Word Count: 6369

Plagiarism Percentage 21%

Matches

1 **World Wide Web Match**
View Link

2 **World Wide Web Match**
View Link

3 **World Wide Web Match**
View Link

4 **World Wide Web Match**
View Link

5 **World Wide Web Match**
View Link

6 **World Wide Web Match**
View Link

7 **World Wide Web Match**
View Link

8 **World Wide Web Match**
View Link

9 **World Wide Web Match**
View Link

10 **World Wide Web Match**
View Link

11 **World Wide Web Match**

View Link

12 **World Wide Web Match**
View Link

13 **World Wide Web Match**
View Link

14 **World Wide Web Match**
View Link

15 **World Wide Web Match**
View Link

16 **World Wide Web Match**
View Link

17 **World Wide Web Match**
View Link

18 **World Wide Web Match**
View Link

19 **World Wide Web Match**
View Link

20 **World Wide Web Match**
View Link

21 **World Wide Web Match**
View Link

22 **World Wide Web Match**
View Link

23 **World Wide Web Match**
View Link

24 **World Wide Web Match**
View Link

25 **World Wide Web Match**
View Link

26 **World Wide Web Match**
View Link

27 **World Wide Web Match**
View Link

28 **World Wide Web Match**
View Link

29 **World Wide Web Match**
View Link

30 **World Wide Web Match**
View Link

Suspected Content

Analysis of Kepler Objects of Interest using Machine Learning for Exoplanet Identification Aditeya Baral

> Department of Computer Science PES University, Bangalore, India aditeya.baral @gmail.com Ameya Rajendra Bhamare Department of Computer Science PES University, Bangalore, India ameyarb1804 @gmail.com Saarthak Agarwal Department of Computer Science PES University, Bangalore, India saarthak1607 @gmail.com 9

Abstract—For several decades,

> planet identification has been a task performed by specialized astronomers and 1

domain experts. With the advent of computational methods and access to satellite data from space missions, this trend has changed. For instance, NASA's Exoplanet Exploration program has provided us with vast amounts of data on celestial objects to assist in space exploration. One such mission of interest is the Kepler mission. Over 3500

> transiting exoplanets have been identified since the commencement of the 2

mission in 2007. It's focus lay on exploring planets and planetary systems. It has provided us with

a catalog of discoveries that help in computing planet occurrence rates as a function of size, star type, insolation flux 2

and orbital period. This information is catalogued

in the Cumulative Kepler Object of Information table 1

available for public domain use on NASA’s exoplanet archive. Four basic models have been compared. Namely, Support Vector Machines, Random Forest Classifiers, AdaBoost and Deep Neural Networks. The

Random Forest Classifier was selected as the optimum machine learning model 1

and returned an F-1 score of 98% on the dataset. Index Terms—kepler, NASA, planetary systems, satellites, ex- oplanet discovery I. INTRODUCTION For ages, astronomy has fuelled

philosophical questions surrounding our existence and role in the universe. It is natural 1

human tendency to question if the celestial objects that we know of are planets or not considering the fact that we inhabit one - and may soon need to find another.

This question has been pondered upon and researched for centuries and the task of identifying 1

exoplanets, which are essentially planets outside of our solar system 28

has led to novel discoveries. It

has traditionally been a time-intensive task reserved for domain experts with access to specialized equipment. These experts 1

study images collected by satellite based telescopes like the Hubble. A

new generation of modern satellites such as Kepler 1

has opened the door for exoplanet exploration

with the goal of partially automating scientific observations and data generation pertaining to exoplanet identification. These satellites not only take pictures but also process those images using astronomical techniques to produce data with a

1

variety of features for identifying these exoplanets. This data has been made publicly available and democratized the once arduous task of exoplanet exploration among data scientists, engineers and statisticians alike.

Machine learning algorithms can be applied to exoplanet data in an attempt to detect overlooked exoplanets in data archives or automate the classification of objects of interest. The

1

models explored include Support Vector Machines, Random Forest Classification, AdaBoost and Deep Neural Networks. They were used to

classify data found in the Kepler Cumulative Object of Interest [1] (KCOI) table.

1

The KCOI table contains 50 features

pre-aggregated from Kepler data. This data has been cleaned to retain the

1

most important features, 19 to be specific. Finally, models have been fit on this data and eventually

used to assign a probability for an observation in the KCOI table being an exoplanet.

1

Traditional methods of researching images of distant stars and planets have changed. A digital transformation in astron- omy is underway.

