e-ISSN 2805-4356
p-ISSN 2805-4364

Check for updates

10.5281/zenodo.10566250 Vol. 07 Issue 01 Jan - 2024 Manuscript ID: #01201

IDENTIFYING IMPORTANT FEATURES FOR EXOPLANET DETECTION: A MACHINE LEARNING APPROACH

Abdul Karim[1*], Jamal Uddin[1] and Md. Mahmudul Hasan Riyad[1]

[1]Department of Applied Mathematics, Noakhali Science and Technology University, Noakhali-3814, Bangladesh.

***Corresponding author:** akarim.amth@nstu.edu.bd

Abstract

The study and discovery of exoplanets (planets outside the solar system) have been a major focus in astronomy. Many efforts have been made to discover exoplanets using ground based and space based observatory, NASA's Exoplanet Exploration Program being one of them. It has developed modern satellites like Kepler which are capable of collecting large array of data to help researchers with these objects. With the increasing number of exoplanet candidates, identifying and verifying their existence becomes a challenging task. In this research, we propose a statistical and machine learning approach to identify important features for exoplanet identification. For this purpose, we use the Kepler Cumulative Object of Interest (KCOI) dataset. After pre-processing the data we utilize statistical methods namely ANOVA F-test, Mutual Information Gain (MIG), Recursive Feature Elimination (RFE) to select the most significant features and have trained 10 state-of-the-art classifiers on them recursively to identify the features that leads to best performance. According to the results of our investigation, classifiers trained on features chosen by Recursive Feature Elimination with Random Forest as estimator produces superior results, with CatBoost classifier being the best with an accuracy of 99.61%. Our findings demonstrate the potential of machine learning in helping astronomers to efficiently and accurately verify exoplanet candidates in large astronomical datasets.

Keywords

Exoplanet, Machin Learning, KCOI, Important Features.

This work is licensed under Creative Commons Attribution 4.0 License.

Copyright © The Author(s). All Rights Reserved © GLOBAL PUBLICATION HOUSE | International Journal of Applied Science |

1 INTRODUCTION

One of the most ancient natural sciences in human history is astronomy. For centuries, people have gazed up at the night sky to see the dazzling stars. The observable universe contains hundreds of billions of galaxies each of which contains billons of stars and these stars also have their own planetary system. More intriguing questions emerged as our understanding of the cosmos increased. Exoplanets were discovered as a result of people's interest about whether planets similar to our own existed in other star systems.

Exoplanet discovery is a tedious and time-consuming process that typically involves a team of professionals who devote their lives to it. Usually, they use data collected from ground based observatories and satellite-based telescopes together with their expertise, intellect and perseverance to hunt for exoplanets. But with the launch of specialized satellites like Kepler to discover exoplanets this process became much simpler. These satellites capture and process images and generate usable data for the scientists to interrogate them with little to no processing needed.

There has been a lot of work done to identify exoplanets using machine learning but not much has been done to identify the important features which leads to the identification. In this study, we explore the publicly available Kepler cumulative object of interest (KCOI)[1] dataset. This dataset contains information collected by the Kepler satellite and stored into several fields. We use this data to identify the important features to improve the accuracy of identifying exoplanets using statistical methods and machine learning.

The detection and characterization of exoplanets is a rapid growing field of research, and the use of machine learning techniques becomes increasingly popular in recent years. In the context of exoplanet detection, machine learning is being used to analyze large datasets from telescopes and identify patterns that indicate the presence of exoplanets.

Shallue and Andrew Vanderburg proposed a convolutional neural network to train on the light curve and predicted two new exoplanets according to their model predictions[2]. Yucheng et al. used several machine learning algorithms on NASA's Kepler dataset to identify exoplanet and produced 99.79% accuracy using multilayer perceptron[3]. Also, Malik et al. used gradient boosting classifier on the Kepler dataset and achieved an accuracy of 98%[4]. In another work, Manry B et al. implemented their machine learning pipeline to process the KCOI dataset to identify the exoplanet and obtained 99% accuracy using random forest[5]. A supervised learning approach was undertaken by Bugueno et al. extracting features from light curve data from Kepler to discover exoplanets[6]. A study was done to analyze various machine learning based exoplanet transit finding strategies by Jara-Maldonado M et al[7].

2 BACKGROUND

In this section we introduce the concepts needed to develop an understanding to better understand the tools and techniques used in the study.

2.1 NASA's Exoplanet Mission

NASA's efforts to discover and study exoplanets have a long and interesting history. The first exoplanet was discovered in 1992 by astronomers using ground-based telescopes. However, it wasn't until the NASA's launch ofKepler spacecraft in 2009 that the search for exoplanets really took off. Kepler used the transit method to detect exoplanets using the periodic dimming of starlight as a planet

passed in front of its star. Over its nine-year mission, Kepler discovered thousands of exoplanets and confirmed the existence of hundreds more.

In 2018, Kepler was retired, but its legacylives on through the ongoing analysis of its data. At the same time, NASA launched the Transiting Exoplanet Survey Satellite (TESS), which continued the search for exoplanets using the transit method. TESS has already discovered numerous new exoplanets and is expected to find many more in the coming years.

2.2 Kepler Space Telescope

Kepler was a space observatory launched by NASA in March 2009 with the mission to search for exoplanets. It has been used the transit method, which observes a star's brightness over time to detect periodic dips that indicate the presence of an orbiting planet. Kepler was positioned to look at a specific patch of sky for over four years, studying more than 530,506 stars and detecting over 2,600 confirmed exoplanets, many of which are similar in size and temperature like Earth.

The Kepler mission has revolutionized our understanding of exoplanets and the prevalence of potentially habitable planets in our galaxy. The data collected by Kepler has been used to study exoplanet demographics, atmospheric properties, and the potential for life on other planets. In addition to its scientific discoveries, Kepler has inspired new generations of exoplanet hunters and opened new avenues of research in astronomy. Although the Kepler mission officially ended in 2018 due to hardware failure, its legacy continues to inform and inspire new discoveries in the field of exoplanetary science.

2.3 Exoplanet Identification Techniques

In order to support the traditional exoplanet detection methods of transit time, radial velocity, microlensing, and direct imaging, the Kepler mission observed and archived data. The radial velocity method is a technique used to detect exoplanets by measuring the gravitational pull that a planet exerts on its parent star. This causes the star to move slightly in its orbit, causing a periodic change in the star's radial velocity. By observing these changes and measuring the Doppler shift of the star's spectral lines, astronomers can determine the presence of a planet, its minimum mass, orbital period, and distance from the star. The radial velocity method is one of the most successful techniques for exoplanet detection and has been used to discover thousands of exoplanets to date.

