Detection of Exoplanets

Amisha Srivastava[1], Mansi Anil Patil[2], Yash Agrawal[3]
Electrical Engineering and Computer Science Department
IISER Bhopal
[1]*amisha18@iiserb.ac.in*
[2]*mansi18@iiserb.ac.in*
[3]*yasha18@iiserb.ac.in*

***Abstract*—** The focus of this project is to classify the data consisting of stars and their flux density effortlessly on a large scale to predict the existence of exoplanets in their vicinity without any human intervention. The usage of artificial intelligence techniques are aimed at elucidating the problem of increased computational time and complexity associated with existing deep learning techniques. Various machine learning algorithms and classifiers have been explored and applied on the dataset in terms of runtime, accuracy and quality of label output and a comparative analysis has been done on the performance of each model in consideration.

***Keywords*—** exoplanet, transit method, supervised learning, classification, machine learning, algorithms**.**

I. Introduction

Our solar system is 4.6 billion years old, yet there's no way to know how it came to be. We have no notion if it was one of many or a one-of-a-kind. Is there life outside of Earth? Will humans be able to travel there in the near future? The evolution of the young planetary systems may provide answers to some of the issues. A subproblem should be addressed first in order to find solutions to these challenges. After the discovery of planets in space, further investigation would be required to see if it is habitable. Without a question, looking for life on a new planet is a huge and time-consuming undertaking in and of itself.

However, for the time being, the focus is on the subject of discovering new planets in deep space, commonly known as Exoplanets. Exoplanets are exciting because they may provide answers to questions about the universe. An extrasolar planet, often known as an exoplanet, is a planet that exists outside of our solar system. Exoplanets have become one of astronomy's most active research areas. Exoplanets are found in a wide range of sizes and orbits. Some are massive planets orbiting their parent stars; some are cold, while others are rocky. NASA and other organisations are on the lookout for a planet the size of Earth that orbits a sun-like star in the habitable zone. The habitable zone is the area surrounding a star where people can live.

As a result, our research focuses on discovering exoplanets, which may then be studied to look for signs of life. This, in turn, might lead to a game-changing revelation for humanity. Our research interests are in

astrophysics and the use of artificial intelligence (AI) to transdisciplinary subjects.

The main goal of this project is to find an exoplanet (or exoplanets) orbiting a group of stars in the dataset to make it easier to find habitable planets using an appropriate planet detection method. We'll look at which algorithm is best at predicting the presence of an exoplanet in the region of the star in question by using several AI techniques. The test data labels can be properly predicted after comparing the performance of various algorithms using evaluation measures. AI can be used to detect exoplanets in this way, avoiding human mistakes.

II. Contributions

We found a paper published by the astronomers of the "Center for Astrophysics at Harvard University and Smithsonian Institution" that was based on the same topic but used Deep learning methods for prediction after studying existing techniques and ongoing research work on the problem. Similarly, most of the work in this field has been published using such deep learning models.

- We believe that we can make significant contributions to this field of study by using Machine learning methods which is a sub-branch of Artificial Intelligence to provide more accurate results and get a deeper insight into the existing solutions.
- In this project, we will be focusing on the “Transit Method” which has been deployed using deep learning techniques in existing works. We will try to improvise the method with the help of supervised learning techniques and then compare it with the result obtained from the neural networks classifier.
- At the end of our research, after applying several machine learning algorithms we were able to obtain desirable results and get a supervised learning model to predict the exoplanets in the vicinity of the stars provided in the dataset which we consider as a starting step towards an innovative idea in this area of detecting exoplanets.

III. Background

The Kepler Space Telescope was built to determine the frequency of Earth-sized planets orbiting Sun-like stars, but these planets are on the verge of the mission's detection sensitivity. Automatically and accurately assessing the likelihood that individual candidates are indeed planets, even at low signal-to-noise ratios, is required to accurately determine the occurrence rate of these planets.

