**CSE-343 – Machine Learning Project Proposal**
**CELESTE - Exoplanet Detection using Machine Learning**


| Abhimanyu Bhatnagar | Apoorva Arya | Shashank Shekhar Singh | Tarushi Gandhi |
|---|---|---|---|
| 2020273 | 2020032 | 2020119 | 2020579 |
| abhimanyu20273@iiitd.ac.in | apoorva20032@iiitd.ac.in | shashank20119@iiitd.ac.in | tarushi20579@iiitd.ac.in |


## 1. Abstract


*Exoplanents, the planets that orbit other stars outside the solar system, have been of great interest to scientists across different fields. The study of these objects is believed to reveal answers pertaining to how life came to existence on earth, how the solar system evolved and whether we can find another habitable planet. Various traditional methods like gravitational microlensing, wobble method, direct imaging, etc., have been used but these require a huge amount of time and manpower and are also limited by the performance of astronomical telescopes. Thus, since the detection and study of exoplanets is valuable to help us, it is important to identify them first, which becomes significantly faster using classical Machine Learning techniques. We plan on using supervised and unsupervised methods for the same.*
**(GithubLink)**


## 2. Introduction

The main objective of the model is to identify (predict) the exoplanets from the Kepler Objects of Interest (KOI) which are classified as potential exoplanet candidates. This would help in reducing human labor that goes into identifying the exoplanets and getting the results much more efficiently

## 3. Literature Survey

IDENTIFYING EXOPLANETS WITH MACHINE LEARNING METHODS : In the paper, the authors propose the idea of using machine learning models to identify exoplanets and they try to achieve this by performing *three-categorical classification with false positive, candidate and confirmed as classes on kepler dataset for supervised learning and using k-means clustering on the confirmed exoplanet dataset for unsupervised learning*. By performing EDA they found out that the confirmed exoplanets follow a Gaussian distribution with respect to the *log scale orbital period* in

the range between -2 and 4 and that false positive samples are most likely to have just one or two planets in their solar system. For feature extraction they found that variables that store the deviation of observations are highly dependent on the observed values so they were removed. *Then 4 models, Decision Tree, Random Forest, Naive Bayes and Multilayer Perceptron were trained and achieved accuracies of 99.06%, 92.11%, 88.50%, and 99.79%, respectively and hyperparameters were found by experiment.* Then they evaluated the performance of these models with 10-fold cross-validation. As a result, the random forest model achieved the best average accuracy of 82.39%. They also suggested further investigation of the *receiver operating characteristics (ROC) and the precision-recall curves* to understand the diagnostic ability of different machine learning models.

Comparing Classification Models on Kepler Data : The *paper is centered around an experiment conducted to find the most optimal classification model, best able to classify the data into two categories: CANDIDATE or FALSE POSITIVE. According to the EDA conducted on the dataset by the author, two variables proved to be the most insightful - koi_depth and koi_duratio*n since they represent the transit attributes of the astronomical body. During feature selection, they dropped off 18 features to reduce the dimensionality of the data and eliminate redundancy. Following further preprocessing, three algorithms were chosen - Logistic Regression, Decision Trees and Artificial Neural Network. Best parameters for these models were found using GridSearch with k-fold cross validation (k=5) and corresponding ROC and precision-recall curves were also plotted in order to get the best accuracy and to get the best ratio of True Positives and False Positives. The models were compared on three factors: Execution Time, Statistical Significance Test and Prediction Performance. Based on the prediction performance, Neural Network gives the best accuracy but surprisingly performs relatively poorly to Logistic

Regression and Decision Trees with respect to precision and recall. Since there's need for less False Positives, comparing between Logistic Regression and Decision Trees, Decision Trees (max-depth=6) out-perform Logistic Regression in terms of accuracy and relevancy.

## 4. Dataset

### 4.1 Dataset description

The Kepler space observatory was launched in 2009 by NASA. In order to find exoplanets the observatory must discover other solar systems. The observatory does this by identifying stars that have signatures that indicate transiting planets. These stars are called Kepler's objects of interest (KOI). The data we are using is a record of all the Kepler objects of interest that the Kepler space observatory has been monitoring.
The distributions for CONFIRMED(0) and CANDIDATE (1) KOIs are pretty similar, whereas the FALSE POSITIVE (2) distributions seem to have more variation and higher medians.