1

As opposed to satellite based telescopes from the past, the

primary function of these machines is to collect and process a variety of data,

1

not just images. We now find

various state-of-the-art Machine Learning

9

techniques at our disposal to look for exoplanets. As this data has been made publicly available, this makes the task all the more easy. The

> Kepler space telescope which was launched by NASA in 2009. To date, it has been the most successful telescope **1**

when it comes to the discovery of exoplanets. It has identified several thousand objects of interest, with over 4300 of them confirmed exoplanets. The catalog has a high reliability rate, averaging 85-90% over the radius plane. It continues to improve as follow-up observations continue. While Kepler has been officially decommissioned as of October 2018 as it ran out of fuel, the

> statistical data it produced is expected to produce new exoplanet discoveries for years. **1**

Fig. 1. Comparison of Exoplanets Discovered II. BACKGROUND A. NASA Exoplanet Program The NASA Exoplanet Exploration Program (ExEP) under- takes missions to answer humankind's most timeless questions. These questions include the kind of planetary systems that orbit other stars in our galaxy, what exoplanets are like and whether humans are the only species to exist. The ultimate aim of these explorations is to

> be able to discover and characterize Earth-like **17**

planets, search for habitable conditions on those planets, and reveal signatures of life. For instance,

> Kepler-22b was the first exoplanet considered to contain the ingredients needed to support life as we know it. **1**

B. The Kepler Mission The

> Kepler space telescope was launched by NASA in 2009. It has been the most successful telescope in the discovery of exoplanets. **1**

The

> mission is specifically designed to survey our region of the Milky Way galaxy. It discovers hundreds of Earth-size and smaller planets in or near the habitable zone. It additionally determines the fraction of the hundreds of billions of stars in our galaxy that might have such planets. **5**

A dip in a star's brightness could indicate one of its planets is passing between the star and the observing telescope. Transit time is the time it takes for the planet to pass between the start and observing telescope. It is measured in hours. The 1

magnitude of reduction in brightness coupled with transit time can provide mathematical clues to the relative size and position of the planet relative to its star. 1

The collected data is periodically released and is hosted by CalTech under contract with NASA. The

satellite was officially retired in October 2018 as it ran out of fuel. Though it was officially decommissioned, the statistical data that Kepler produced continues to produce new exoplanet discoveries. 1

C. Exoplanet Identification Techniques Microlensing

measures the bending of light as energy from a star passes a planet 1

making it

an indirect method of planet detection. It has the 1

ability to detect the smallest and most distant of planets. 1

The

shift of a star resulting from the gravitational pull of the planets orbiting 1

it is measured using the

Radial Velocity. It is positively correlated to both, the mass and orbital period of a planet. 1

Fig. 2. Number of Planets by Discovery Method Direct imaging is an age old method used to identify exoplanets. It involves using

extra-terrestrial telescopes to capture high resolution pictures of star fields. The pictures are analysed by scientists to determine if planets exist. While good for detecting stars, this method has proven to be inadequate for exoplanet identification. 1

A solar eclipse occurs

when the moon passes directly in front of the sun, blocking its light. This is quite similar to how the transit method finds exoplanets. When a planet passes between an observer and a star, it blocks some of that star's light. 6

The star gets dimmer briefly. Though it is a small change in brightness, it hints to astronomers

the presence of an exoplanet around a distant star. 26

Planetary transits furnish information which leads

to an estimate of the object of interest's size, mass, speed and period of orbit and density of 1

its star. Traditional methods like direct imaging and radial velocity techniques have been replaced. The reason being that they

are biased towards the detection of large exoplanets. In contrast, Kepler's transit time data allows for the detection of smaller Earth-sized exoplanets. 1

This

has opened up a new window of 17

planets to be discovered. III. PREVIOUS WORK Natalie M. Batalha, in her paper titled 'Exploring exo- planet

populations with NASA's Kepler Mission 17

reports

on the progress Kepler has made in measuring the prevalence of exoplanets orbiting within 1 AU of their host stars. This is in support of NASA's long-term goal 8

> of finding habitable environments beyond the solar system.

She talks about

> catalog reliability, planet confirmation and characterization and the requirements for reliable planet occurrence rates. **21**

Another paper titled

> ‘Advances in exoplanet science from Kepler’ by Jack J. Lissauer et. al **24**

discusses the highlights of the Kepler mission. While the two papers discuss the highlights of the mission, they fail to talk about any exoplanet detection methods.

