia.org/wiki/Logistic_function), also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value)

Where e is the [base of the natural logarithms](https://en.wikipedia.org/wiki/E_(mathematical_constant)) (Euler’s number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform. Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.



Logistic Function

Now that we know what the logistic function is, let’s see how it is used in logistic regression.

# Naive Bayes Classifiers

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) \* P(d2|H) and so on.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

# Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **"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.

**Decision tree Algorithm:**

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree,testing the attribute specified by this node,then moving down the tree branch corresponding to the value of the attribute as shown in the above figure.This process is then repeated for the subtree rooted at the new node

1. **ANN** is rarely used for predictive modelling. The reason being that Artificial Neural Networks (ANN) usually tries to over-fit the relationship. ANN is generally used in cases where what has happened in past is repeated almost exactly in same way. For example, say we are playing the game of Black Jack against a computer. An intelligent opponent based on ANN would be a very good opponent in this case (assuming they can manage to keep the computation time low). With time ANN will train itself for all possible cases of card flow. And given that we are not shuffling cards with a dealer, ANN will be able to memorize every single call. Hence, it is a kind of machine learning technique which has enormous memory. But it does not work well in case where scoring population is significantly different compared to training sample. For instance, if I plan to target customer for a campaign using their past response by an ANN. I will probably be using a wrong technique as it might have over-fitted the relationship between the response and other predictors.

**Steps for Machine Learning Algorithms**

1. Install Anaconda Latest Version
2. Open anaconda Prompt
3. Conda create -n tf python=3.7
4. Conda activate tf
5. Install require softwares

scikit-image==0.17.2

scikit-learn==0.23.2

pandas==1.1.1

matplotlib==3.3.1

Pillow==7.2.0

plotly==4.10.0

opencv-python==4.4.0.42

spacy==2.3.2

lightgbm==3.0.0

mahotas==1.4.11

matplotlib==3.3.1lightgbm==3.0.0

mahotas==1.4.11

nltk==3.5

matplotlib==3.3.1

xgboost==1.2.0

Jupyter

1. Activate environment for jupyter notebook(For execute the in jupter notebook)

python -m ipykernel install --user --name=

1. Goto project Directory

Note: For Text related project. Need to Download

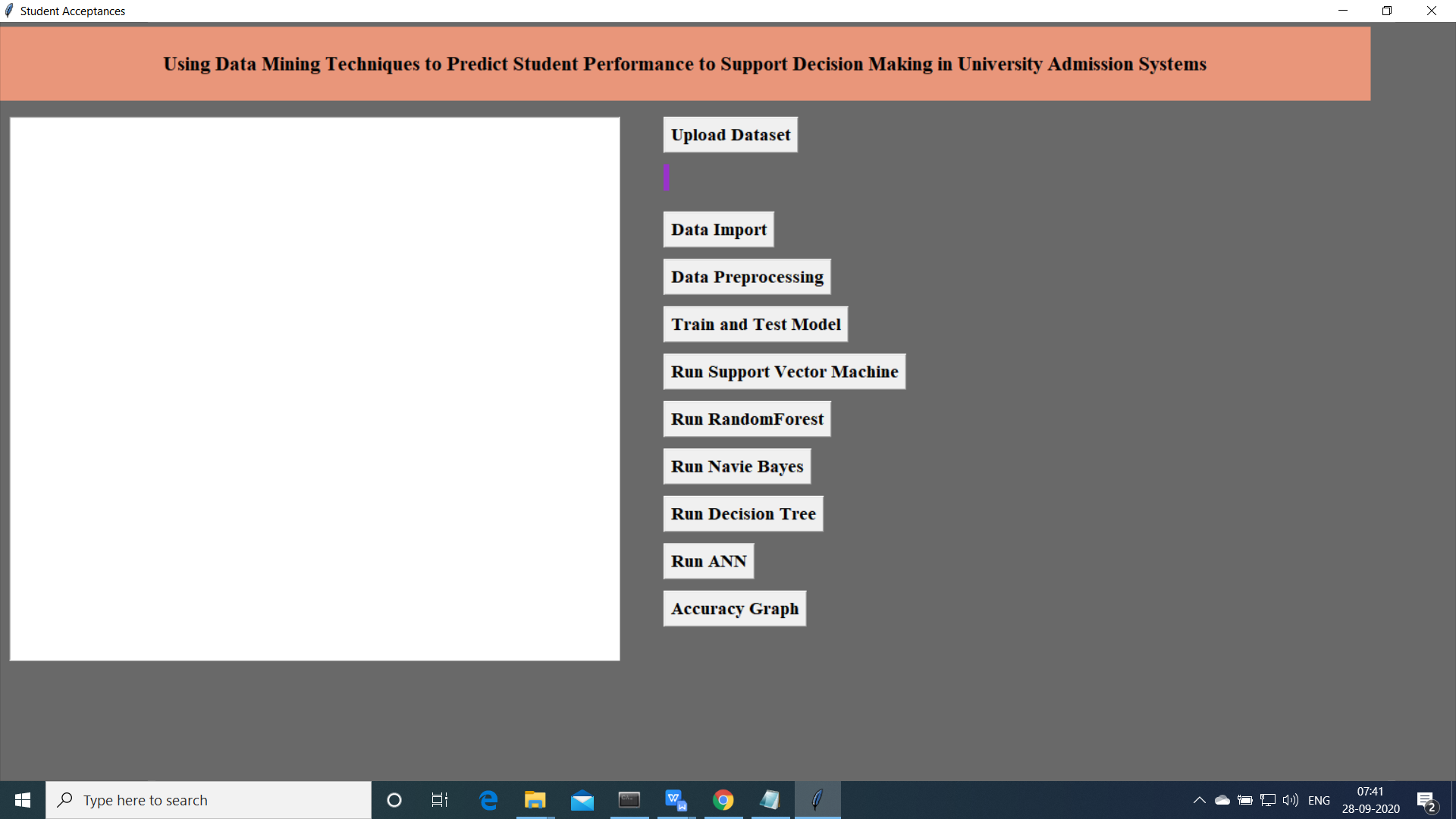
1. Open anaconda Prompt
2. Python
3. Import nltk
4. Nltk.download()

**Project Development Modules:**

1. **Data Collection:**Collect sufficient data samples and legitimate software samples. 
2. **Data Preporcessing**:Perform effective data processing on the sample and extract the features. 
3. **Train and Test Modelling: Split the data into train and test data Train will be used for trainging the model and Test data to check the performace**
4. **Feature Selection:**Further select the main features for classification.
5. **Modelling:** SVM Navie bayes, Random FOrest,KNN,Ada boost, Decision tree, Ada boost with randomforest . Combine the training using machine learning algorithms and establish a classification model.