- The distributions for CONFIRMED(0) and CANDIDATE (1) KOIs are pretty similar, whereas the FALSE POSITIVE (2) distributions seem to have more variation and higher medians.
- There is significant separability between the FALSE POSITIVE labels and the other labels. This tells us that the data will work well when classifying FALSE POSITIVE labels with different classification models.
- The above observation also tells us that there will be difficulty when trying to distinguish between CONFIRMED and CANDIDATE labels due to their points overlapping.Also, the CANDIDATE observations are not very useful to us since they are unable to give a binary answer to our problem statement - is this KOI an exoplanet or not?
- Thus, it is safer to use the status CONFIRMED in koi_disposition column to be the accurate response of exoplanet identification.
- CONFIRMED exoplanets also generally have much less measurement error value and less outliers data points than both FALSE POSITIVE and CANDIDATE counterparts

Features of interest include:

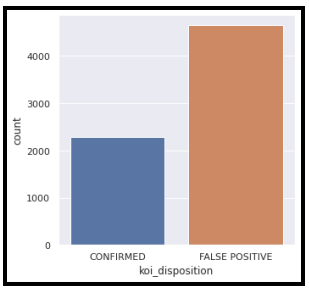
- koi_disposition: Category of KOI from Exoplanet Archive taken. Designations are taken from p_disposition.
- koi_pdisposition: Disposition using Kepler Data. Ongoing designation of KOI that describe planets with the most probable physical explanation. All data could be changed over time.
- koi_period: Orbital period(days), the interval between consecutive planetary transits.
- koi_duration: Transit Duration (hours), the duration of the observed transits. Duration is measured from first contact between the planet and star until last contact.
- koi_depth: Transit Depth (parts per million), the fraction of stellar flux lost at the minimum of the planetary transit.
- koi_fpflag_nt: A KOI whose light curve is not consistent with that of a transiting planet. This includes, but is not limited to, instrumental artifacts, non-eclipsing variable stars, and spurious (very low SNR) detections.
- koi_fpflag_ss: A KOI that is observed to have transit-like events is most likely caused by eclipsing binary and other astronomical events.
- koi_fpflag_co: The source of signal is from a nearby star.
- koi_fpflag_ec: The KOI shares the same period and epoch as another object that is judged to be the result of flux contamination in the aperture or electronic crosstalk.

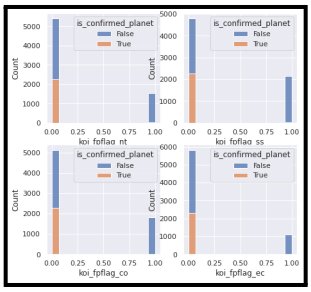
### 4.2 Data Preprocessing

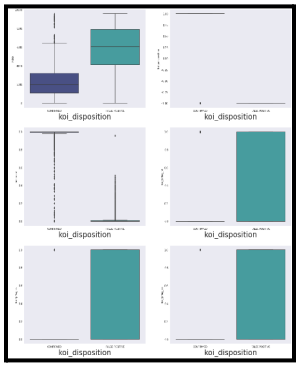
Missing values : 2 features, namely koi_teq_err1 and koi_teq_err2, were dropped because they had more than 1000 missing values. Samples that had greater than 10 missing feature values were also removed. The rest of the missing feature values were filled with the mean of that particular feature.
Removing Candidate exoplanet samples:
The label column had values: Candidate exoplanet, confirmed exoplanet and confirmed non-exoplanet. Since we wished to classify the Kepler objects of interest(KOI) as exoplanets or not, classifying them as exoplanet candidates would have defeated our purpose. So all the Candidate exoplanet samples were dropped.
Standardization: The data was standardized using the formula $X = (x - \mu)/\sigma$
Where $\mu$ is the mean and $\sigma$ the standard deviation.
Highly correlated features: If two features had a correlation value of greater than 0.95 then one of those 2