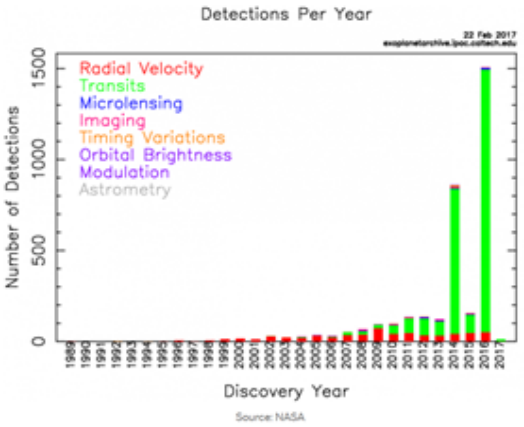
Previously people have worked on different methods for classifying potential planet signals using deep learning, a class of machine learning algorithms, where a deep convolutional neural network is trained to predict whether a given signal is a transiting exoplanet or a false positive caused by astrophysical or instrumental phenomena. Such works inspired us to develop techniques and make analysis of our own to try and gain even better results. The

exoplanet detection problem helped us dive deeper into the machine learning and application realm of Artificial Intelligence.

IV. Materials and Methods

Many traditional methods have been used for detection of exoplanets :

- Pulsar Timing
- Astrometry
- Doppler Effect
- Gravitational Microlensing
- Transit Method



As shown in the graph, transit detection (green in the graph) is the method that contributes significantly more than any other method. And it was with this manner that we discovered the exoplanets.

The transit approach works by looking for any periodic decline in brightness in a pre-selected group of stars. If any of the picked stars show a decrease in brightness, it is probable that a planet moving in front of it is reducing the star's observable brightness.If there is such an observable pattern in the data, we can use AI to compel programmes to learn the patterns from data and construct algorithms capable of automatically predicting the outcomes. For this, we'll need the data and a description of it, as well as information on the approaches we'll be employing.

Data and the source:

NASA launched the Kepler telescope in 2009 with the goal of discovering new planets circling other stars in our Milky Way galaxy or beyond. It also employed the "Transit Detection Method," which we covered before, to find planets. During its time on the job, Kepler detected approximately 2000 exoplanets, with additional candidate planets awaiting confirmation.

Organizations who were holding the data published a part of that for study and research purposes. We found this dataset on Kaggle.

Dataset:

The data we will be using consists of the change in flux (light intensity) of several thousand stars. Each star has a binary label 1 and 2. The label 2 indicates that the star is confirmed to have at least one exoplanet in orbit.

Trainset:

- ➔ 5087 rows or observations
- ➔ 3198 columns or features
- ➔ Column 1 is the label vector. Column 2-3198 are the flux values over time

Test set:

- ➔ 570 rows or observation
- ➔ 3198 columns or features
- ➔ Column 1 is the label vector. Column 2-3198 are the flux values over time.

Evaluation metrics: Evaluating a model is a major part of building an effective machine learning model. The most frequent classification evaluation metric that we use should be *'Accuracy'*.However, it is not always true and can be misleading in some situations. So, we used following evaluation metrics for analysing the best classifier :

1. ***Confusion Matrix:*** The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.
2. The key classification metrics:
    a. ***Accuracy:*** It is the fraction of predictions our model got right.
    b. ***Recall:*** In a binary classification problem, recall is calculated as the number of true positives divided by the total number of true positives and false negatives.
    c. ***F-measure:*** It is the harmonic mean of a model's precision and recall.
3. *Receiver Operating Characteristic (ROC) curve :* It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0. Put another way, it plots the false alarm rate versus the hit rate. An operator may plot the ROC curve for the final model and choose a threshold that gives a desirable balance between the false positives and false negatives.

Data Analysis:

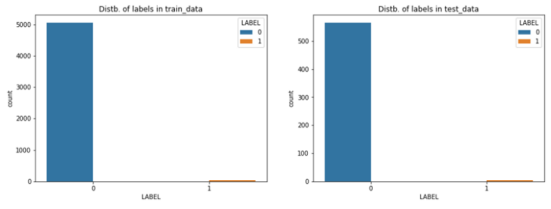
After importing the libraries and loading the train and test data, our dataset looks like:

	LABEL	FLUX.1	FLUX.2	FLUX.3	FLUX.4	FLUX.5	FLUX.6	FLUX.7
0	1	93.85	83.81	20.10	-26.98	-39.56	-124.71	-135.18
1	1	-38.88	-33.83	-58.54	-40.09	-79.31	-72.81	-86.55
2	1	532.64	535.92	513.73	496.92	456.45	466.00	464.50
3	1	326.52	347.39	302.35	298.13	317.74	312.70	322.33
4	1	-1107.21	-1112.59	-1118.95	-1095.10	-1057.55	-1034.48	-998.34

5 rows × 3198 columns

It is observed that there are 319 columns(i.e. the data is 3198 dimensional). One metric that we used is "Recall Score". So Recall is the metric which lies between 0 to 1. And that will be maximum(1) if there are all points which belong to positive class and also predicted as positive points and will be minimum(0) if there are all points which belong to positive class but predicted as negative points. Therefore, we changed our labels from 2 to 1 and 1 to 0. Distributions of class labels in both the train and test datasets:

```
Out of all  5087 points in training data, there are -
 5050 points with class label 1, which is  99.27 % of the whole data and
 37 points with class label 2, which is  0.73 % of the whole data.
------------------------------------------------------------------------
Out of all  570 points in the test data, there are -
 565 points with class label 1, which is  99.12 % of the whole data and
 5 points with class label 2, which is  0.88 % of the whole data.
```

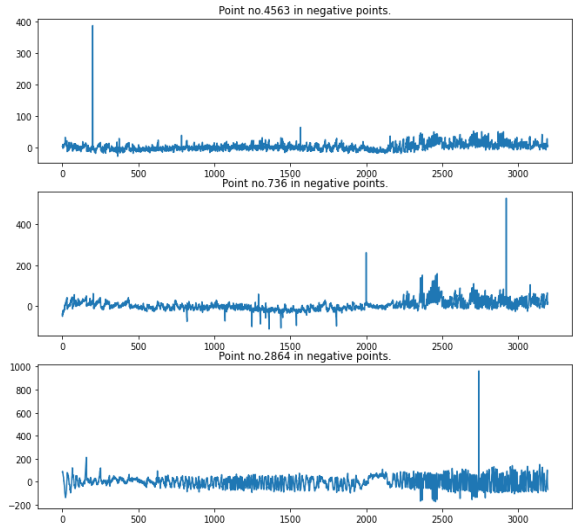
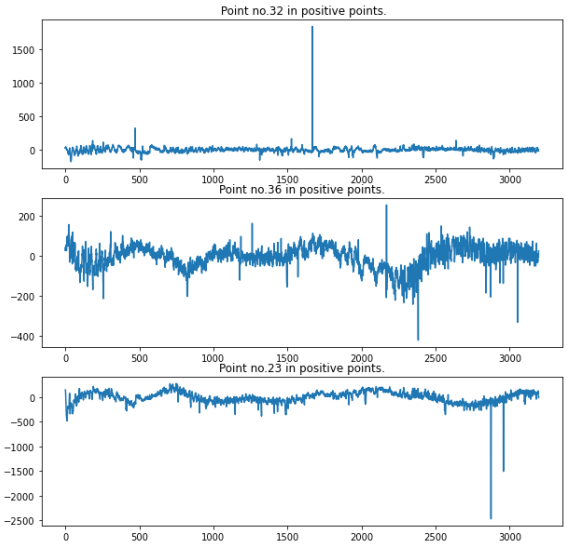


Distributions of labels in both datasets

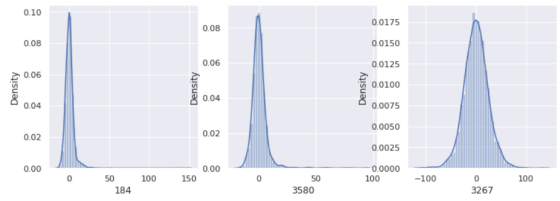
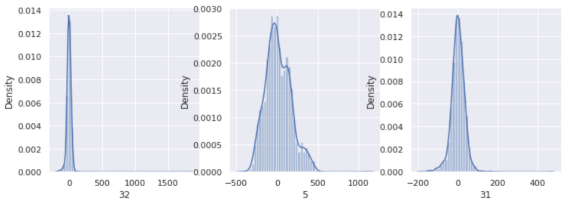
More than 99 percent of data points in both the train and test datasets belong to the negative class(class 1). And the positive

class (class 2) accounts for less than 1% of all data points. Simply put, it means we have more than 99 percent of the data points we need. It will not be classified as an exoplanet. As a result, only about 1% of observations are real Exoplanets. As it can see, the data is highly imbalanced.