> ‘Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90’ by Christopher J. Shallue and Andrew Vanderburg **11**

presents

> a method for classifying potential planet signals using deep learning. **25**

Convolutional Neural Networks (CNNs) are used

> to predict whether a given signal is a transiting exoplanet or a false positive caused by either astrophysical or instrumental phenomena. The model **2**

> ranks plausible planet signals higher than false positive signals on the test set **2**

98.8% of the time. They

> statistically validate two new planets that are identified with high confidence by the model. **2**

This paper explores deep learning approaches but does not compare its performance with other machine learning models. Miguel Jara-Maldonado et. al, in their paper ‘Transiting Exoplanet Discovery Using Machine Learning Techniques’

propose a model to create synthetic datasets of light curves. The performance of several machine learning models is used to identify transit exoplanets. 4

They have conclude that multires- olution

analysis in the time-frequency domain can improve exoplanet signal identification, because of the characteristics of light curves and transiting exoplanet signals. 4

The

use of machine learning algorithms is demonstrated to a 10

high degree.

Specifically, a Gaussian process classifier reinforced by other models is used to perform probabilistic planet validation by incorporating prior probabilities for possible false positive scenarios. This paper identifies the 10

caveats against the use of single-method planet validation techniques and cautions against it. In

‘False Positive Probabilities for all Kepler Objects of In- terest: 1284 newly validated and 428 likely false positives’ by Timothy Morton et. al, astrophysical 15

false positive probability calculations for every Kepler Object of Interest 2

are presented. It was

the first large-scale demonstration of a fully automated transiting planet validation procedure. 19

IV. PROPOSED SYSTEM

There are three primary categories of machine learning applications - regression, clustering, and classification. Clas- sification is a supervised machine learning technique where observations are assigned a known class value based upon their 1

explanatory or dependent variables. The classes can either be binary, or even multi-class.

Classification is

one of the major applications of machine learning 29

algorithms and is used in everyday life to solve large scale real world problems across various domains.

This study focuses on a binary classification of objects of interest as “FALSE POSITIVE” or “CONFIRMED” exoplanets. The classification of “FALSE POSITIVE” is used by NASA to indicate the satellite incorrectly tracked an object of interest. 1

We do not consider the observations labelled as “CANDIDATE” since these are yet to be labelled by NASA and hence, are unknown to us. For our analysis, we have used four models, each with its own unique characteristics to tackle the problem at hand from different angles. The four models used are Support Vector Machines (SVM), Random Forest, AdaBoost and a Feed-Forward Neural Network. A.

Support Vector Machine Support Vector Machine is a machine learning algorithm 22

which seeks to identify a hyperplane between classes and attempts to maximise the

distance between the hyperplane and points in the opposing classes. 1

It also possesses a property, known as the ”kernel trick”. It is able to tackle the issue of classes being non-linearly separable by

map the input feature space to a higher dimensional feature space 16

by performing various linear and non-linear transformations. It then uses a maximal distance hyperplane in the transformed space to separate the classes. B.

Random Forest Random forest classification consists of a large number of individual decision trees that operate as an ensemble. Each 7

of these trees

spits out a class prediction and the class with the most votes becomes our model’s prediction, like an election of 7

sorts. They

utilize bagging and randomized feature selection to build the ensemble of decision trees. Bagging 1

results in only a portion of the dataset being utilized in any single decision tree in the ensemble. 1

Randomized feature selection involves selecting

a subset of the features from the dataset randomly to construct individual decision trees as part of the ensemble. This helps to generalize the model as the effects of correlation and overfitting can be minimized by 1

randomization. The final

classification is obtained by combining the results of each tree to reach a decision. 1

C. AdaBoost AdaBoost, short for “Adaptive Boosting” is an ensemble learning method. It

uses an iterative approach to learn from the mistakes of weak classifiers and turns them into strong ones. We 3

define

a weak classifier as one that performs better than random guessing, but still performs poorly at designating classes to objects. A single classifier may 3

not be able to accu- rately predict the class of an object. 3

However, by combining multiple weak classifiers in a way that

each one progressively learns from the others’ wrongly classified objects, we 3

are left with a strong model. These classifiers could be any of the classifiers that we know of, Decision Trees

being the default. D. Neural Network Put quite simply, a

neural network is an attempt to sim- ulate the network of neurons that make up a human brain. Computers are able to learn things and make decisions in a human

13

like manner by doing so. These are genetically inspired algorithms

created by programming regular computers to behave as though they are interconnected brain cells. A neural