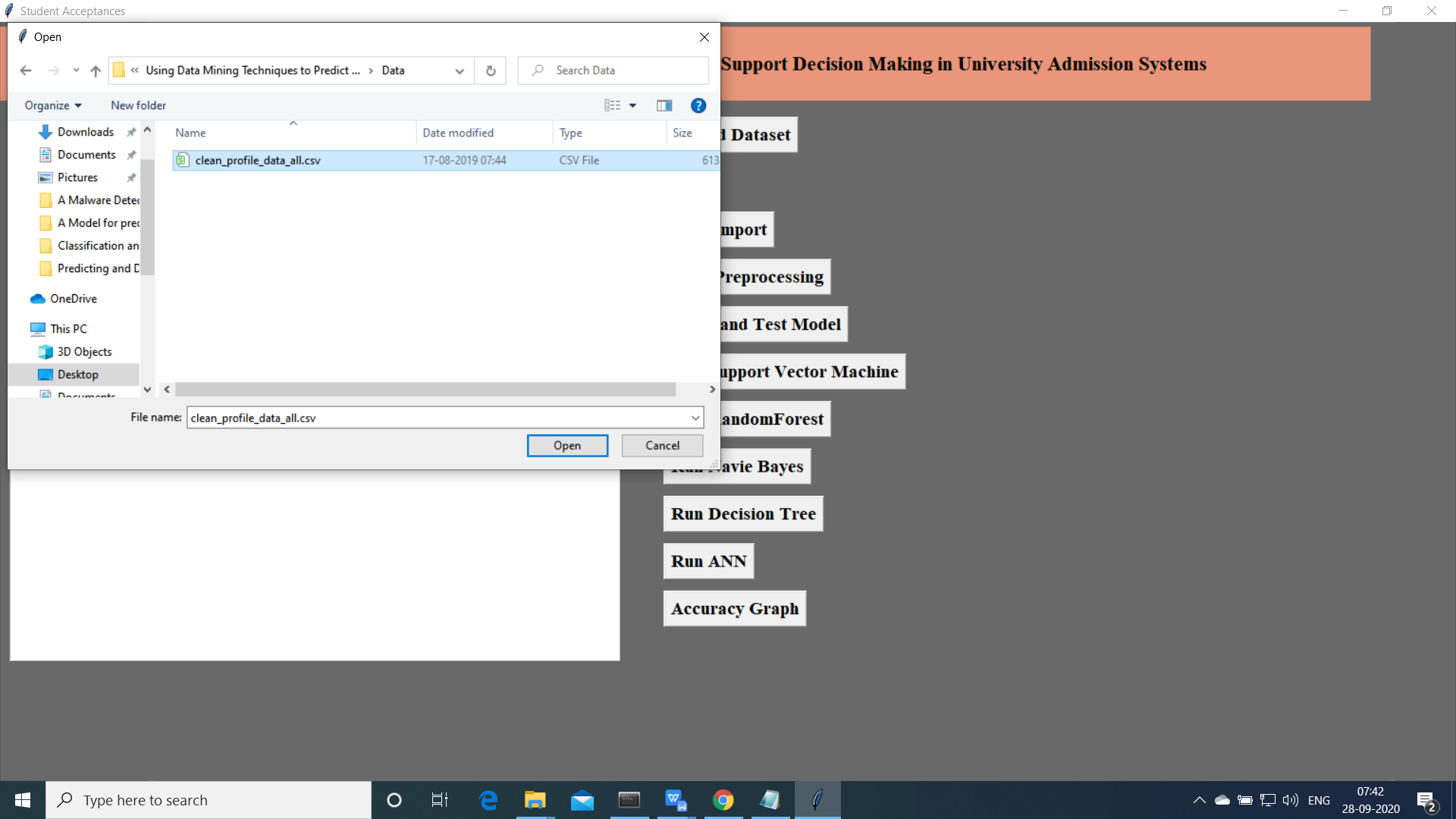
**Execution Steps:**

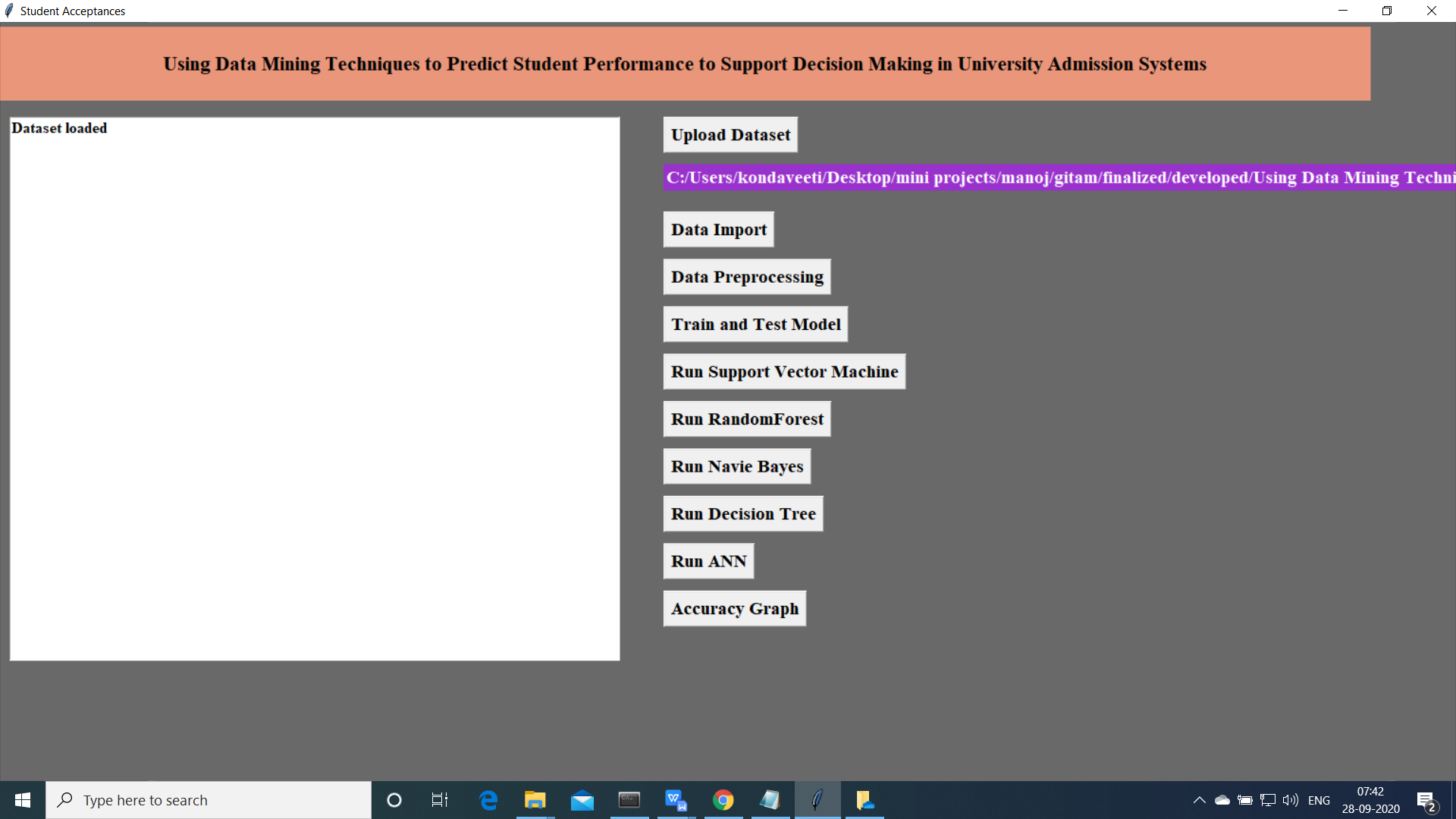
1. Open anaconda Prompt
2. Conda activate tf
3. Goto Project Directory
4. Python final.py