Let's look at the actual data points from both the positive and negative classes now. To gain a basic understanding of the behaviour of points from both classes:

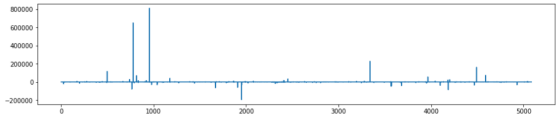
Plotting Points From Negative Class	Plotting Points From Positive Class
	
It can be seen in all three graphs of three data points, the range of flux values doesn't vary that much. It means that the intensity of light we receive from any star does not vary greatly. As a result, there should be no exoplanets. Because these graphs are made up of negative data points (those without an exoplanet), hence this notion is correct.	Just opposite from negative points here in all above 3 plots the flux values are ranging much with a pattern. It means that the light intensity values coming from the star are varying much. They simply go down and then again come up. Hence there should be an exoplanet that is orbiting the star. When that exoplanet orbits in front of the star then the intensity value of light goes down. And when after some time that star reaches behind the star then intensity values of light again come up.

Probability Distribution functions of points from negative class	Probability Distribution functions of points from positive class

	
In comparison to the positive class, the values of flux in negative class points come from a narrow range. It means that the intensity values of light emitted by any star are almost identical. As a result, no exoplanet is present.	Positive data points show that flux values come from a wide range, whereas negative data points show that flux values come from a narrow range. It signifies that the light intensity values from any star are dimming and then increasing again. As a result, the star has an exoplanet around it. That's why, as a result of the exoplanet, the light intensity fluctuates.

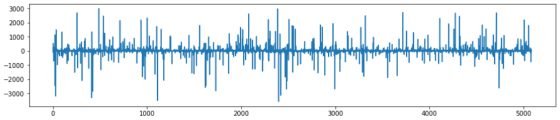
Data Preprocessing:

Detecting the Outliers:



As we can clearly see in the plot there are very few high lines. It means there are only a few values that don't lie with the rest of the values and they are extremely high or low.

Removing the outliers:

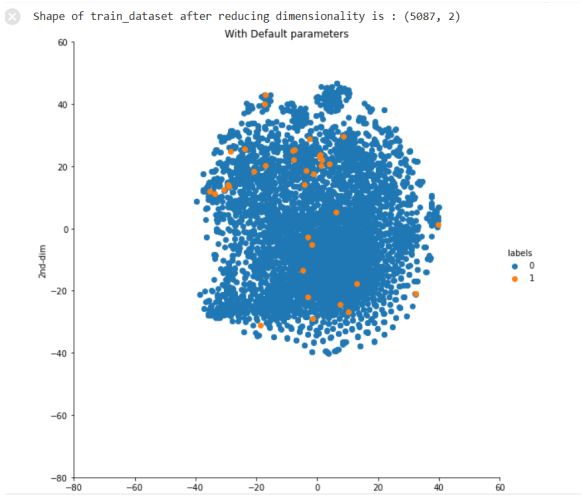


There is no line with very high value that is alone. It means there is no point that lies in the range of rest all values. Therefore, the outliers have been removed.

Dimensionality reduction:

After removing the outliers and min max scaling the dimensionality is reduced and the whole data is visualized at once. The Training data is more than 3k dimensional. To visualize it, we need to reduce its dimensionality to 2 or 3 dimensions. This is done using T distributed-Stochastic Neighborhoods Embeddings (TSNE)

The plot shown here is the training data after applying TSNE.

