A Threshold-Crossing Event (TCE) is a sequence of transit-like features in the flux time series of a given target that resembles the signature of a transiting planet to a sufficient degree that the target is passed on for further analysis. This is what we were supposed to train our machine learning model for i.e., making a statistical/machine learning model that should classify the data of threshold crossing events, with the parameters used for classification. Along with that, we were also supposed to make a web-based tool that inputs a CSV file from the user and outputs the predictions made by the model.

Starting off with our project, we created a machine learning model based on the Random Forest algorithm. Random Forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Moving on to the next task we created a web-based application for the user to input data. The application has been made using HTML, CSS and JavaScript. The website has been made interactive to enhance user experience.

Now comes the crucial part of the project i.e., integrating both the above-mentioned components. For that we have used Flask. Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gave us flexibility and is a more accessible framework. We combined the web-based application and the machine learning model together using Flask.

When the user inputs data in a CSV file, the Flask back-end sends the file to the machine learning model which performs the required computations and outputs the resultant CSV file and relevant graphs. The CSV file gets downloaded while the graphs are loaded in various HTML pages on the web browser.

**Accuracy:**

The accuracy of our model is as follows:

1. Predicting AFP: Around 70%
2. Predicting NTP: Around 89%
3. Predicting PC: Around 91%
4. Predicting UNK: Around 78%

**Limitations:**

We have not been able to increase the accuracy of the model beyond the above-mentioned values. This may be because of large data or the number of features. In addition to that, the deployed website does not show the plots (the plots are visible if you run it using local host).

**Scope of Improvements:**

1. We could further finetune the hyperparameters of our classifier model.
2. We can also try researching other machine learning algorithms that can handle large data sets better.
3. We must also develop a deeper insight on the physical significance of the features.
4. The deployed website should run faster and should also show up the plots.