18

network is capable of generalising well and is known to be a Universal Function Approximater and can easily learn non-linear complex relationships between the dependent and independent variables. E. Ensemble Learning To complement the already present algorithms, we also propose an ensemble of these algorithms by combining them together in a sequentially. The final predictions obtained will be a majority vote across its individual classifier predictions. This architecture helps V. WORKFLOW The Cumulative Kepler Object of Information Dataset is not without faults. The dataset is a raw one, obtained directly from the readings measured by Kepler after assigning labels, and hence has not only missing values but a lot of unnecessary columns as well. Some of these can be dropped since they are ID attributes assigned but a majority of them are crucial to our analysis need to be filled in appropriately. This step helps us tune our model's performance and remove redundant features which do not add any information to our analysis. A. Data Preprocessing We begin by first observing the columns that have missing values. We observe that though quite a lot of these columns do contain missing values, most of these missing values correspond to the errors associated with attribute values. We also observe that two error attributes are filled with missing values - the positive and negative errors associated with Equilibrium Temperature. Hence we decide to drop them completely, since their values cannot be imputed. For Attribute Number of Missing Values kepler name koi depth 7270 363 koi depth err1 454 koi depth err2 454 koi duration err1 454 koi duration err2 454 koi impact 363 koi impact err1 454 koi impact err2 454 koi insol 321 koi insol err1 321 koi insol err2 koi kepmag 321 1 koi model snr 363 koi period err1 454 koi period err2 454 koi prad 363 koi prad err1 363 koi prad err2 363 koi score 1510 koi slogg 363 koi slogg err1 468 koi slogg err2 koi srad 363 468 koi srad err1 468 koi srad err2 468 koi steff 363 koi steff err1 468 koi steff err2 483 koi tce delivname 346 koi tce plnt num 346 koi teq 363 koi teq err1 9564 koi time0bk err1 koi teq err2 9564 454 koi time0bk err2 454 TABLE I COUNT OF MISSING VALUES IN ATTRIBUTES the remaining error attributes, we observe a highly skewed distribution of a few attributes. It would be ill-advised to replace these values with their average, and hence we decide to fill up their values with the median error value. This handles the case for near normally distributed data, as well as skewed data. The attribute kepler name, koi tce delivname and koi tce plnt num are dropped from our analysis. These are ID attributes which have been assigned to the objects of interest after analysis has been performed and they have been labelled (into exoplanets and non-exoplanets), and hence do not play a part in the building of the classification model. These attributes are thus removed from the analysis. We finalise the data preprocessing step by standardising our data, to ensure all attributes are on the same scale. This increases performance and also decreases computational effort required by the models, thus speeding up the process of classification. Additionally, it is to be noted that since the majority of the error values have a low order of magnitude, we propose to build two separate variations of models - one considering these error metrics and one without. This helps us analyse the effect of these attributes. Fig. 3. Right Ascension of Exoplanets showing near Normal Distribution Fig. 4. Equilibrium Temperature of Exoplanets showing

Skewed Distribution Attribute Description dec ABC koi depth ABC koi duration ABC koi fpflag co ABC koi fpflag ec koi fpflag nt ABC ABC koi fpflag ss ABC koi impact ABC koi insol ABC koi kepmag ABC koi model snr ABC koi period ABC koi prad ABC koi slogg ABC koi srad ABC koi steff koi teq ABC ABC koi time0bk ABC ra ABC TABLE II ATTRIBUTES CHOSEN FOR ANALYSIS B. Feature Correlation and Variance To reduce redundancy in the number of attributes chosen, we study the correlation between the attributes and observe the results. We see that while there is a correlation present between the attributes picked in our study, most of the correlations are not of a high order. Fig. 5. Pearson Correlation without Error Attributes Fig. 6. Pearson Correlation with Error Attributes Additionally, only a few attributes exhibit high correlation with the dependent variable. Most attributes are observed to have only a medium or low correlation. We also observe that most attributes show a negative correlation with the dependent variable, suggesting that lower values of these attributes correspond to higher probabilities of an object being an exoplanet. To observe the variance added to the dataset by the at- tributes, we perform Principal Component Analysis (PCA) on our dataset and