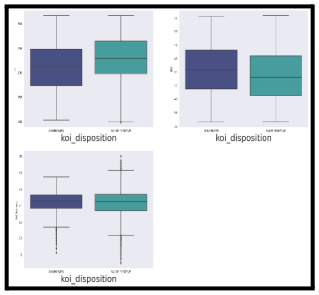
features was dropped. 7 columns were dropped to the high correlation observed amongst the features. Features dropped: koi_depth_err2, koi_duration_err1, koi_insol_err,koi_period_err1, koi_prad_err2 , koi_score, koi_time0bk_err2

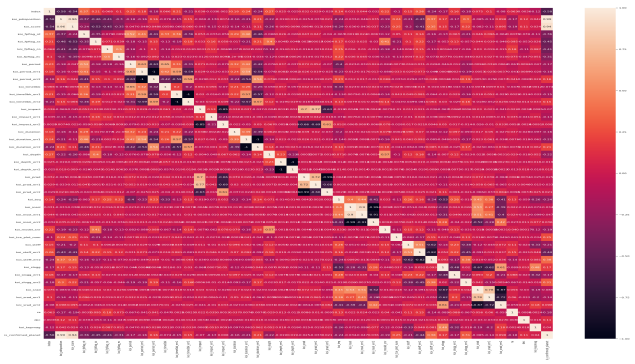
### 4.3 Data Visualization



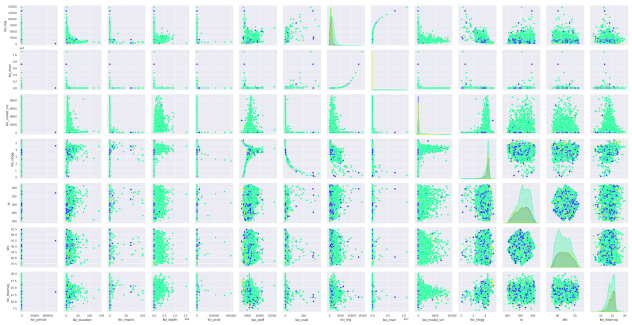



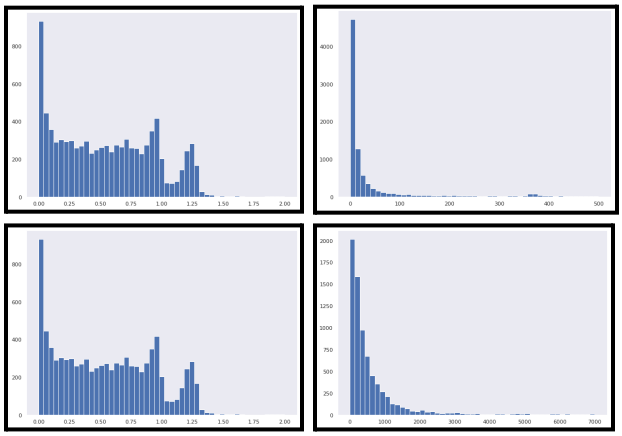



Heatmap displaying correlations

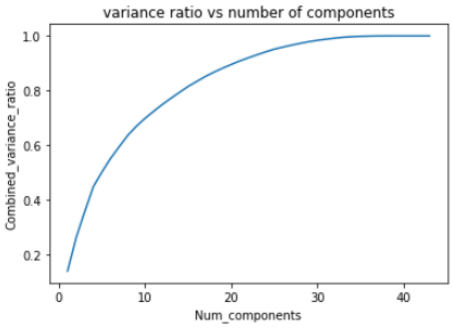


Pairplots



## 5. Methodology, Model details

- Data cleaning: This was quite an important step since there were a lot of missing values in our data. 6939 samples were left after data cleaning.
- Data visualization: EDA was performed in order to gain insight on the data using seaborn and matplotlib.
- Data Standardization: The data was standardized using StandardScaler of Sklearn
- Feature extraction : Principal component analysis was used for feature extraction and 90 percent variance was retained. The data had 20 features after feature extraction.