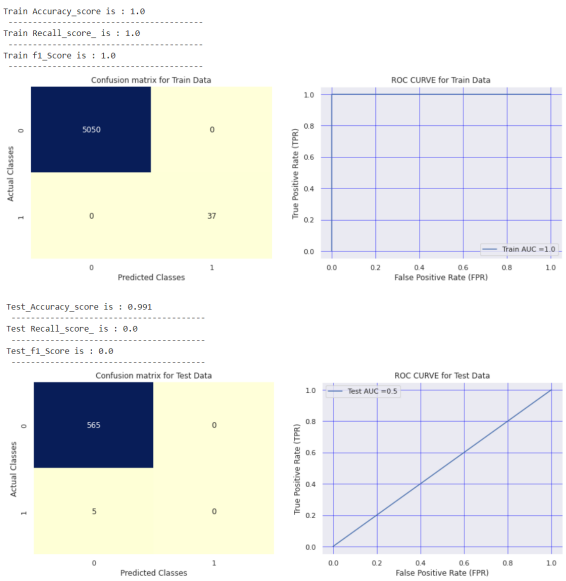


The positive points are clearly overlapping with negative classes points. This implies that it will be hard to make a model which

can separate points of both classes. It could be possible because we are going from 3k dimensions to just 2 dimensions.

Modeling on Imbalanced Data: We tried modelling on the imbalanced data. And these are the results:

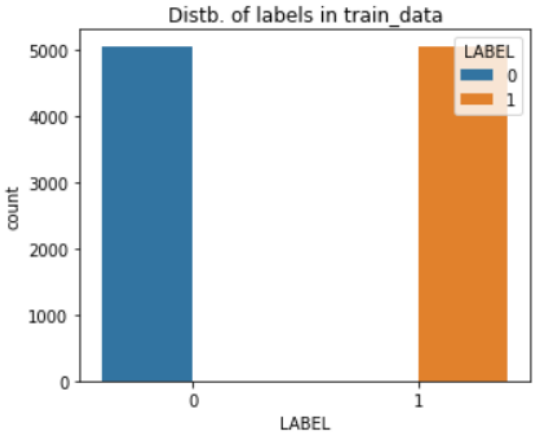
All models (KNN shown in fig) are performing very well on Train data. But bad on Test data. Also In Test data models are performing good on Negative class points. But bad on positive class points.The main reason for this is the " Imbalance nature of dataset" or in other words "lack of positive class points". Train score is high and test score is low then we are heading towards overfitting. We saw models are overlearning negative points and not learning much about positive points.



Oversampling and Balancing the data:

There are many ways we can get rid of the imbalance problem. Widely used methods are: Under sampling and Oversampling. We already had too little data(only 5087 points). Hence we couldn't afford under-sampling. So we oversampled our data and increased the number of points of minority class.

After oversampling the balanced data looks like this:



Now our data is balanced.

Modeling on Balanced Data:

This balanced data was then used for prediction with the following classifiers:

1. **KNN**- K nearest neighbors are known as one of the simplest nonparametric classifiers. For a fixed value of k, KNN assigns a new observation to the class of majority of the k nearest neighbors.
2. **Logistic regression** - Logistic regression is a statistical method used to demonstrate the relationship between the categories and the features by estimating probabilities using a logistic function in order to find a linear decision boundary between the classes
3. **Decision Tree**- A decision tree classifier is a systematic approach for multiclass classification. It poses a set of questions related to its features, to the dataset.
4. **Support Vector Machine** - Support Vector Machine follows an optimisation algorithm that produces linear vectors that try to maximally separate the target classes.It searches

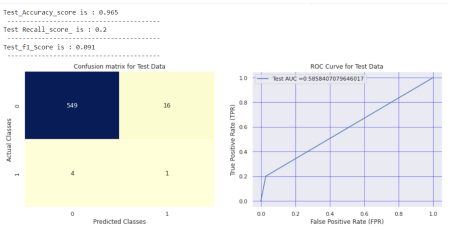
for a separating hyperplane for the categories.