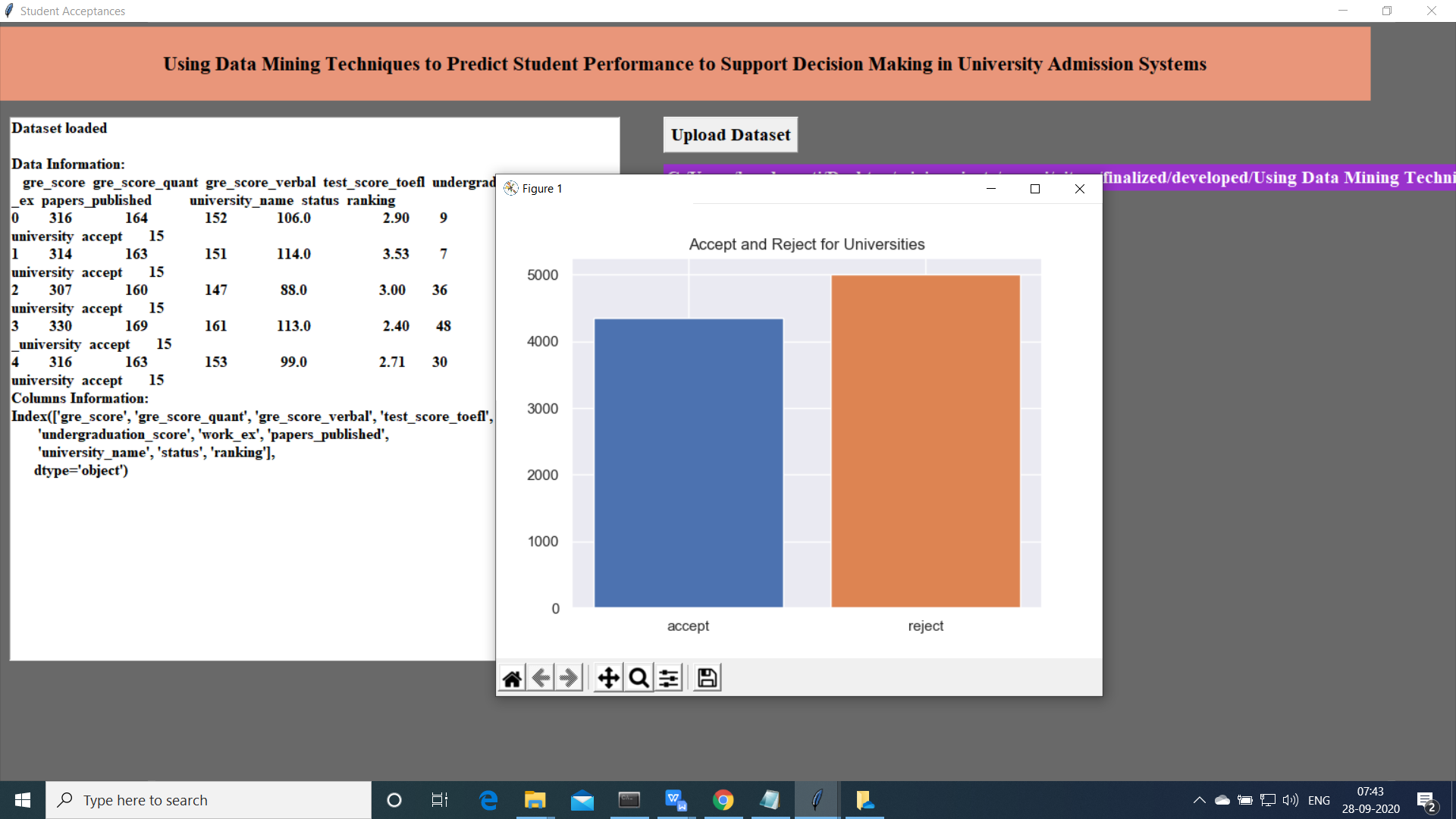
Above screen will be opened.