The transit method is a relatively inexpensive and straightforward way to detect exoplanets and is particularly well suited for finding small, Earth-sized planets. The method is also useful for characterizing exoplanetary atmospheres, as the dimming can be analyzed to determine the composition and atmospheric properties of the planet. The transit method has been extremely successful in finding exoplanets and has been used to discover thousands of exoplanets to date, including some of the closest and most habitable exoplanets known.

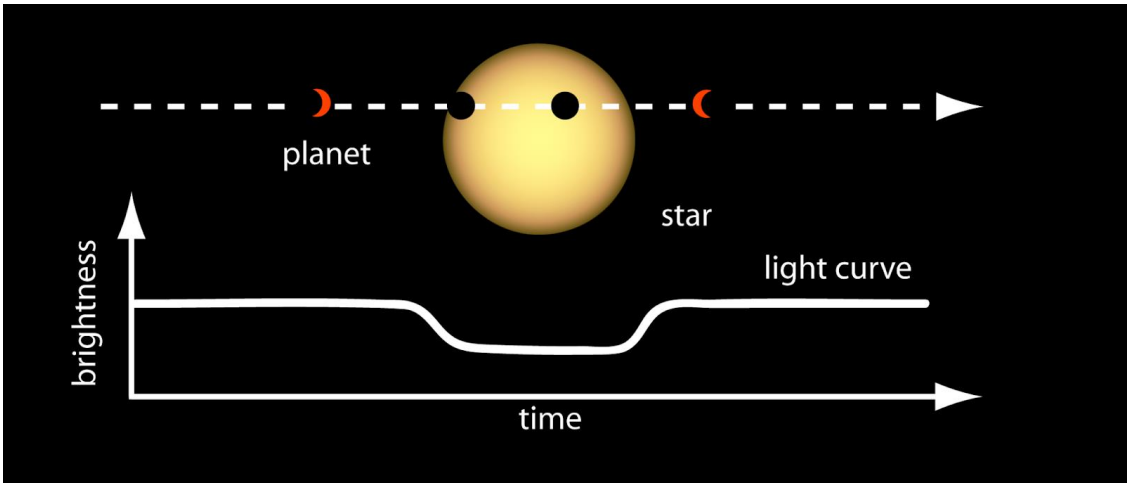


Figure 2.1: Transit Method [8]

The direct imaging method is a technique used to detect exoplanets by directly capturing images or spectra of the planet itself. This method is used to observe large, young exoplanets that are far from their parent star and emit significant amounts of light in the infrared spectrum. The direct imaging method can provide detailed information about the planet's size, temperature, and atmospheric properties.

The microlensing method use to detect exoplanets by observing the gravitational lensing effect that occurs when a foreground star passes in front of a background star. If a planet is present in the foreground star system, its gravity will cause a deviation in the lensing effect, producing a characteristic brightening in the light curve of the background star.

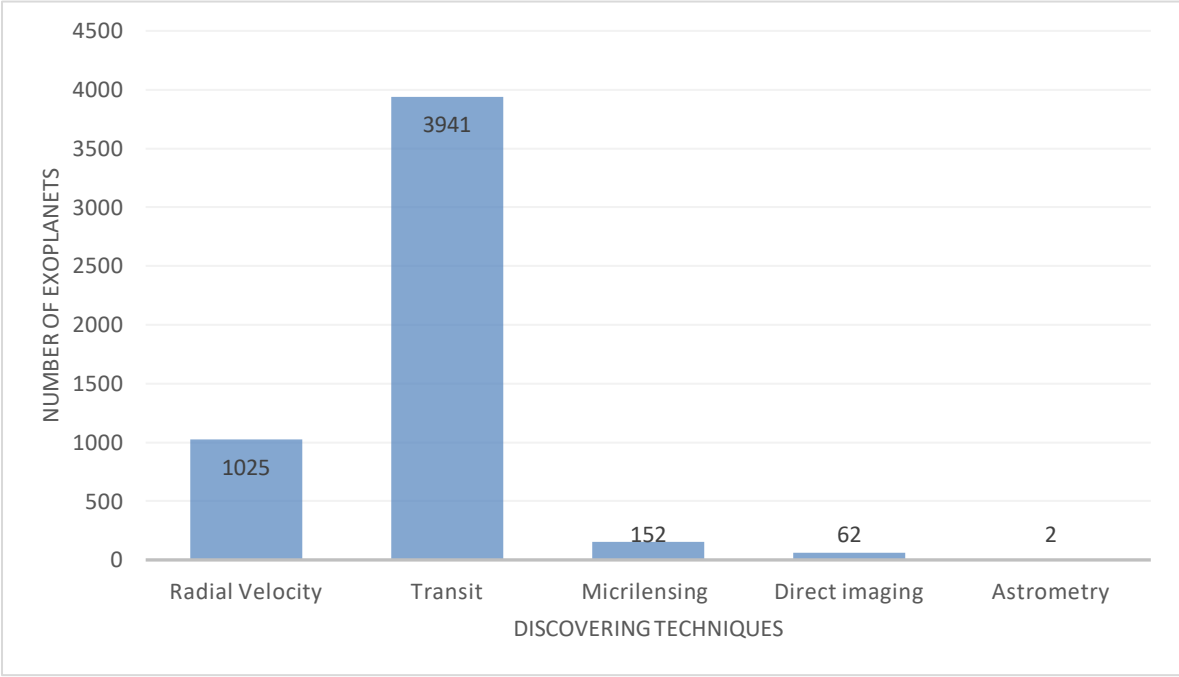


Figure 2.2: Exoplanets discovered by various techniques. Numbers are taken from NASA's website[9]

3 METHODOLOGY

The Kepler cumulative object of interest (KOI) dataset is a collection of data collected by the Kepler spacecraft on its mission to search for exoplanets. The dataset includes information on thousands of potential exoplanet candidates, including their orbital periods, sizes, and distances from their host star. The KOI dataset has been instrumental in the discovery of hundreds of confirmed exoplanets and continues to be used by astronomers and researchers to study the diversity of exoplanetary systems. In this data there are a total of 9,564 observations of which 4,839 are "FALSE POSITIVE", 2,671 "CONFIRMED" and 2054 "CANDIDATE" exoplanets. For the purpose of our research we dropped the observations with "CANDIDATE" flag.