5. **Random Forest** - A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees
6. **Naive Bayes** - It is based on the Bayes theorem and an independence hypothesis, generating a statistical classifier based on probabilities.
7. **Neural Networks** - A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

V. Results

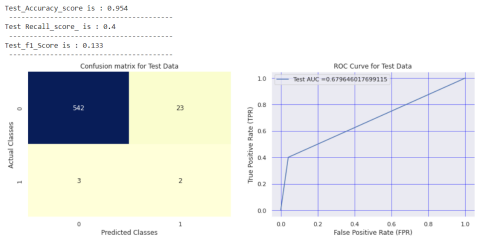
Using the above models for prediction, we got the following evaluation metrics scores as a means of results :

1. Naive Bayes Algorithm:



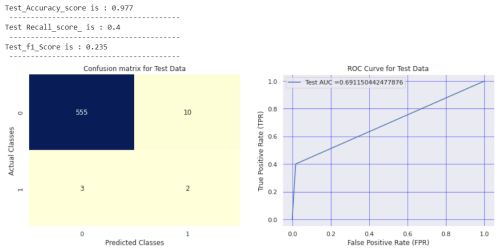
Here it can be clearly seen that the test_accuracy score and the f1_score are not very satisfactory. Although, the number of False Positives is 16 which is decent value but it could only classify 549 labels correctly out of 570. Moreover, the ROC curve is somewhat declined towards the normal Gaussian distribution axis which shows its bad performance in classification.

2. Logistic Regression:



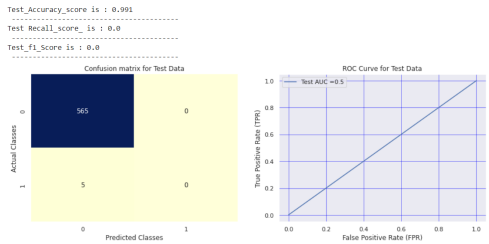
Here, the number of False positives is 23 while the number of true positives is 542 only. Logistic regression performs best in case of binary classification but here it's not predicting well. This might be because of undersampling of the test data and no linear correlation with the features of the test labels.

3. Decision Tree classifier:



Here as we can see, again the test accuracy is low as compared to the other models. Although, it can be seen that the f1-score is decent but not enough to become the best classifier. The number of True positives is 555 while the False Positives is 10. Moreover, the ROC curve looks satisfactory as it is close to the True positives axis.

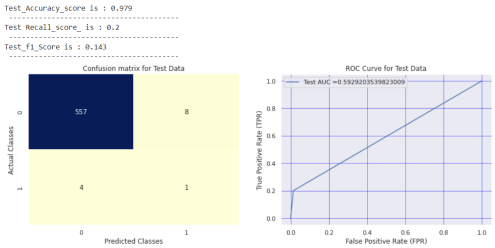
4. Random Forest Classifier:



Here, clearly it's the worst performer of all the classifiers as we can see that except the large quantity of True

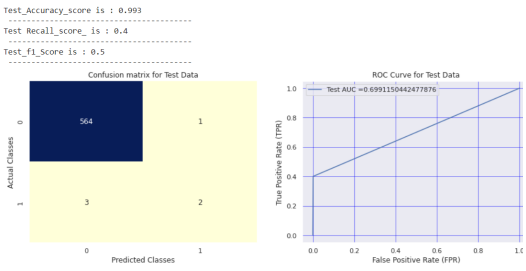
positives predicted by the classifier everything else indicates that this model cannot be used for classification. The ROC curve is almost similar to the normal axis which shows the poor classification of this model.

5. Support Vector Classifier:



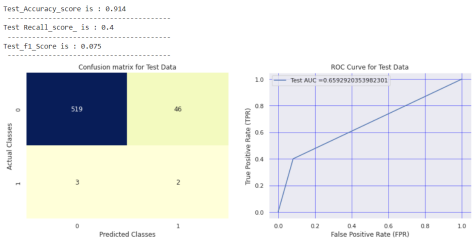
This algorithm is also a good contender for the best classification model in this project. It has a good accuracy score of 0.979 and f1-score of 0.143. The number of True positives is 557 and False Positives is 8. Also the ROC curve is very close to the True Positives Rate axis.SVM performs better in binary classification as it classifies data by finding the best hyperplane that separates all data points of one class from those of the other class.

6. kNN classifier:



The accuracy score of this model is 0.993 while the f-1 score is 0.4 which is the best compared to the other models. Moreover, the number of True Positives is 564 and False Positives is 1. The ROC curve is almost attached to the y-axis of the curve indicating the best performance of any of the classifiers.