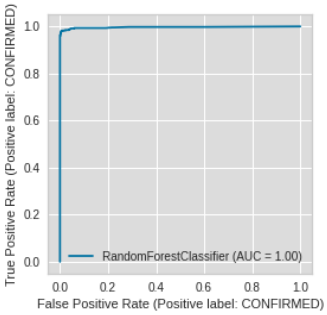

- Train Test split : 80:20 stratified train test split was made. This was done in order to ensure that we had sufficient training data and testing data. Also this ensured that the ratio of the classes in both train and test was nearly the same.
- We built the following models : Random forest, Multilayer Perceptron, Logistic regression,

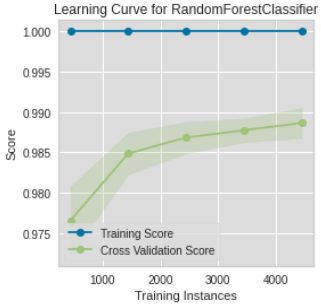
adaboost, decision tree, Naive bayes, Support vector machine classifier, XGBoost, semi-supervised clustering using logistic regression

- Cross-validated grid search: For all the models built we used grid search with cross validation to find the best parameters. For the grid search the scoring for the model was done on the basis of accuracy.
- For all the models built calculated accuracy, precision, recall, F1 score, AUROC and also plotted the AUROC curves.
- Parameters that we grid searched for each model are
- Random forest - Criterion: [gini ,entropy] , max_depth: [6, 8, 10, 12, 14, 16, 18], n_estimators: 80, 100, 120, 140]
- Logistic regression - Penalty:[l1,l2], solver: [newton-cg, lbfgs, liblinear], C: [0.001, 0.01 ,0.1 ,1 ,10 ,100 ,1000]
- Multiplayer Perceptron - Activation: [identity, logistic, tanh, relu], Solver: [lbfgs, sgd, adam] , max_iter:[300,500]
- Decision tree - Criterion: [gini ,entropy] , max_depth: [2, 4, 6, 8, 10, 12],
- Adaboost - Base estimator : decision tree with max depth:[1,2 ,3], n_estimators : [40,50,60,70]
- Support Vector Machine - C : [0.001, 0.01, 0.1, 1,10,100,1000] , kernel: [linear, rbf, sigmoid , poly]. Only for kernel poly : degree : [2,3,4,5]
- XGBoost : reg_lambda: [0.001, 0.01, 0.1, 1,10,100,1000] , max_depth:[6,8,10,12,14,16,18], n_estimators:[80,100,120,140]

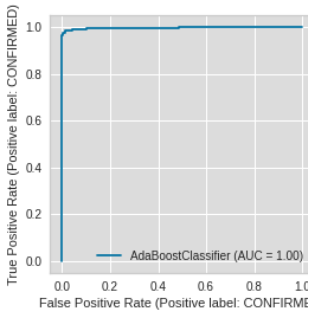
## 6. Results and analysis

### Random Forest:



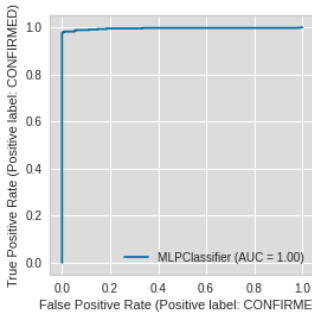



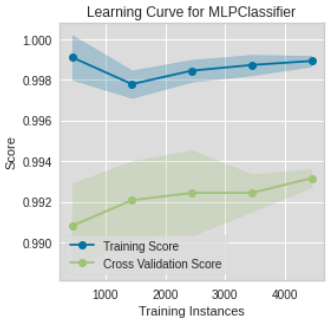

### AdaBoost:



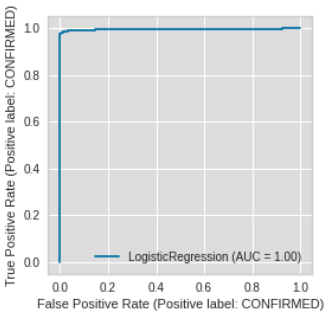


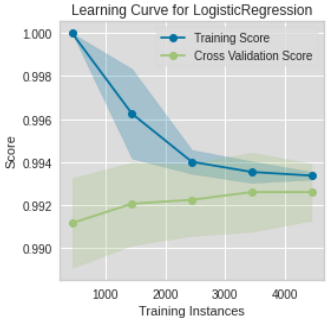
### MLP:



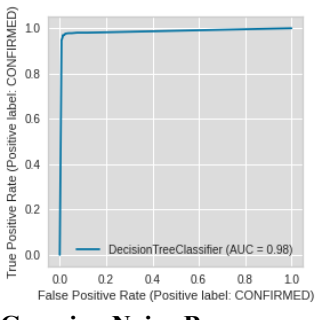



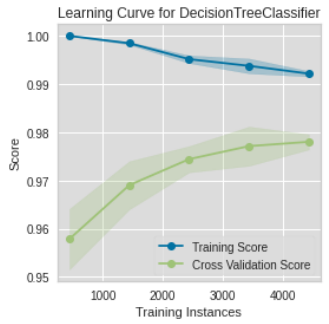

### Logistic Regression:



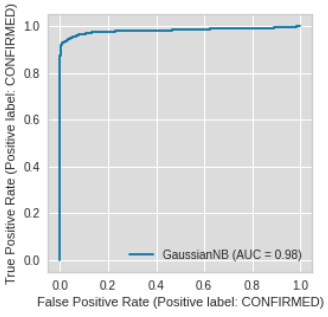



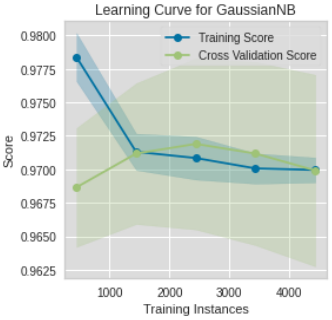

### Decision Tree:



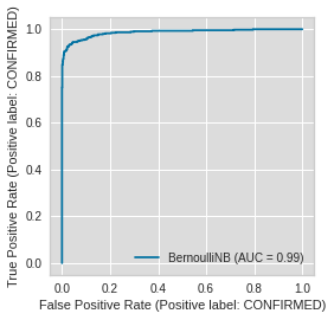



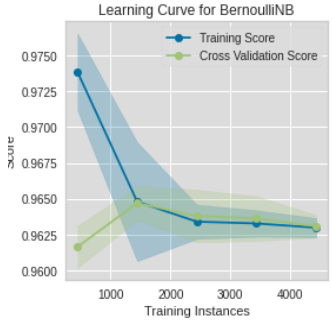

### Gaussian Naive Bayes:



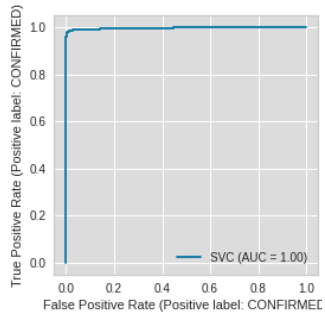



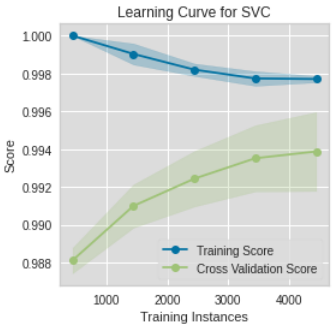
**Bernoulli Naive Bayes :**



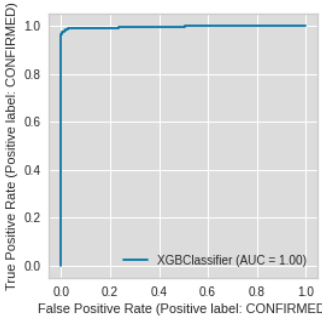



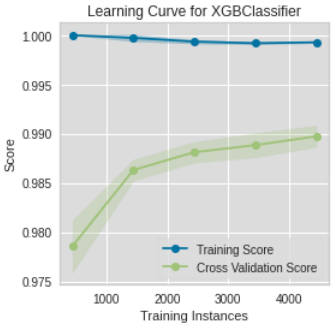

**SVM:**



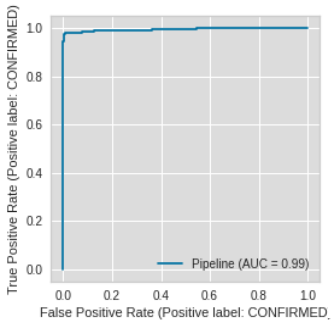



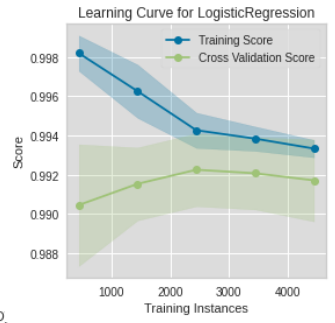

**XGBoost:**




**K-Means clustering:**




| Model | Accuracy | Precision | Recall | F1-score | AUROC |
|---|---|---|---|---|---|
| Gaussian NB | 0.9640 | 0.9574 | 0.9903 | 0.9735 | 0.9766 |
| Bernoulli NB | 0.9668 | 0.9623 | 0.9892 | 0.9756 | 0.9886 |
| Logistic Regression | **0.9921** | 0.9883 | **1.0** | **0.9941** | 0.9963 |
| Decision Trees | 0.9762 | 0.9807 | 0.9839 | 0.9823 | 0.9799 |
| Random forest | 0.9891 | 0.9841 | **1.0** | 0.9920 | 0.9966 |
| AdaBoost | 0.9899 | 0.9872 | 0.9978 | 0.9925 | 0.9955 |
| XGBoost | 0.9870 | 0.9830 | 0.9978 | 0.9903 | 0.9972 |
| MLP Classifier | 0.9914 | 0.9882 | 0.9989 | 0.9936 | 0.9968 |
| SVM Classifier | 0.9906 | **0.9914** | 0.9946 | 0.9930 | **0.9974** |
| Clustering* | 0.9870 | 0.9820 | 0.9989 | 0.9904 | 0.9948 |

* clustering was semi-supervised and used Logistic Regression

**Accuracy:** Logistic Regression > Multi-Layer Perceptron > SVM Classifier > AdaBoost > Random Forest > Clustering > XgBoost > Decision Tree > Bernoulli Naive Bayes > Gaussian Naive Bayes

**Precision:** SVM Classifier > Logistic Regression > Multi-Layer Perceptron > AdaBoost > Random Forest > XgBoost > Clustering > Decision Tree > Bernoulli Naive Bayes > Gaussian Naive Bayes

**Recall :** Random Forest = Logistic Regression > Multi-Layer Perceptron > Clustering > AdaBoost > XgBoost > SVM Classifier Gaussian Naive Bayes > Bernoulli Naive Bayes > Decision Tree.