3.1 Data Cleaning

Although, KCOI data table is relatively clean however there are some serious issues that's needs to be taken care of before training a machine learning model. We assessed every feature manually and dropped the features which have little to no predictive ability which includes unique identifiers, raw text comment etc. Additionally, we had to drop the feature that leak information to the model which in terms influence the prediction of the model. Information leakage occur when the data used to train a machine learning model contains information that would not have been available at the time when the prediction was made in the real world. This can result in overly optimistic performance metrics and make the model appear to be more accurate than it actually is.

Table 3.1: List of features dropped during initial data analysis and cleaning

Feature	**Reason**
kep_id	Unique identifier
kepoi_name	Unique identifier
kepler_name	Information Leakage
koi_pdisposition	Information Leakage
koi_score	Information Leakage
koi_disp_prov	Zero variance
koi_comment	Unstructured text
koi_time0bk	Duplicate
koi_longp	All zero
koi_ingress	All zero
koi_limbdark_mod	Unstructured text
koi_Idm_coeff3	Zero variance
koi_tce_delivname	Unstructured text
koi_count	Information Leakage
koi_trans_mod	Unstructured text
koi_mode_dof	All zero
koi_chisq	All zero
koi_sage	All zero
ra	Coordinate
dec	Coordinate

3.2 Missing Data

Different data, software and hardware issues led to missing value during Kepler mission and most of the time NASA scientists discard these observations[10]. But in this research we retained these missing observations using KNN imputation method.KNN imputation is a statistical method used to fill in missing values in a dataset by using the k nearest neighbors of the missing values. The idea

behind this method is that a value for the missing data point can be estimated based on its similarity to other data points in the dataset. The missing value is replaced with the average value of its k nearest neighbors. This method is simple and effective for small amounts of missing data, but can lead to bias in the imputed values as the number of missing values increases.

3.3 Correlation Analysis

We have done a correlation analysis of the data. Fig-3.1 shows the hierarchical correlation plot of highly correlated features. The features on the upper right hand of this figure are highly correlated. Several of these features can be dropped to reduce the computational complexity of machine learning algorithms if needed.

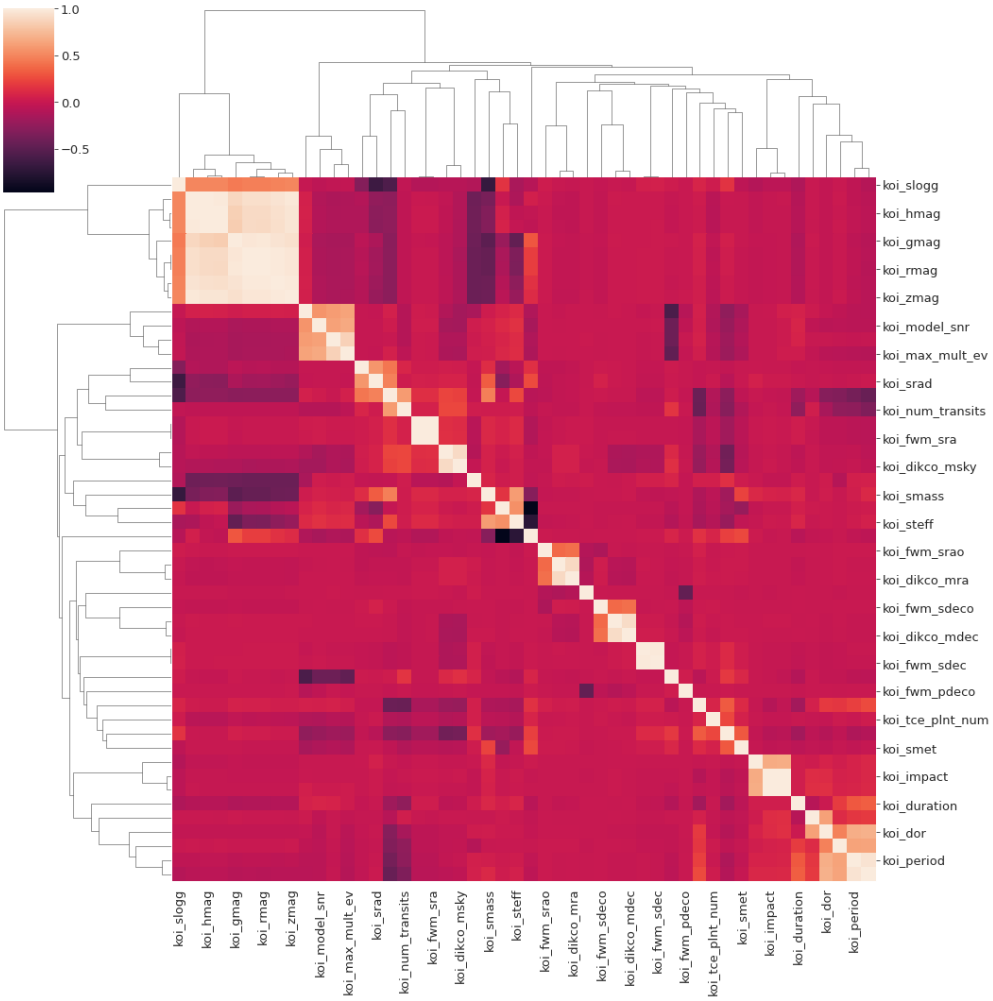


Figure 3.1 Hierarchical correlation plot of the cleaned KCOI data

3.4 Feature Selection

Feature selection is a technique in machine learning and data science that involves choosing a subset of relevant features from a larger set of features to be used in model building. The goal is to identify and select the most important features having the greatest impact on the outcome of the model, while removing those features that have little or no effect.

In this study we used three well known feature selection techniques named ANOVA f-test, Mutual Information Gain and Recursive Feature Elimination.

3.4.1 ANOVA F-test

ANOVA (Analysis of Variance) F-test is a statistical test to determine if there is a significant difference between the means of two or more groups[11]. The F-test is used to test the null hypothesis that all group means are equal against the alternative hypothesis that at least one group mean is different. The test statistic used in the ANOVA F-test is the F-ratio, which is the ratio of the variation between the groups to the variation within the groups.

The F-ratio follows the F-distribution with degrees of freedom between and degrees of freedom within. If the F-ratio is large, it suggests that there is a significant difference between the means of the groups, and the null hypothesis is rejected. The p-value is calculated from the F-ratio and the degrees of freedom to determine the level of significance.