choose the number of principal components as the **12**

number of chosen attributes. We observe that a full 100% variance is reached only after taking into account every single attribute or component and the first principal component explains just 15% of the overall variance. The Fig. 7. Correlation of Non-Error Attributes with Dependent Variable Fig. 8. Correlation of Attributes with Dependent Variable curve is a smooth decline, thus showing that each component individually adds significant variance to the dataset. We hence decide to move on with our set of attributes without removing any of them. However, this changes when we consider the error attributes as well, as we observe that the first 30 components contribute to a full 100% of the variance, out of the complete set of 39 attributes. For this case, we consider only the first 30 principal components for our analysis and thus remove the remaining components. This step allows us to retain information from all the attributes while reducing the number of features as well. Final observations also indicate that the spread of values for non-exoplanets is far greater than the spread of values for confirmed exoplanets. This shows that the magnitude of attributes for exoplanets is lesser compared to other objects of interest. C. Evaluation Metrics and Cross Validation To counter the imbalance of the dataset, we propose differ- ent evaluation metrics, which take in account the imbalance. Fig. 9. PCA Decomposition Fig. 10. Variance Distribution without Error Attributes Fig. 11. Variance Distribution with Error Attributes Fig. 12. Radial Visualisation without Error Attributes Fig. 13. Radial Visualisation with Error Attributes These include the F1 Score, Cohen Kappa score, Balanced Accuracy Score and finally the Confusion Matrix. Additionally, to test out our classifier on different sets, we use F-Fold cross-validation across our entire dataset to ensure that we are not underfitting our classifier by introducing high bias. Since the dataset is imbalanced, we again use both - a non-stratified as well as a stratified splitting to ensure that within each fold the number of positive and negative examples are equal. We measure our classifier’s performance across each split and finally take the mean of the performance achieved. D. Classification Models Preliminary visualisations showed us that the dataset is imbalanced .i.e the distribution of examples for each class is skewed towards the negative class. This owes to the abundance of non-exoplanet bodies in the universe compared to actual exoplanets. There is also an abundance of noise in the dataset, with inter-mixed classes present. To observe how different classifiers tackle this issue, we propose to use classifiers working on different approaches - traditional classifiers like the SVM, Bagging Ensemble models like the RandomForest, TABLE III GRIDSEARCH PARAMETERS FOR SVM Attribute C 1, 1.05, 1.10, 1.15, 1.20, 1.25, 1.35, Possible Values 1.40, 1.45, 1.55, 1.65, 1.75, 1.85, 1.90, 1.95, 2, 2.05, 2.10, 2.15, 2.20, 2.25, 2.30, 2.35, 2.40, 2.45, 2.50, 2.55, 2.60, 2.65, 2.70, 2.75, 2.80,

2.85, 2.90, 2.95, 3 gamma scale, auto shrinking True, False tol 0.001, 0.01, 0.1, 1, 10, 100, 1000 class weight None, balanced TABLE IV CHOSEN GRIDSEARCH PARAMETERS FOR SVM Attribute Chosen value C 1.8500000000000008 Fig. 14. Distribution of Classes class weight None gamma scale shrinking true Boosting Classifiers like the AdaBoost and finally a Feed- tol 1 Forward Neural Network. Additionally, we also create an class weight None, balanced Ensemble majority vote classifier that combines all of these together to take the majority vote and return the predicted label.

input feature space to a higher dimensional feature space **16**

by performing a Gaussian transformation. We then tune our model by performing cross-validated GridSearch on it, allowing it to run through multiple combinations of parameters and finally choosing the best model based on its performance. 2) Random Forest: Our second classifier is an Ensemble based Bagging Classifier based on the Decision Tree. The RandomForest works by training multiple individual Decision Trees on each subset of the dataset and finally averaging the result obtained on classifying the input test example. Each Decision Tree in the Random Forest is alike, and can be tuned based on its hyperparameters. The Random Forest is extremely powerful and hence requires less fine tuning to build an accurate model. We again deploy GridSearch to tune each Decision Tree's hyperparameters to choose the most optimum set which returns the best results. Fig. 15. Distribution of Class Labels TABLE V The attributes being considered in our study include all GRIDSEARCH PARAMETERS FOR RANDOM FOREST attributes except the ID attributes which assign names and ID's Attribute Possible Values to each object of interest, along with any other attributes which num estimators 100, 200, 300, 400, 500, 600 are assigned by NASA after studies have been performed on