1. Now click on “Upload data ”



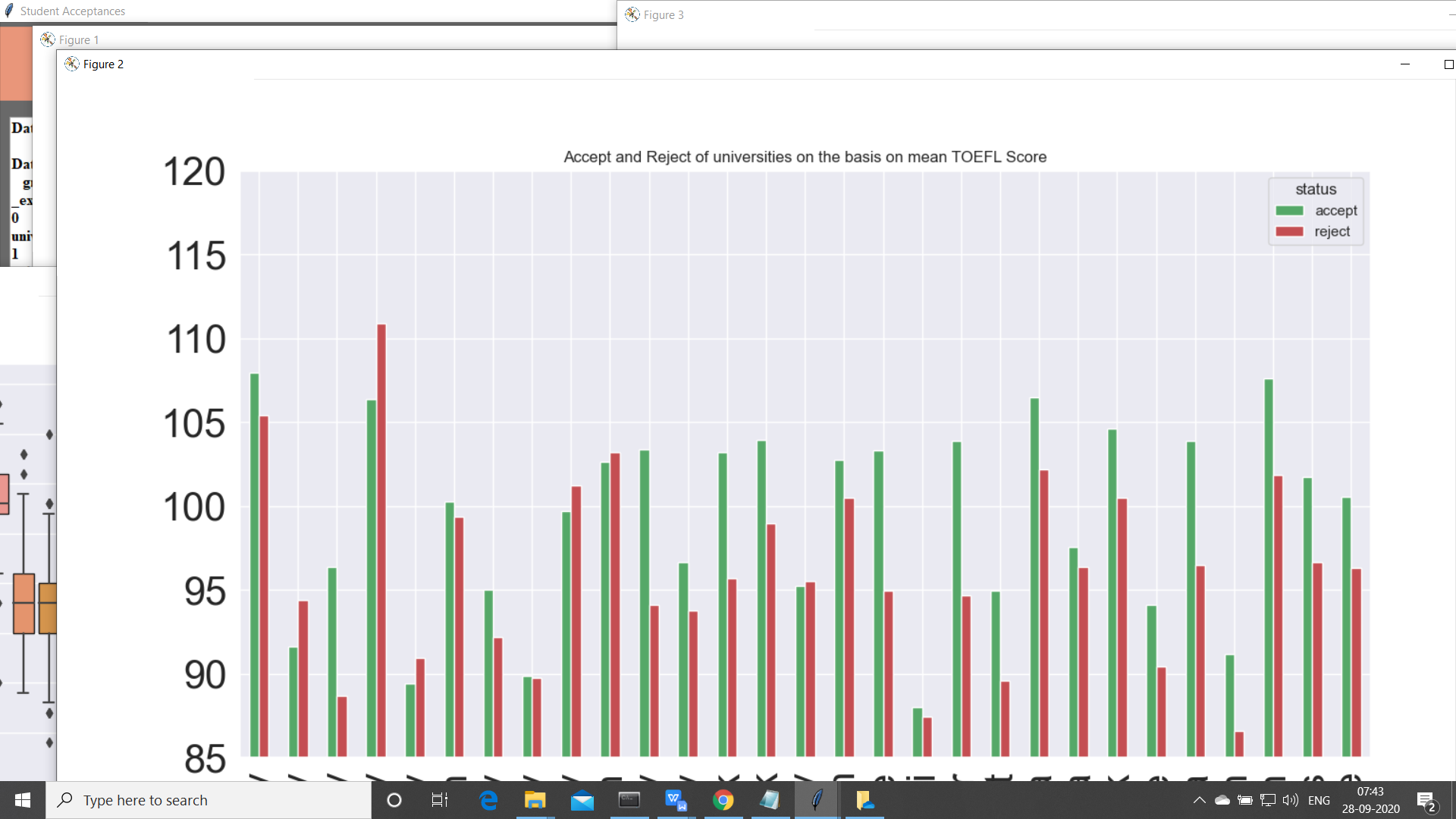


1. Import the data

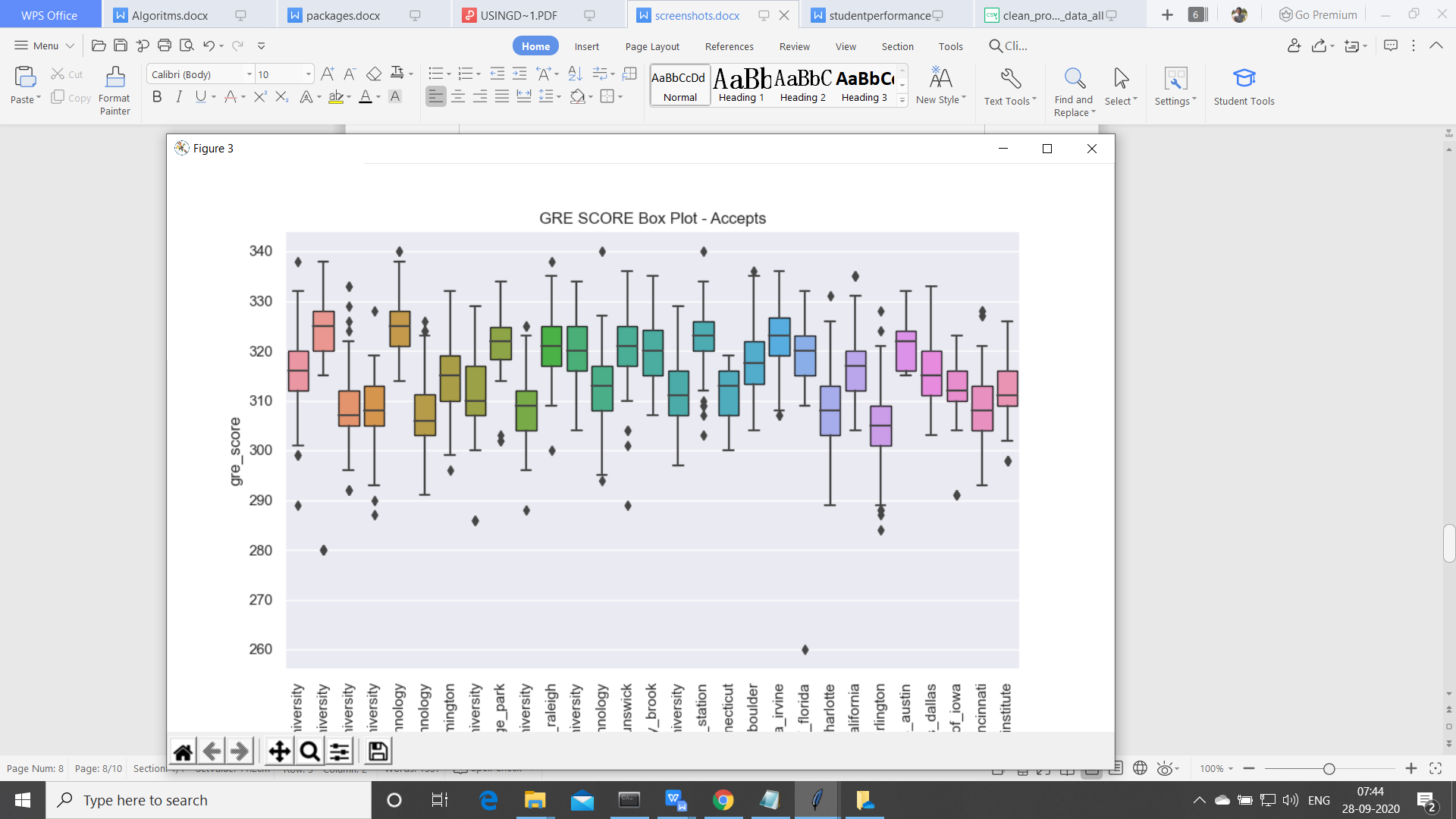


Upload the data and read the basic data information will be shown on the screen