If $\overline{X_i}$ = Mean of individual groups, $\overline{X}$ = Mean of all observations, n = Number of observations, m = Number of groups, $df_b$and df_w are the degrees of freedom between and within then,

The variation within the groups,

$$SSW = \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2 \tag{1}$$

$$df_w = (n - m)$$

$$MSW = \frac{SSW}{df_w} \tag{2}$$

The variation between the groups,

$$SSB = \sum_{i=1}^{n} n_i \left(\overline{X_i} - \overline{X}\right)^2 \tag{3}$$

$$df_b = (m - 1)$$

$$MSB = \frac{SSB}{df_b} \tag{4}$$

Then the F ratio is calculated as,

$$F = \frac{MSB}{MSW} \tag{5}$$

3.4.2 Mutual Information Gain

Entropy, condition entropy, joint entropy, and mutual information are some of the essential ideas of information theory that are primarily introduced in this section [12]–[14]. In 1948, Shannon introduced a new way to quantify information which proposed the concept of information entropy in his "A mathematical theory of communication". Entropy is a fundamental concept in information theory, which is the study of how to represent, transmit, and process information. Entropy is used to measure the amount of uncertainty or randomness in a system or a signal.

Let X and Y be two discrete random variable with n and m number of different values $X = \{x_1, x_2, \dots\dots\dots, x_n\}$and $Y = \{y_1, y_2, \dots\dots\dots, y_m\}$, and the entropy of X is denoted by $H(X)$:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i) \tag{6}$$

where,

$$p(x_i) = \frac{number\ of\ instants\ with\ value\ x_i}{total\ number\ of\ instants}$$

The entropy for a continuous random variable is defined as,

$$H(X) = -\int p(x)\ logp(x)\ dx \tag{6}$$

The conditional entropy of variable Y is the amount of uncertainty left in variable Y after the introduction of variable X and it is defined as follows,

$$H(Y|X) = -\sum_{x_i \in X}\sum_{y_j \in Y} p(y_j, x_i)\ logp(y_j|x_i) \tag{7}$$

The joint entropy of X and Y is the uncertainty that occurs simultaneously with two variables and it is denoted as $H(X,Y)$ and defined as,

$$H(X,Y) = -\sum_{x_i \in X}\sum_{y_j \in Y} p(y_j, x_i)\ logp(y_j, x_i) \tag{8}$$

where$p(y_j, x_i)$is the joint probability of y_j and x_i

The relationships between entropy, conditional entropy and joint entropy is as follows,

$$H(Y,X) = H(X,Y) = \begin{cases} H(X) + H(Y|X) \\ H(Y) + H(X|Y) \end{cases} \tag{9}$$

Mutual information gain (also known as Mutual Information) is a measure of the reduction in uncertainty of one random variable given the knowledge of another random variable. It is a non-negative value that quantifies the dependence or association between two variables.

In information theory, mutual information is used to determine the amount of information one random variable contains about another [16]. The more correlated two variables are, the higher the mutual information between them. Mathematically, mutual information can be defined as the expectation of the Kullback-Leibler (KL) divergence between the joint distribution of the two variables and their individual distributions. In other words, it is the reduction in the entropy (uncertainty) of one random variable given the knowledge of another random variable. The mutual information between variable Y and variable X is denoted by $I(Y;X)$ and formulized as follows,

$$I(Y;X) = \begin{cases} H(Y) - H(X|Y) \\ H(X) - H(Y|X) \\ H(Y) + H(X) - H(Y,X) \end{cases} \quad (10)$$

$$I(Y;X) = -\sum_{x \in X} \sum_{y \in Y} p(y,x)\, log \frac{p(y,x)}{p(y)p(x)} \quad (11)$$

Mutual information has applications in a wide range of fields, including data analysis, machine learning, and image processing. In data analysis, it can be used to identify relationships between variables, determine the relevance of features, and select the most informative features for a particular task. In machine learning, mutual information can be used to select features for feature selection or feature extraction.

3.4.3 Recursive Feature Elimination

The RFE) is used to select a subset of features from a large number of features in a dataset. This method is used in machine learning and data analysis to improve the accuracy and interpretability of models by removing irrelevant, redundant, or noisy features. In this study we used recursive feature elimination technique with random forest as estimator.

The basic idea behind RFE is to recursively eliminate features, starting with the most insignificant features, and then build a model with the remaining features. This process is repeated until the desired number of features is reached or until the accuracy of the model no longer improves. Here is how the RFE method works:

- Initialize the feature set: The initial feature set contains all the features in the dataset.
- Build a model: A model is built using the initial feature set and evaluated based on a certain performance metric.
- Eliminate the least significant feature: The feature that has the smallest contribution to the model's performance is removed from the feature set.
- Re-build the model: A new model is built using the reduced feature set and evaluated again based on the performance metric.
- Repeat steps 3 and 4 until the desired number of features is reached or until the accuracy of the model no longer improves.

RFE is useful because it allows us to select the most relevant features for a given problem, reducing the dimensionality of the data and increasing the interpretability of the model. It is also computationally efficient, making it a good choice for large datasets with many features.

5 RESULTS AND DISCUSSION

5.1 ANOVA F-test Results

After data preprocessing and analysis we've used several statistical methods to select features recursively and fed them to machine learning models to evaluate the performance. We used 5 fold cross validation to train our models. As we trained our machine learning models on features selected recursively using ANOVA F-test we observe that there isn't much gain in performance metrics compared to the models trained on the most important 15 features. Figure 4.1 shows the performance of different models on the different selected features using ANOVA F-test.

Although every model performed pretty good CatBoost performed better in particular. It gives 98.27% accuracy. The other metrics are also very satisfactory.