**F1 Score :** Logistic Regression > Multi-Layer Perceptron > SVM Classifier > AdaBoost > Random Forest > Clustering > XgBoost > Decision Tree > Gaussian Naive Bayes > Bernoulli Naive Bayes.

**AUROC :** SVM Classifier > XgBoost > Multi-Layer Perceptron > Random Forest > Logistic Regression > AdaBoost > Clusterin > Bernoulli Naive Bayes > Decision Tree > Gaussian Naive Bayes

## 7. Conclusion

Logistic Regression, Multi-Layer Perceptron, AdaBoost, SVM Classifier had almost similar performance matrices. It was observed that the Logistic regression classifier was the best based on F1-score. F1-score is the most important metric in our case. This is because in our problem the dataset is imbalanced and we assign an equal weightage to a wrong prediction for both confirmed and false positive classes.

### 7.1 Learning

- Exploring different techniques to visualize data.
- Cleaning datasets to improve our model.
- Evaluating and comparing different models.
- Training and testing other ML models like SVM, XGBoost, semi-supervised clustering etc.

### 7.3 Each members contribution

- Preprocessing Data - Shashank, Abhimanyu
- Data Visualization - Apoorva, Tarushi
- Data Analysis - Shashank, Apoorva
- Feature Extraction - Abhimanyu, Tarushi
- Development of models - All
- Report and Slides - All

## 8. References

[1] Yucheng Jin, Lanyi Yang, and Chia-En Chiang. Identifying
Exoplanets with Machine Learning Methods: A Preliminary
Study. International Journal on Cybernetics & Informatics, 11(2):31–42, April 2022.
[2] Abhishek Malik, Benjamin P. Moster, and Christian Obermeier. Exoplanet Detection using Machine Learning. Monthly Notices of the Royal Astronomical Society, page stab3692, December 2021.
[3] George Clayton Sturrock, Brychan Manry, and Sohail Rafiqi. Machine Learning Pipeline for Exoplanet Classification.2(1):29,2019.