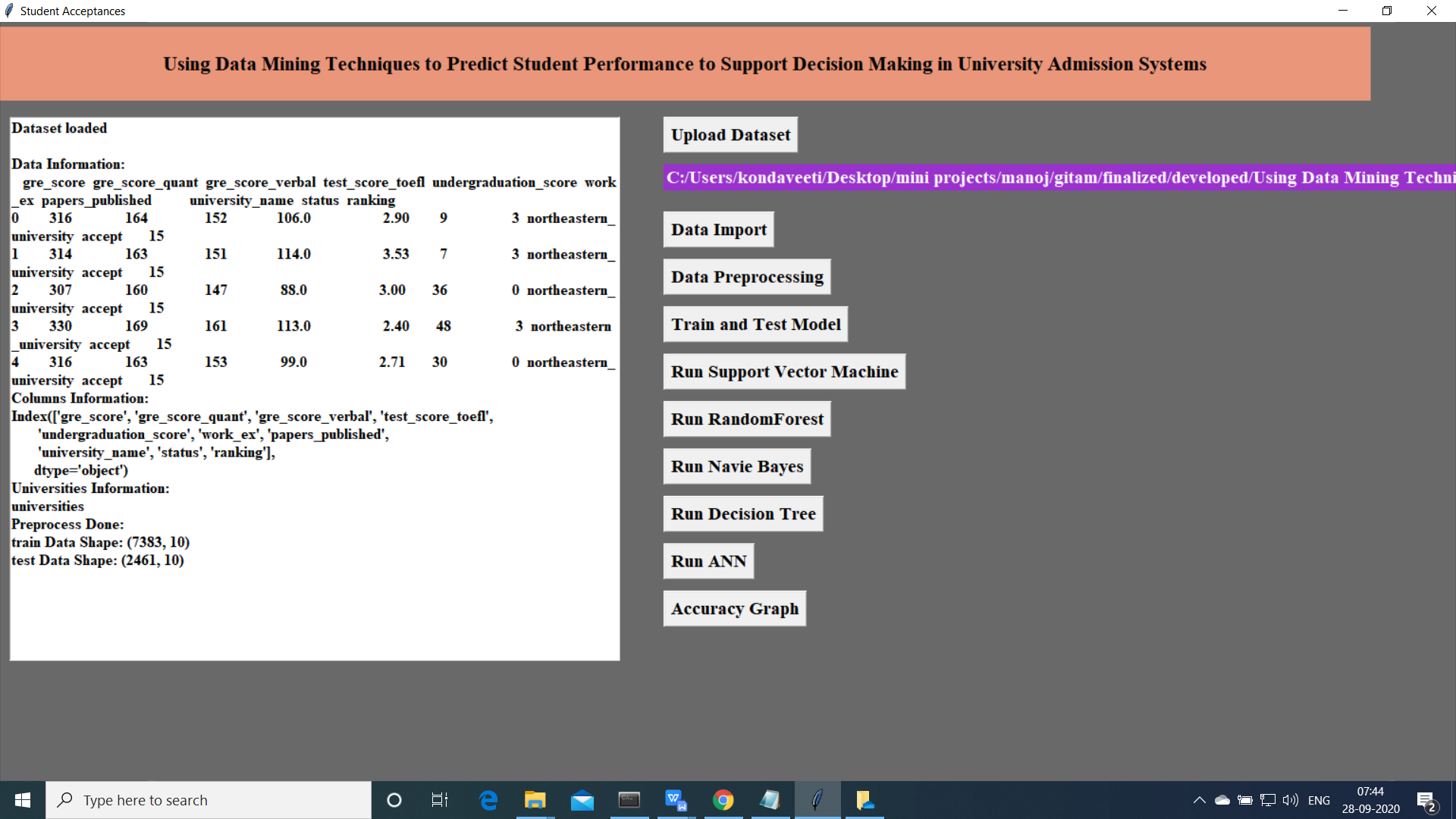
1. Data Preprocessing



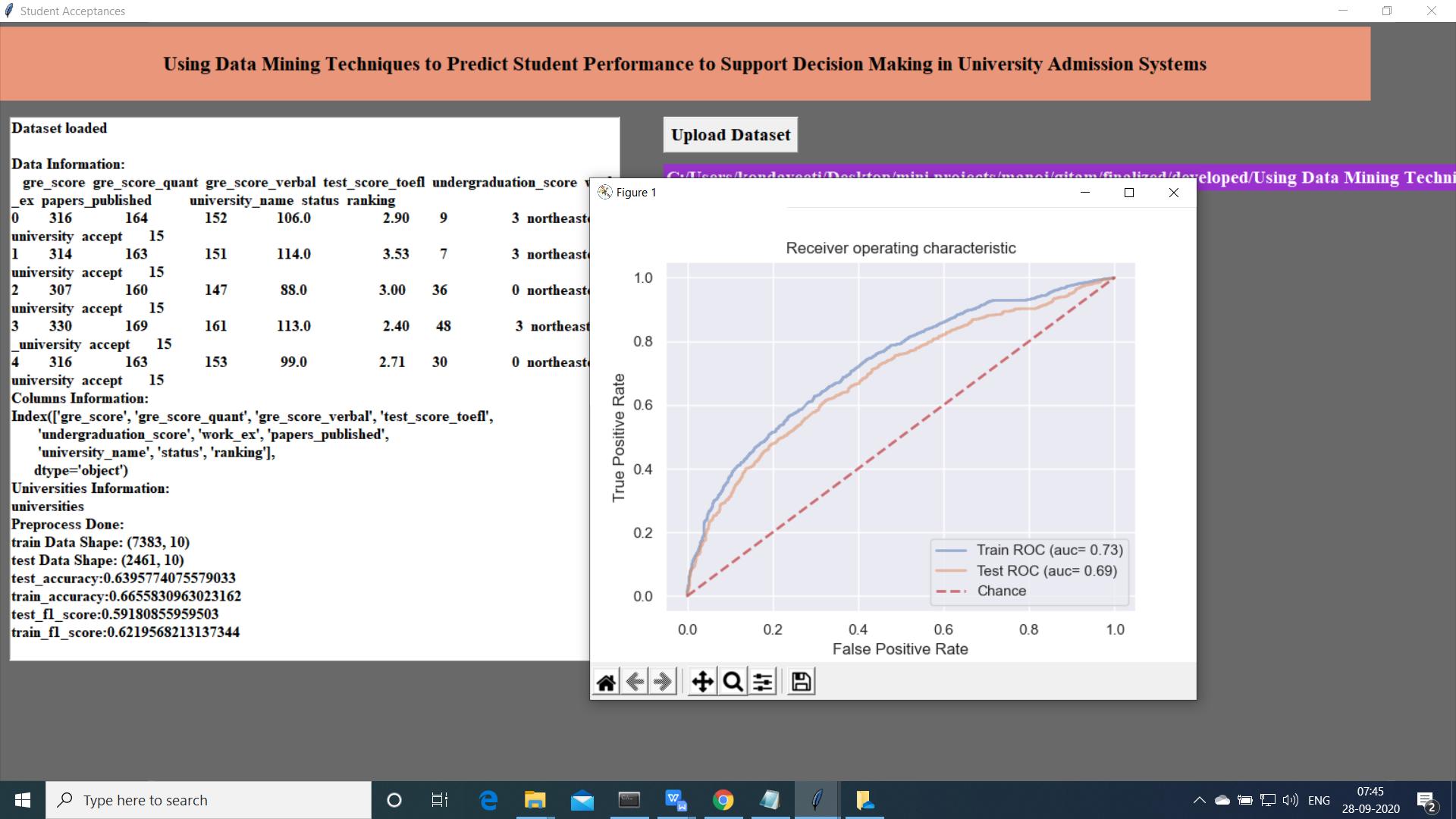


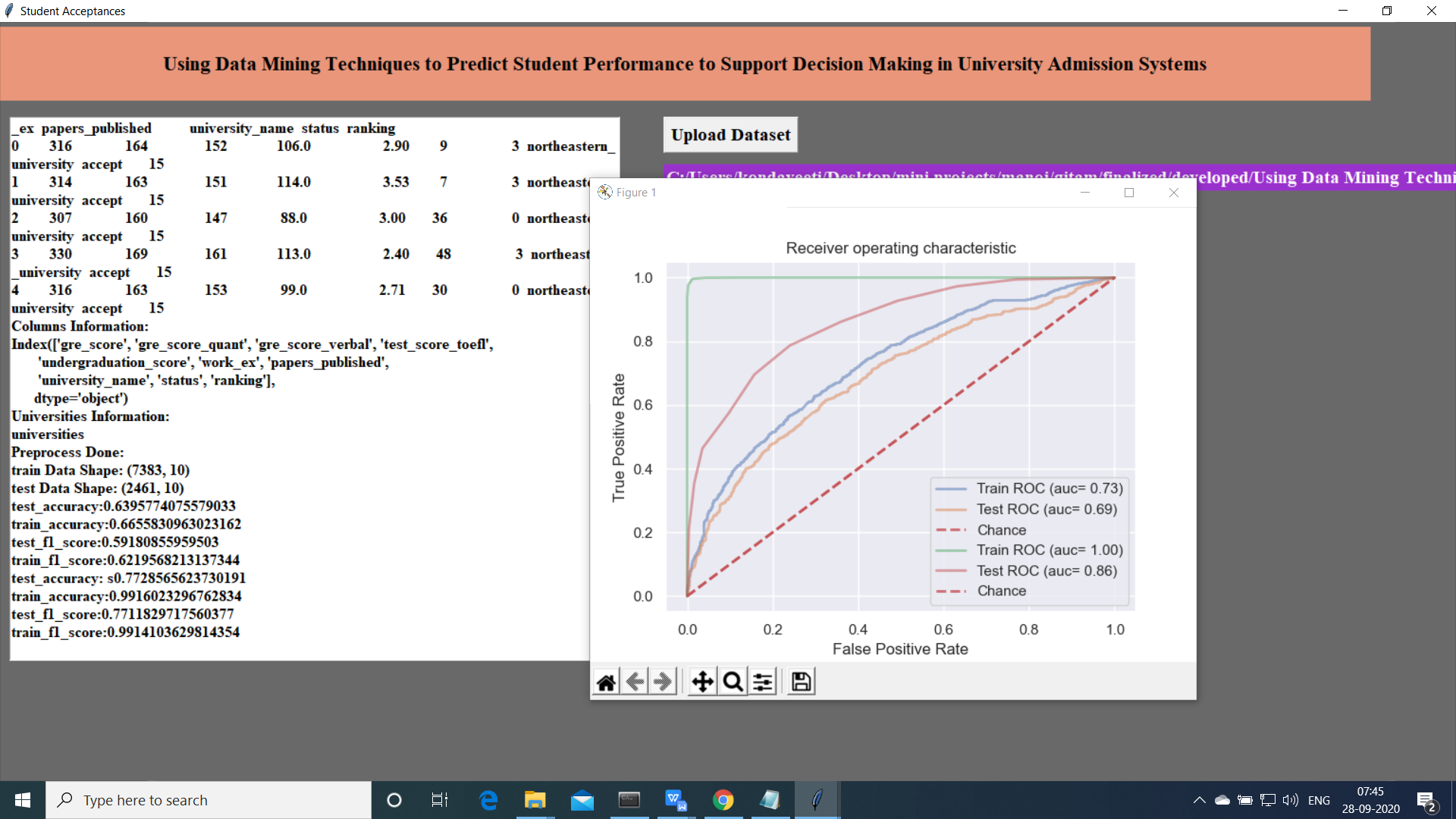