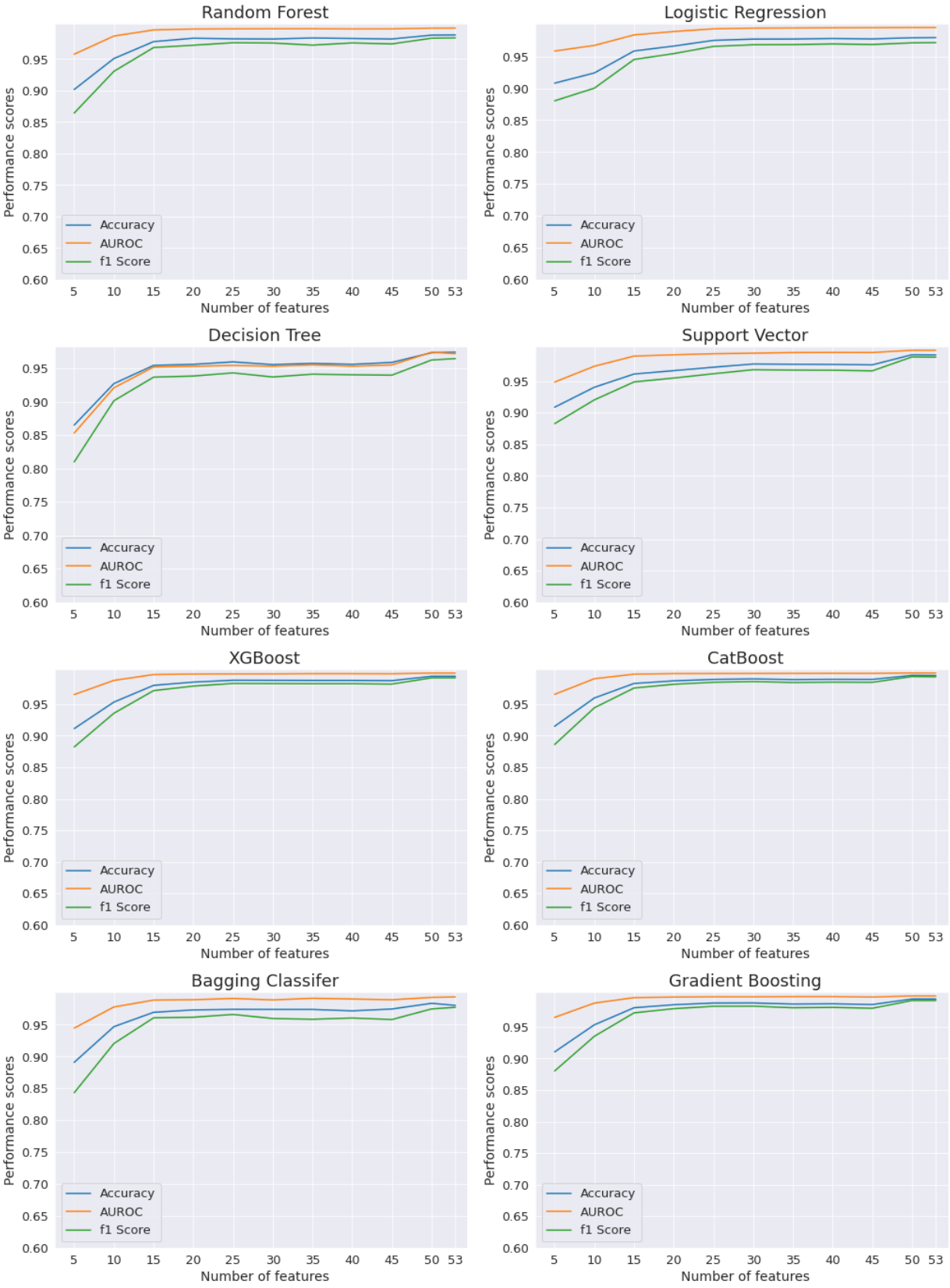


Figure 4.1: Performance of different models on ANOVA F-test selected features

Table 2: The top 11 selected features by ANOVA F-test

Features	Description	ANOVA Score
koi_fpflag_ss	Stellar Eclipse Flag	2168.33
koi_fpflag_co	Centroid Offset Flag	1709.89
koi_depth	Transit Depth (parts per million)	422.48
koi_fittype	Planetary Fit Type	425.79
koi_incl	Inclination (deg)	777.46
koi_teq	Equilibrium Temperature (Kelvin)	565.19
koi_Idm_coeff1	Limb Darkening Coefficients	387.57
koi_num_transits	Number of Transits	350.12
koi_smet	Stellar Metallicity	897.37
koi_dicco_msky	PRF $\Delta\theta_{SQ}(OOT)$units: arcseconds	1338.43
koi_dikco_msky	PRF $\Delta\theta_{SQ}(KIC)$units: arcseconds	1229.21

Table 3: Performance scores of the machine learning models trained on the top 11 features selected by ANOVA F-test

Models	**Accuracy**	**Precision**	**Recall**	**f1 score**	**AUROC**
Logistic Regression	95.86%	91.79%	97.64%	94.52%	98.38%
Decision Tree	95.43%	93.95%	93.67%	93.67%	95.19%
Random Forest	97.76%	98.28%	95.58%	96.82%	99.62%
Support Vector	96.10%	92.46%	97.60%	94.85%	98.93%
XGBoost	97.95%	96.72%	97.53%	97.12%	99.66%
AdaBoost	97.55%	96.03%	97.12%	96.56%	99.52%
CatBoost	**98.27%**	**97.66%**	97.45%	**97.55%**	**99.74%**
Bagging Classifier	96.92%	97.60%	94.38%	96.06%	98.84%

5.2 Mutual Information Gain (MIG) Results

The performance of the models on the features selected by mutual information gain was is different from the results obtained from ANOVA F-test. It needed 30 features to hit the plateau of performance curve. Figure 4.2 shows the performance of different models on the different selected features using MIG.