7. Neural Networks (Deep Learning Technique:



It is clear from the above image that this model is not very predictive in classification techniques due to the less amount of dataset we used. It was initially mentioned that this model won't perform very well due to the less amount of training data.Still it got a decent accuracy of 0.914 while the f-1 score is 0.075.

The above results can be summarised in the following table:

Method	Accuracy	Recall	F1 Score
Naïve Bayes Algorithm	0.965	0.2	0.091
Logistic Regression	0.954	0.4	0.133

Decision Trees Classifier	0.977	0.4	0.235
Random Forest Classifier	0.991	0.0	0.0
Support Vector Classifier	0.979	0.2	0.143
***k*-NN Classifier**	**0.993**	**0.4**	**0.5**
Neural Network Classifier	0.914	0.4	0.075

From the above table it can easily be concluded that the best classifier that outperformed all the others was the kNN classifier. The second best classifier might be Decision Tree classifier due to its f-measure score and precision score. But the worst classifier among all of them is Random Forest Classifier. It was intuitive that random forest could not perform well on the testing dataset because of its complexity and latency.

VI. Discussion

Initially, it was observed that the dataset that we used for the training was **imbalanced.** Imbalanced data typically refers to a classification problem where the number of observations per class is not equally distributed; often you'll have a large amount of data/observations for one class (referred to as the majority class), and much fewer observations for one or more other classes (referred to as the minority classes). So, in the collected dataset, the number of stars having confirmed exoplanets were very less as compared to those who didn't have exoplanets revolving around them. It was absolutely necessary to balance the data for implementation of the models. To balance the data, oversampling was done after which almost the same amount of labels for each class were received. The process was also repeated for some of the classifiers namely Logistic Regression, Decision Trees and *k*NN classifier without oversampling the dataset. Although, we got good accuracies for the classifiers but as its clear that the performance of the classifier can't only be determined by their accuracies only. The precision, recall and the f-measure for each of the classifiers came out to be 0. Hence, this clearly indicated that the balancing must be done if decent results are required.

Moreover, it was also observed that the dataset was noisy and contained a significant amount of outliers. Hence during the data preprocessing the data was cleant and got rid of the outliers. The performance of various models improves on the test data if we analyse the comparison of the new models that were trained on balanced data to those trained on imbalance earlier.

Now, coming to the evaluation part, it is clear from the above table that the *k*NN

classifier outperforms all of the other classifiers when tested on the test data. When the kNN model was trained earlier on the imbalance data, the f-measure value on test data was literally 0%.But after oversampling, it improved upto 50%, which in this case is excellent result by the classifier.***k*-NN** is suitable than the rest because of its storing capability on the train data and classifying the test data based on the similarity.For a new instance (x), predictions are formed by scanning the whole training set for the K most similar instances (neighbours) and summarising the output variable for those K instances.This means that as new data emerges, it may be quickly sorted into a suitable category

Also, we can find a slight improvement in the prediction of positive points in test data for each model except **Random Forest.** This might be due to the fact that a large number of trees makes the algorithm slower and ineffective for real-time predictions.

VII. Conclusions

As seen from the table it was observed that the ***k*-NN** classifier had the highest accuracy, highest F1 score and the highest recall among all the classifiers. The neural network model (ie. deep learning) could not outperform the other classifiers and was the worst performer. Supervised learning methods proved to be more useful and accurate than this deep learning method.
The performance of each classifier observed was in the terms of runtime and accuracy achieved by each depended on the data volume (number of samples and features, our data had around 5k samples) and data quality (outliers, which we removed and imbalance data, which we balanced). In this project we predicted the presence of exoplanets using supervised learning techniques and deep learning techniques. The future work in this project includes the enhancement in the performance of the deep neural network techniques by using a larger dataset. As here, in this project the training dataset was only around 5k samples which might be due to less access to larger dataset of telescopes or space observatories, so neural networks typically could not perform well in just these small amounts of training dataset. The Deep Neural Network model could be applied on a larger dataset and this work can then be improvised to get even more accurate results.

VIII. References

[1] Catanzarite, Joseph H. “Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24.” 2015.
[2] Shallue, Christopher J., and Andrew Vanderburg. “Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around KEPLER-80and an Eight Planet around KEPLER-90.” vol. I, no. 10.3847/1538-3881/aa9e09, p. 23.