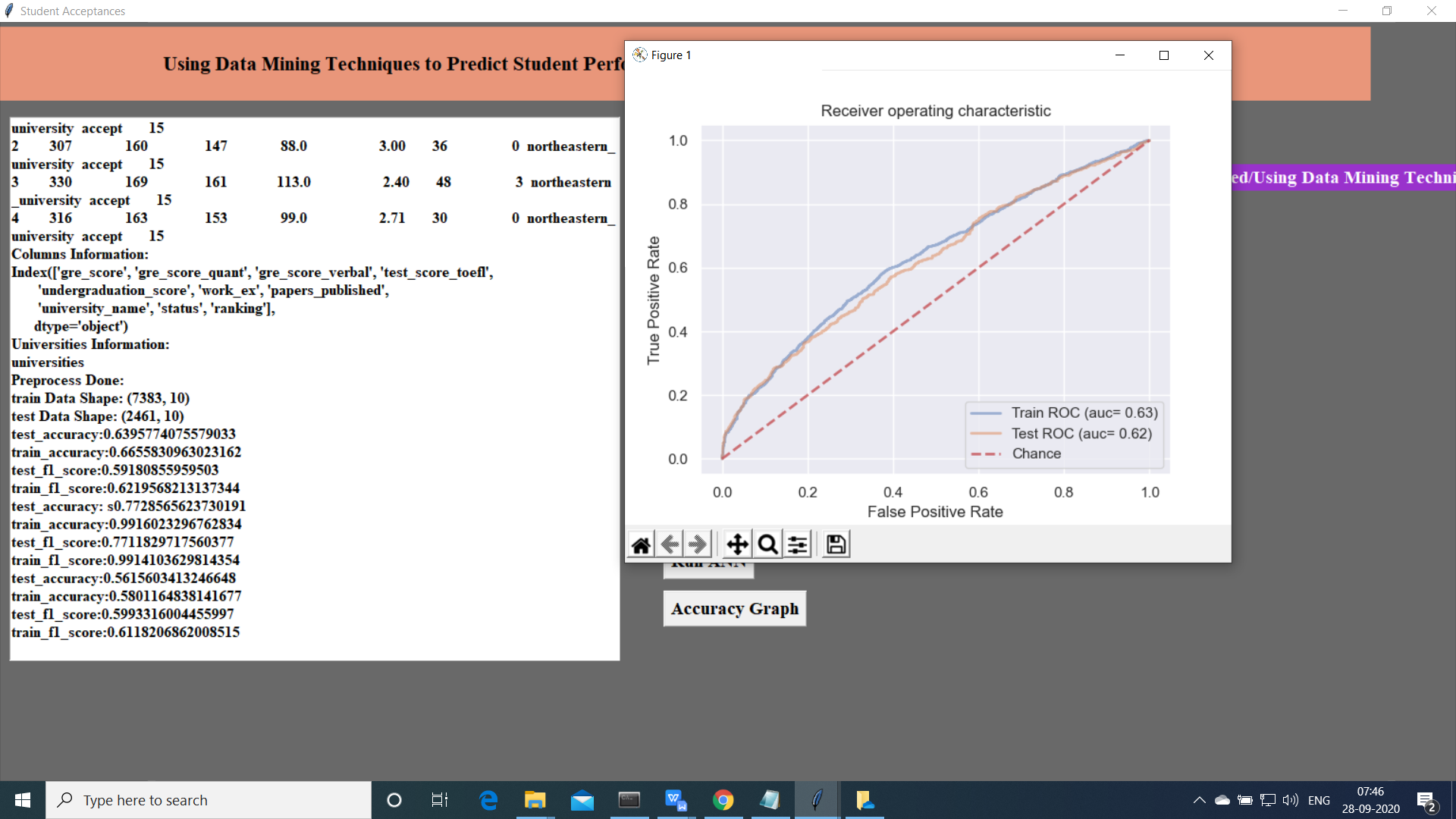
1. Now click on “Train and Test model”. split the data into train and test and traain will be used for training and to tets the performace we are using test data