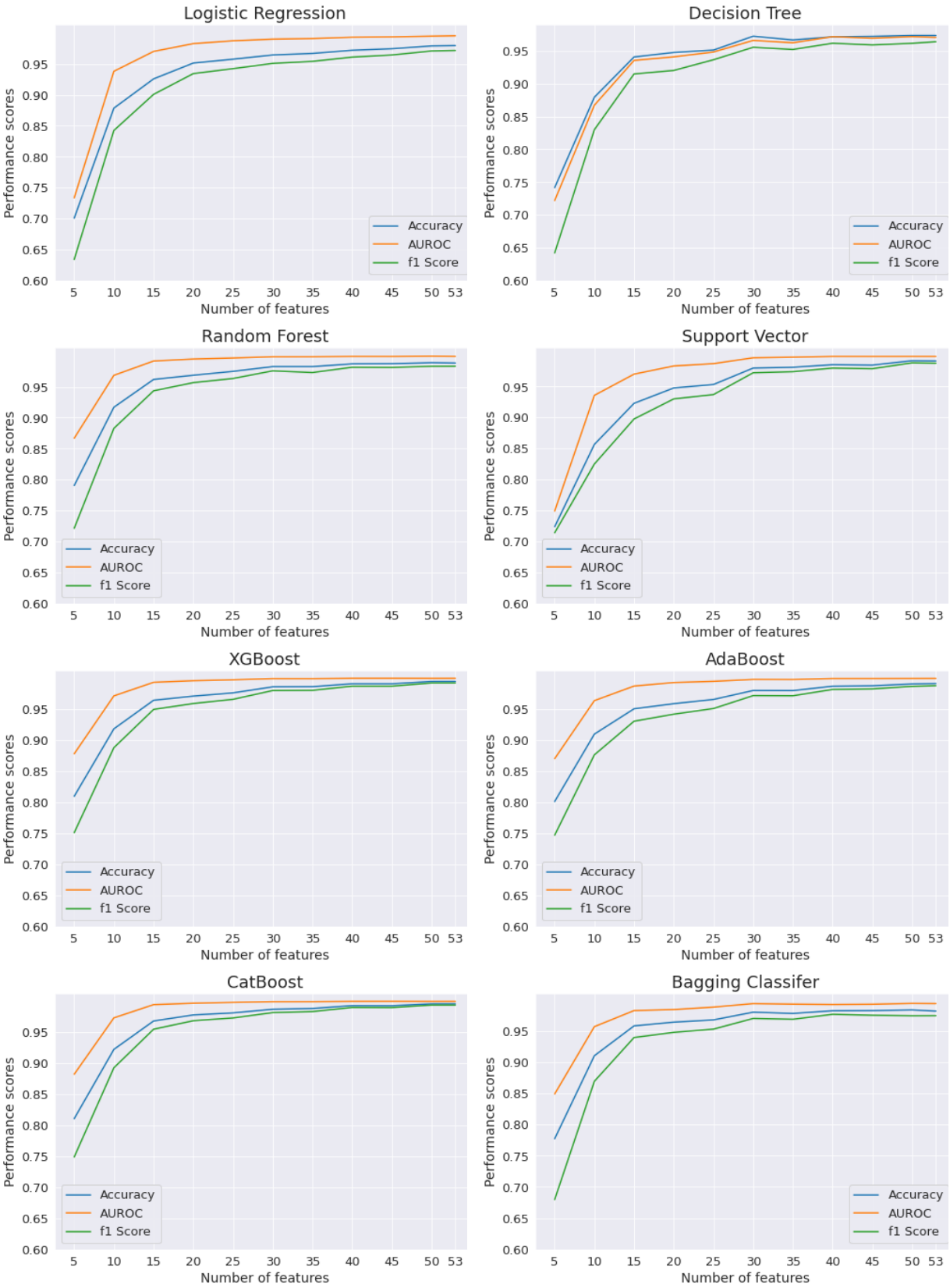


Figure 4.2: Performance of different models on selected features by Mutual Information Gain

Table 4: The top 23 selected features by Mutual Information Gain (MIG)

Features	Description	MIG Score
koi_fpflag_ss	Stellar Eclipse Flag	0.156
koi_period	Orbital Period (days)	0.145
koi_impact	Impact Parameter	0.098
koi_depth	Transit Depth (parts per million)	0.105
koi_ror	Planet-Star Radius Ratio	0.194
koi_prad	Planetary Radius (Earth radii)	0.196
koi_sma	Orbit Semi-Major Axis (Astronomical Unit (au))	0.093
koi_incl	Inclination (deg)	0.135
koi_teq	Equilibrium Temperature (Kelvin)	0.102
koi_insol	Insolation Flux [Earth flux]	0.107
koi_Idm_coeff1	Limb Darkening Coefficients	0.165
koi_ldm_coeff2	Limb Darkening Coefficients	0.156
koi_num_transits	Number of Transits	0.122
koi_steff	Stellar Effective Temperature (Kelvin)	0.100
koi_smet	Stellar Metallicity	0.116
koi_fwm_sdeco	FW $\Delta\delta(OOT)$units: arcseconds	0.194
koi_fwm_pdeco	FW Source $\Delta\delta(OOT)$units: arcseconds	0.139
koi_dicco_mra	PRF $\Delta\alpha_{SQ}(OOT)$units: arcseconds	0.193
koi_dicco_mdec	PRF $\Delta\alpha_{SQ}(KIC)$units: arcseconds	0.195
koi_dicco_msky	PRF $\Delta\theta_{SQ}(OOT)$units: arcseconds	0.256
koi_dikco_mra	PRF $\Delta\delta_{SQ}(KIC)$units: arcseconds	0.205
koi_dikco_mdec	PRF $\Delta\delta_{SQ}(KIC)$units: arcseconds	0.196
koi_dikco_msky	PRF $\Delta\theta_{SQ}(KIC)$units: arcseconds	0.261

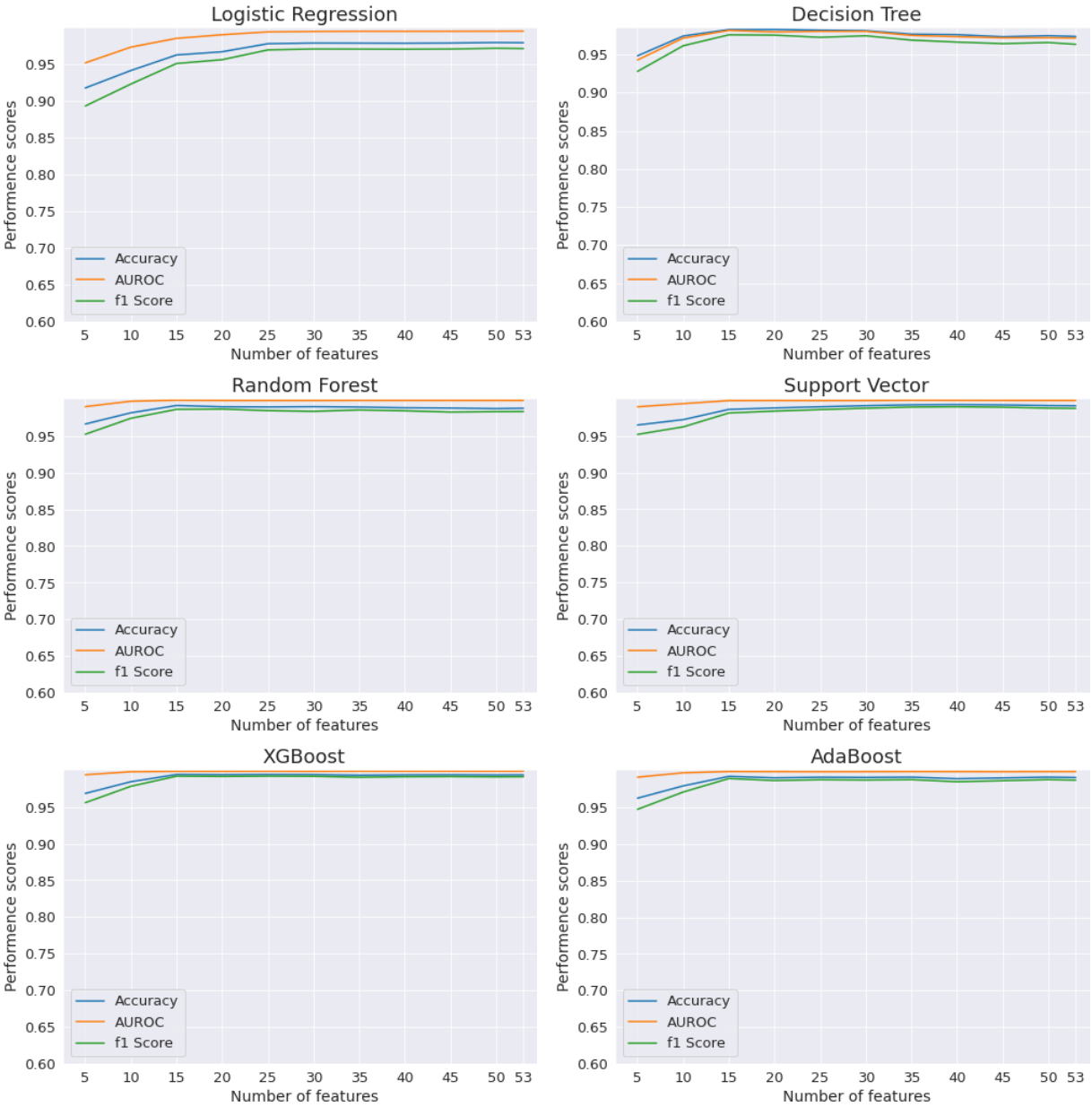
The models trained on the features mentioned in the Table 4 gives good performance score but among them CatBoost seems to be giving the overall best results again. It gives an accuracy of 98.66%. Random Forest also shows a good overall performance.