max depth 1, 2, 3 , 4, 5, 6, 7, 8, 9 min samples leaf 0, **20**

0.2, 0.4, 0.6, 0.8, 1.0 these data points. We add to the aforementioned classifiers by max features None, sqrt, log2 training our model using two sets of considered attributes - class weight None, balanced, balanced subsample one with and without the error attributes present. Finally, we perform GridSearch on each classifier with different values for its hyperparameters to choose the optimum set of values that TABLE VI CHOSEN GRIDSEARCH PARAMETERS FOR RANDOM FOREST provide maximum performance. 1) Support Vector Machine: We deploy a Support Vector Attribute Chosen value Machine (SVM) as our first classifier. We additionally use num estimators 100 criterion gini a Gaussian or an RBF kernel to tackle the issue of our max depth None classes being non-linearly separable. This kernel helps map the min samples 2 TABLE VII GRIDSEARCH PARAMETERS FOR ADABOOST No. of estimators Attribute 70, 80, 90, 100, 110, 120 Possible Values learning rate 0.8, 0.85, 0.9, 0.95, 1, 1.05, 1.1, 1.15 algorithm SAMME, SAMME.R TABLE VIII CHOSEN GRIDSEARCH PARAMETERS FOR ADABOOST Attribute Chosen value Algorithm SAMME base estimator DecisionTreeClassifier(max depth=3) learning rate 0.95 n estimators 80 3) AdaBoost: AdaBoost is the third classifier we use. AdaBoost is a powerful Ensemble Boosting algorithm which also uses multiple Decision Trees, similar to Random Forest. However, instead of training each Decision Tree on a subset of the training data, each Tree is individually and sequentially trained on the entire dataset, such that the "harder examples" (which consist of the examples predicted wrongly) are given larger importance. This is done by assigning weights to each training example, and modifying these weights based on the current classifier's predictions. A corresponding weight for an example is increased for the next classifier if the current classifier predicts it incorrectly and vice versa. However this also makes it highly

prone to overfitting, and hence we choose the average out the results using cross-validation. We tune AdaBoost by using the following parameters. Since AdaBoost has its own hyperparameters (apart from the Decision Tree hyperparameters), we focus on tuning the AdaBoost classifier itself, and only vary the depth of the Decision Trees used. 4) Neural Network: As our final model, we train a regular Feed-Forward Neural Network architecture to perform classifi- cations. A neural network is capable of generalising well and is known to be a Universal Function Approximater and can easily learn non-linear complex relationships

between the dependent and independent variables. Considering the distribution of 12

our data's classes, this choice is justified. To tune our neural network, we vary the number of layers as well as the neurons per layer. However, we fix our activation functions for all layers except the final output layer to be ReLu and the final layer to be sigmoid since we are performing binary classification. Finally, we choose Adam as the optimizer and Binary Cross Entropy as the Loss Function. TABLE IX HYPERPARAMATERS FOR NEURAL NETWORK Hyperparamter Chosen

Optimizer adam Loss binary crossentropy Metrics accuracy Epochs 20 27

Fig. 16. Feed-Forward Neural Network Architecture E. Analysis of Feature Importance Different machine learning algorithms work differently. They all differ in the method used to classify, the sampling as well as initialisation of variables used for modelling. This also leads to differences in the features being analysed, and the priority or importance given to each feature while performing a prediction or classification. We analyse the differences in the feature importance by performing a recursive elimination of features and observing the features that provide maximum changes to the model's performance. These features are then plotted in order of their relevancy to show how each model ranks the features. This however, is not possible with the Support Vector Machine, since the features are transformed into a higher dimensional feature space. The Random Forest classifier assigns the highest importance to flag variables namely the Centroid Offset flag and Non- Transit like Flag. We also observe a steep decline in the importance of attributes, beyond the set of flag attributes thus suggesting that the Random Forest classifier is extremely powerful while working with categorical variables and uses them to make the core decisions while splitting the dataset into decision nodes. This also proves that categorical variables help perform classifications faster since unlike continuous variables, they do not have to be binned. The AdaBoost classifier displays a different order of feature importance, even though it uses a Decision Tree, just like the Random Forest classifer. Flag variables although exhibiting a high level of importance, do not take the top spot. This is accounted by the difference between Boosting and Bagging. Unlike Random Forest, we also observe that the feature importance tends to show a pattern, with the magnitude of importance taking a discrete set of proportions. Additionally, Fig. 17. Feature Importance for AdaBoost Classifier we also observe that the AdaBoost classifier can assign an importance of zero to any feature. Fig. 18. Feature Importance for Random Forest The Neural Network again returns a completely different order of ranked features. However, we observe that the flag attributes are ranked the highest, similar to the Random Forest classifier. This confidently shows that in classification of obser- vations, categorical variables drive the process of making well- informed decisions leading to accurate predictions. Categorical variables help form decisions faster, and are quick to decrease the overall entropy in a set of observations regardless of the algorithm deployed. Additionally, because each algorithm ranks features differ- ently, there is a possibility of combining together all these