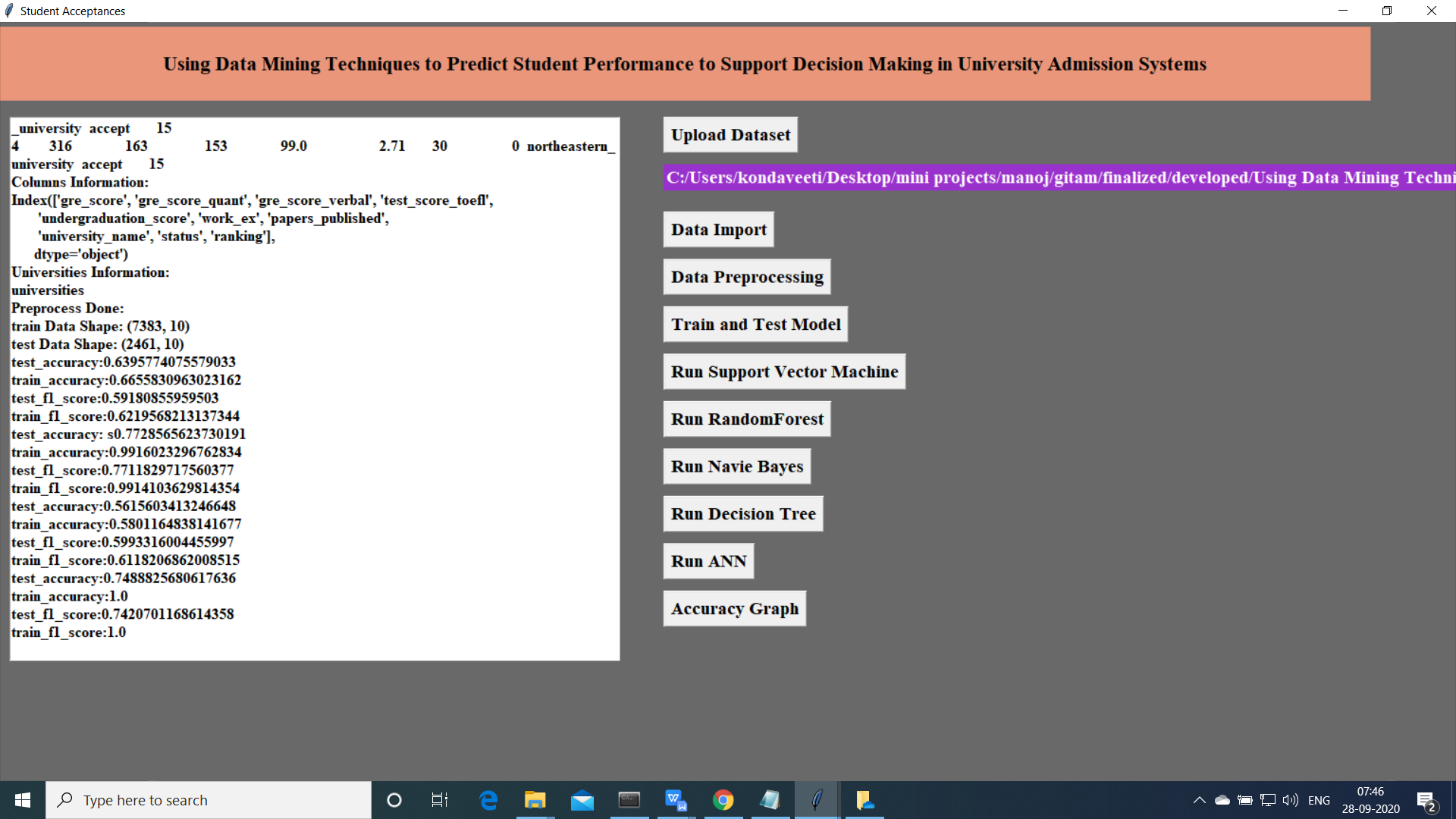


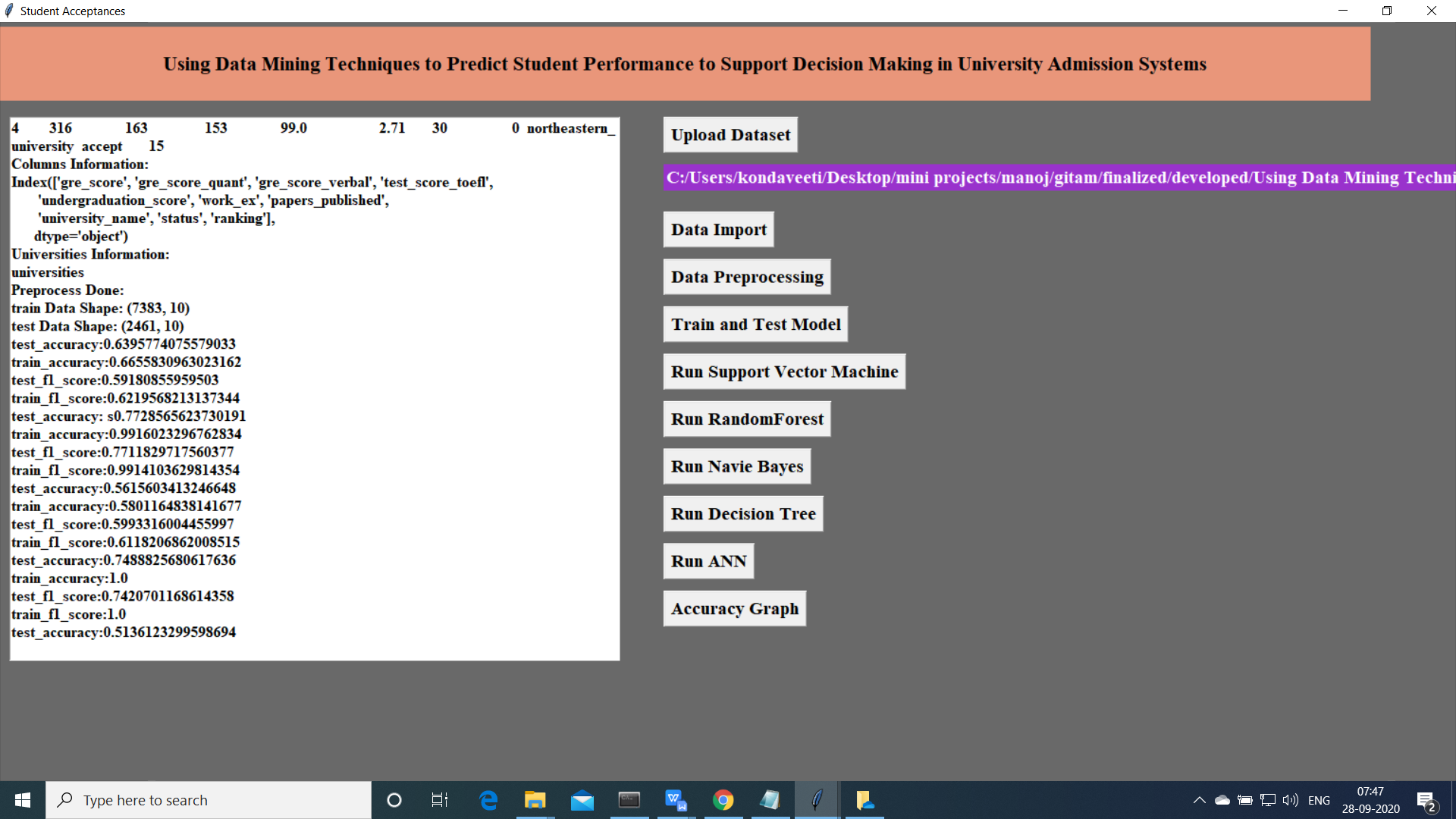
1. Now click on “Run Algoruimns”. Mentioned algorithms will be run on the data