Table 5: Performance scores of the machine learning model trained on the top 23 features selected by Mutual Information Gain

Models	Accuracy	Precision	Recall	f1 score	AUROC
Logistic Regression	96.44%	93.60%	96.63%	95.08%	99.00%
Decision Tree	97.26%	96.09%	95.62%	95.57%	96.61%
Random Forest	98.26%	**98.98%**	96.33%	97.57%	99.85%
Support Vector	97.98%	95.99%	**98.54%**	97.22%	99.64%
XGBoost	98.52%	97.80%	98.09%	97.93%	99.84%
AdaBoost	97.96%	96.83%	97.53%	97.14%	99.75%
CatBoost	**98.66%**	98.02%	98.24%	**98.11%**	**99.87%**
Bagging Classifer	98.00%	97.93%	95.96%	97.00%	99.38%

5.3 Recursive Feature Elimination (RFE) Results

Similar to ANOVA F-test there is also seems to be not much gain in the performance of the models compared with the results of the models trained on top 11 features selected by recursive feature elimination technique. Also, the performance on these features is better elimination.



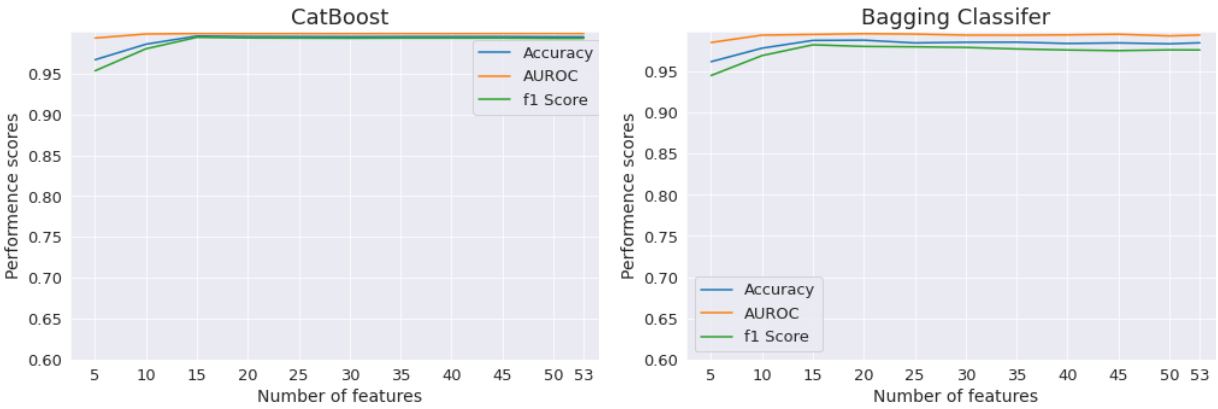


Figure 4.3: Performance of different models on selected features by Recursive Feature

Table 6: The top 11 selected features by Recursive Feature Elimination

Features	Description	RFE Score
koi_fpflag_nt	Not Transit-Like Flag	0.098
koi_fpflag_ec	Ephemeris Match Indicates Contamination Flag	0.019
koi_depth	Transit Depth (parts per million)	0.033
koi_ror	Planet-Star Radius Ratio	0.054
koi_prad	Planetary Radius (Earth radii)	0.076
koi_insol	Insolation Flux [Earth flux]	0.024
koi_dor	Planet-Star Distance over Star Radius	0.032
koi_max_sngle_ev	Maximum Single Event Statistic	0.015
koi_model_snr	Transit Signal-to-Noise	0.036
koi_dicco_msky	PRF $\Delta\theta_{SQ}(OOT)$units: arcseconds	0.095
koi_dikco_msky	PRF $\Delta\theta_{SQ}(KIC)$units: arcseconds	0.117

The models trained by the selected feature on recursive feature elimination give relatively better result than previous methods. Here also we observe CatBoost is overall the better performer. Random Forest, Gradient Boosting and MLP classifier also performed almost as good as CatBoost.