algorithms sequentially since an example classified wrongly by one algorithm may be classified correctly by another. This will also average out the feature importance for low scoring attributes while retaining the high scores for important ones. This leads to our majority-vote ensemble classifier, which performs a majority vote among its models to classify observations. Fig. 19. Feature Importance for Neural Network Model Stratified F-1 Score Non-Stratified F-1 Score SVM 97.72 97.66 RandomForest 98.05% 98% AdaBoost 98.03% 98.05% Neural Network 97.78% 94.46% Majority - Vote Ensemble 98.36% TABLE X 98.29% MODEL PERFORMANCE WITHOUT ERROR ATTRIBUTES VI. RESULTS We propose to test using 10-Fold cross-validation on the entire set of observations. Cross-validation helps analyse the model's performance against each subset of the dataset by training the model on n-1 folds and testing the model on the nth fold. Cross-validation also creates stratified splits, thus creating a pipeline to estimate the model's accuracy even on imbalanced datasets. Each of the aforementioned models are subjected to cross-validation using both stratified as well as non-stratified splits. Additionally, error attributes are also considered to build additional models. We use the F-1 Score to mark a model's performance, since the dataset is imbalanced and a plain accuracy score would paint the wrong picture. We observe that on a dataset without the error attributes, all the models work really well. However, in terms of numbers the Random Forest classifier obtains

majority vote ensemble model outperforms the rest by a good margin, attaining an F- 1 score of 98.3%. This shows that models which rank features differently can be combined together sequentially to improve overall performance. On considering the error attributes as well, it is observed that the performance increases. This is because PCA helps us maximise variance and retain an uncorrelated and independent feature set. In this case as well, we see that the Random Forest outperforms the others with a greater F-1 score. It even beats its own F-1 score achieved on removing the error attributes. The majority vote ensemble model again performs well, but fails to achieve the same scores as done previously. Model SVM RandomForest AdaBoost Neural Network Majority - Vote Ensemble Stratified F-1 Score 98.04 98.28% 96.65% 98.16% 98.29% TABLE XI Non-Stratified F-1 Score 98.04 98.30% 96.27% 98.27% 98.32% number of clusters. At this point, both the value of the SSE as well as the number of clusters is balanced to neither underfit nor overfit the set of observations. MODEL PERFORMANCE WITH ERROR ATTRIBUTES We can conclude that the models built with the Error attributes tend to do better than the models built after remov- ing the error attributes. Although all models perform almost equally well, the Random Forest classifier outperforms the rest and also performs well across all cases, while the majority vote ensemble model is consistent as well. VII. OBSERVATIONS There are many criteria that exist which are used to group planets. Planets are grouped based on their mass, orbital range and composition to name a few. Planets which are part of the same group or cluster often exhibit similar properties as its neighbours and are used to classify or group newer planets. Over the years, scientists have tried to analyse readings to group together such planets, in an effort to make analysis of these heavenly bodies simpler and organised. We can today categorise the different exoplanets into 4 broad categories. 1) Gas Giant - similar to Jupyter or Saturn 2) Neptune Like - similar to Uranus or Neptune 3) Super Earth - more massive than Earth but lighter than Neptune 4) Terrestrial - Rocky and Earth-like In recent times to make study of these exoplanets easier, these have been further separated into a total of 8 categories. Since these additional categories are branched off from the main ones, there is quite a bit of overlap between them. The 8 categories are Gas Giant, Mesoplanet, Mini Neptune, Planemo, Planetar, Super Jupyter, Super Earth and Sub Earth. We can analyse the distribution of the exoplanets in the dataset using a cluster analysis. This will allow us to visualise the differences in

characteristics between these exoplanets as well as observe their distribution into categories. A.

K-Means Clustering K-Means clustering is an iterative clustering algorithm 23

used to group together similar objects and identify patterns. Clusters are formed by assigning and recalculating centroids iteratively until convergence. These centroids are highly representative of a cluster, and are often used to represent characteristics for the entire group as well. K-Means however requires the number of clusters (or cat- egories) in this case to be fed into the model as a hyperpa- rameter.