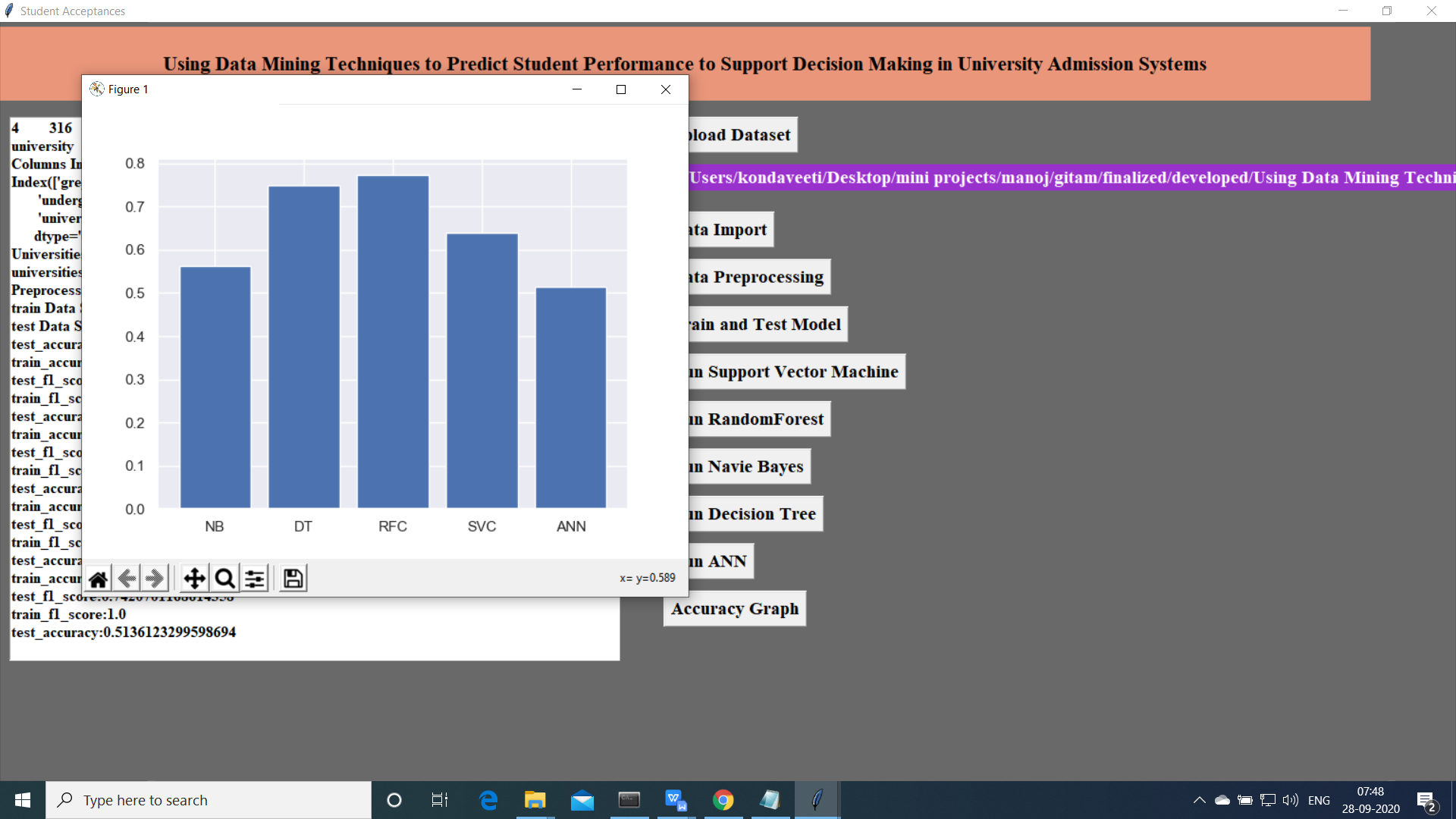








1. Accuracy Comparision for all the models



Extension is random forest acciracy is increased