Table 7: Performance scores of the machine learning model trained on the top 11 features selected by Recursive Feature Elimination

Models	**Accuracy**	**Precision**	**Recall**	**f1 score**	**AUROC**
Logistic Regression	96.30%	92.63%	98.05%	95.14%	98.56%
Decision Tree	98.26%	97.45%	97.64%	97.57%	98.15%
Random Forest	99.21%	99.81%	97.79%	98.69%	**99.94%**
Support Vector	98.63%	97.33%	99.06%	98.14%	99.83%
XGBoost	99.44%	99.41%	99.03%	99.21%	99.88%
AdaBoost	99.25%	98.86%	99.06%	98.96%	99.92%
CatBoost	**99.61%**	**99.85%**	99.06%	**99.46%**	99.91%
Bagging Classifier	98.76%	99.07%	97.49%	98.23%	99.49%

6 CONCLUSION

Exoplanet detection is a rapidly growing field in astronomy that has the potential to greatly advance our understanding of the universe. In order to detect exoplanets, astronomers analyze large amounts of data from various sources, such as transit photometry, radial velocity, and direct imaging. One of the important challenges in this field is to accurately identify the most relevant features in the data that can be used to detect exoplanets and distinguish them from other celestial objects. In this work, we investigated the Kepler Cumulative Object of Interest (KCOI) dataset hosted in NASAExoplanet Archive. We performed pre-processing to get rid of useless features and features that leak information to the machine learning model. After, we performed the feature selection method recursively and trained the machine learning models with the selected features and searched for the features that gives the best performance. Here, we only used the data that were already uploaded by NASA. In the future, we are aiming to extract the features from the raw lightcurve data from the Kepler satellite to further investigate the important features that leads to better identification of the exoplanets. We want to explore dimensionality reduction techniques, such as principal component analysis (PCA) and independent component analysis (ICA) to reduce the number of features in the data and identify the most important ones. These techniques can help to simplify the data and make it easier to analyze, while also preserving the most relevant information for exoplanet detection. We also want to interpret our models using LIME and SHAP analysis.

7 REFERENCES

[1] "NASA Exoplanet Archive." https://exoplanetarchive.ipac.caltech.edu/ (accessed Feb. 09, 2023).

[2] C. J. Shallue and A. Vanderburg, "Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90," *Astron J*, vol. 155, no. 2, p. 94, Jan. 2018, doi: 10.3847/1538-3881/aa9e09.

[3] Y. Jin, L. Yang, and C.-E. Chiang, "Identifying Exoplanets with Machine Learning Methods: A Preliminary Study," *International Journal on Cybernetics & Informatics*, vol. 11, no. 2, pp. 31–42, Apr. 2022, doi: 10.5121/ijci.2022.110203.

[4] A. Malik, B. P. Moster, and C. Obermeier, "Exoplanet Detection using Machine Learning," Nov. 2020, doi: 10.1093/mnras/stab3692.

[5] G. Clayton Sturrock, B. Manry, S. Rafiqi, G. Clayton, and G. Sturrock, "Machine Learning Pipeline for Exoplanet Classification.".

[6] M. Bugueno, F. Mena, and M. Araya, "Refining exoplanet detection using supervised learning and feature engineering," in *Proceedings - 2018 44th Latin American Computing Conference, CLEI 2018*, Oct. 2018, pp. 278–287. doi: 10.1109/CLEI.2018.00041.

[7] M. Jara-Maldonado, V. Alarcon-Aquino, R. Rosas-Romero, O. Starostenko, and J. M. Ramirez-Cortes, "Transiting Exoplanet Discovery Using Machine Learning Techniques: A Survey," *Earth Science Informatics*, vol. 13, no. 3. Springer, pp. 573–600, Sep. 01, 2020. doi: 10.1007/s12145-020-00464-7.

[8] "5 Ways to Find a Planet | Explore – Exoplanet Exploration: Planets Beyond our Solar System." https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/ (accessed Feb. 09, 2023).

[9] T. D. Morton *et al.*, " False Positive Probabilities For All Kepler Objects Of Interest: 1284 Newly Validated Planets And 428 Likely False Positives." *Astrophys J*, vol. 822, no. 2, p. 86, May 2016, doi: 10.3847/0004-637x/822/2/86.

[10] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, "Feature Selection and Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor," in *Procedia Computer Science*, 2015, vol. 54, pp. 301–310. doi: 10.1016/j.procs.2015.06.035.

[11] L. Hu, W. Gao, K. Zhao, P. Zhang, and F. Wang, "Feature selection considering two types of feature relevancy and feature interdependency," *Expert SystAppl*, vol. 93, pp. 423–434, Mar. 2018, doi: 10.1016/j.eswa.2017.10.016.

[12] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Trans Neural Netw*, vol. 13, no. 1, pp. 143–159, Jan. 2002, doi: 10.1109/72.977291.

[13] X. Tang, Y. Dai, P. Sun, and S. Meng, "Interaction-based feature selection using Factorial Design," *Neurocomputing*, vol. 281, pp. 47–54, Mar. 2018, doi: 10.1016/j.neucom.2017.11.058.

[14] X. Wang, B. Guo, Y. Shen, C. Zhou, and X. Duan, "Input Feature Selection Method Based on Feature Set Equivalence and Mutual Information Gain Maximization," *IEEE Access*, vol. 7, pp. 151525–151538, 2019, doi: 10.1109/ACCESS.2019.2948095.

[15] T. Hastie, J. Friedman, and R. Tibshirani, "The Elements of Statistical Learning," 2001, doi: 10.1007/978-0-387-21606-5.

[16] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18. pp. 2395–2399, May 2008. doi: 10.1161/CIRCULATIONAHA.106.682658.

[17] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, "Comparison of the CatBoost Classifier with other Machine Learning Methods," 2020.[Online]. Available: www.ijacsa.thesai.org

[18] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics*, vol. 18, no. 6. pp. 275–285, Jun. 2004. doi: 10.1002/cem.873.

[19] R. Punnoose and C. Xlri -Xavier, "Prediction of Employee Turnover in Organizations using Machine Learning Algorithms A case for Extreme Gradient Boosting," *IJARAI) International Journal of Advanced Research in Artificial Intelligence*, vol. 5, no. 9, 2016, Accessed: Feb. 10, 2023. [Online]. Available: www.ijarai.thesai.org

[20] S. Lessmann and S. Voß, "A reference model for customer-centric data mining with support vector machines," *Eur J Oper Res*, vol. 199, no. 2, pp. 520–530, Dec. 2009, doi: 10.1016/J.EJOR.2008.12.017.

[21] T. Chen and C. Guestrin, "XGBoost: Reliable Large-scale Tree Boosting System".

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Adv Neural Inf Process Syst*, vol. 31, 2018, Accessed: Feb. 11, 2023. [Online]. Available: https://github.com/catboost/catboost

[23] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999, Accessed: Feb. 11, 2023. [Online]. Available: www.research.att.com/fyoav,

[24] J. Stephen Bassi, E. Gbenga Dada, A. AbdulkadirHamidu, M. Dauda Elijah, and C. Author, "Students Graduation on Time Prediction Model Using Artificial Neural Network," vol. 21, no. 3, pp. 28–35, doi: 10.9790/0661-2103012835.