There are different methods available to find out the 14

optimum number of clusters, the most common being the Sum of Squares Error (SSE). This method calculates the SSE with every batch of clustering across various number of clusters and plots them. A sharp bend in this curve marks the optimal Fig. 20. SSE Method to Find Optimal Number of Clusters We observe that the optimal number of clusters returned is 7 instead of 8. This is because of the noise present in the set of observations in the form of overlaps between the classes. A visualisation of the distribution shows us that there are 4 primary groups of planets present in the dataset, while more groups have been constructed by breaking down one of the classes. Fig. 21. Inter Cluster Distances There is also an uneven distribution of exoplanets present. We see that the sum of squares error for each cluster is not uniform. Although most clusters have a low error suggesting that those groups consist of planets that share common prop- erties and are very similar to each other, the largest cluster also has the highest sum of squares error, suggesting that a majority of exoplanets have properties vastly different from the other exoplanets, but are themselves not very similar to each other. Fig. 22.

Within Cluster Sum of Squares Error 12

Fig. 24.

Within Cluster Sum of Squares Error 12

B.

Agglomerative Hierarchical Clustering Agglomerative Hierarchical Clustering 14

is another cluster- ing algorithm that builds clusters by iteratively combining observations

similar to each other into a single cluster, 14

until there is only one cluster left. Unlike K-Means, Hierarchical Clustering does not need the number of clusters to group together observations. We can obtain the desired number of clusters needed from

dendogram, which is a hierarchical tree storing the order in which the clusters were formed. Fig. 23. Dendogram Formation However, we can still find the optimal number of clusters recommended using a similar approach. We again obtain the optimal number as 7, thus proving without doubt about the overlap between the classes. C. Analysis of Characteristics of Grouped Exoplanets We observe how the data points are distributed in a 2- dimensional space by colouring in the visualisation performed earlier using PCA. We see that a vast majority of exoplanets are clustered together to show common characteristics. This also further confirms our hypothesis of overlapping classes present in our dataset. Fig. 25. Clustering of Observations in 2-Dimensional Space However, we also notice that some attributes are vastly different across classes compared to the remaining attributes such as Insolation Flux and Equilibrium Temperature, while some attributes are very similar across classes such as the Right Ascension and the Kepler Band Magnitude. This shows that there is no significant trend between attributes themselves, since exoplanets of different sizes and compositions, can have a similar set of attributes. This makes it increasingly difficult for researchers and scientists to pinpoint planets and declare them habitable because of the variation in characteristics. [2] [1] [3] [4] [5] [6] [7] [8] REFERENCES [1] CalTech, “Cumulative koi data,” https://exoplanetarchive.ipac.caltech.edu/cgi- bin/TblView/nph-tblView?app=ExoTblsconfig=cumulative, 2020. [2] S. T. B. Timothy D. Morton, “False positive probabilities for all kepler objects of interest: 1284 newly validated planets and 428 likely false positives,” 2016. [Online]. Available: https://iopscience.iop.org/article/10.3847/0004-637X/822/2/86/meta Attribute Standard Deviation between Classes koi prad ra 2.468743 4.966585 koi depth 8428.5702 koi steff 852.041002 koi model snr 762.228749 koi period 7.060858 koi insol 11604.505149 koi kepmag 1.200468 koi time0bk 17.928517 koi fpflag ss 0.188982 koi impact 0.171315 koi duration koi teq 1013.912337 1.166409 koi slogg 0.258292 dec 1.031913 koi srad 0.667931 TABLE XII STANDARD DEVIATION CLASSES PER ATTRIBUTE [3] M. J.-M. Vicente Alarcon-Aquino, “Transiting exoplanet discovery using machine learning techniques,” 2020. [Online]. Available: shorturl.at/mG069 [4] A. V. Christopher J. Shallue, “Identifying exoplanets with deep learning: A five planet resonant chain around kepler-80 and an eighth planet around kepler-90,” 2018. [Online]. Available: https://arxiv.org/abs/1712.05044 [5] S. T. Jack J Lissauer, Rebekah I Dawson, “Advances in exoplanet science from kepler,” 2014. [Online]. Available: https://www.nature.com/nature/about [6] N. M. Batalha, “Exploring exoplanet populations with nasa’s kepler mission,” https://www.pnas.org/content/111/35/12647, 2014. [7] L. P. Kyle A. Pearson, “Searching for exoplanets using artificial intelli- gence.” [8] T. D. David J Armstrong, Jevgenij Gamper, “Exoplanet validation with machine learning: 50 new validated kepler planets.”