EXOPLANETS-DETECTION-USING-RANDOM-FOREST

A special thank you to my professors for their invaluable guidance and support.

- **Dr Ken McGarry**
- **Dr Basel Barakat**
- **Dr David Nelson**

YOUSSEF IBOURK

INTRODUCTION:

In my previous project, I had explored and gained knowledge about anomaly detection in the astronomy field. Now, I am taking the next step and delving deeper into the subject matter. I am expanding upon my previous research and work by developing an “R” code that extracts the anomalies. This new study will be a continuation of my previous work and will provide more insights and understanding of the topic.

In my first part of the project, I discussed the Kepler telescope and its mission to discover Earth-size and smaller planets in or near the habitable zone of other stars.

It uses the transit method to detect exoplanets, which involves measuring the small dip in a star’s brightness that occurs when a planet passes in front of it, imagine that you are watching a movie from a projector. If a person passes through the projector beam, it creates a shadow, blocking some of the light. Viewers see that part of the light from the projector is being blocked, and they realize that someone is blocking it. Viewers in this case repeat the actions of the Kepler telescope, which detects planets, the smallest changes in the amount of light coming from the star if the planet passes in front of the star.

DATASET :

The dataset used is the cumulative dataset collected by the Kepler telescope, it contains data on exoplanet candidates and their properties, which is a NASA-funded mission that was launched in 2009 with the goal of discovering and characterizing exoplanets.

The dataset includes information on various properties of the exoplanet candidates such as orbital period, impact parameter, duration, depth, equilibrium temperature, insolation flux, surface gravity, stellar radius, and effective temperature. It also includes a label indicating whether the candidate is a confirmed exoplanet or just a candidate.

ALGORITHM USED :

Exoplanet discovery can be made more accurate and effective with the use of data mining techniques because of its ability to make analysis of the vast amount of data gathered by the telescope and the discovery of patterns and connections between various properties that are suggestive of exoplanets.

Additionally, data mining allows for the application of advanced machine learning algorithms, such as random forest, which can improve the accuracy and efficiency of exoplanet detection. Data mining also allows for the identification of important features in the data, and the ability to handle large amount of missing or incomplete data which can help to improve the results and make the predictions more robust.

In the case of exoplanet detection with the Kepler telescope, we will use “random-forest” to analyze the large amount of data collected by the telescope. The random forest algorithm can also be used to estimate the relative importance of different features in the data, which can help identify which features are most relevant for exoplanet detection.

ALGORITHM OVERVIEW :

I will start by loading the appropriate and necessary libraries and then reading in the csv file containing the data. I will then use the table function to count the number of each class in the “koi_disposition” and “koi_pdisposition” columns. Counting the number of occurrences of each class is important to understand the class distribution of the data and can be useful in understanding the overall balance of the data, so that it does not affect the performance of our random forest algorithm, which is why it is important to understand the class distribution in the data.

Next, I will need to know whether an observation is a confirmed or candidate exoplanet. I will then have to split the data into a training set and a testing set to evaluate the performance of the model. The training set is used to train the model and learn the patterns. The testing set is then used to evaluate the performance of the model. All of that, in order to determine how well the model is able to generalize and make predictions on new data.

Next, I will use the random forest model using the training set, putting an “exoplanet” column as the response variable and the other columns as the predictor variables. I will then extract feature importance from the model. And then use the trained model to make predictions on the test set and calculate the accuracy of the predictions. At the end, I will have to extract the exoplanet observations from the test set and identify the anomalies. Finally, I will have to visualise the important sections of the script and most importantly visualise the anomalies.

CODE EXPLANATION :

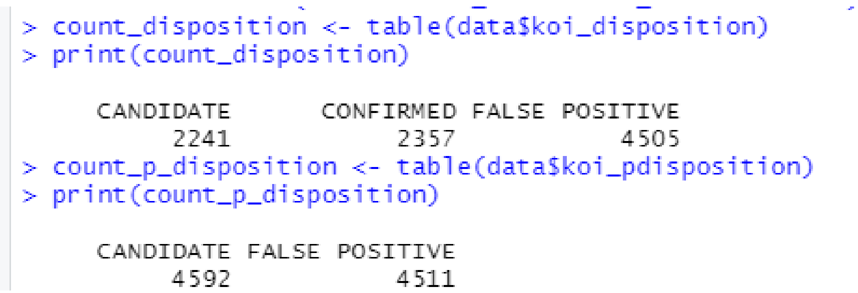
Now that I have explained the algorithm used in this project, I will delve deeper into the technical details by describing the code and the output generated by the implementation of the algorithm. This includes a step-by-step breakdown of the code and an explanation of the results obtained from running the code, including any visualizations or plots generated as a result. This will provide a more in-depth understanding of how the algorithm was implemented and the insights that were gained from the analysis.

Since I will be using Random Forest algorithm to classify exoplanets and detect anomalies in the data. I should start by loading the required libraries and reading in the data.

- library(randomForest): The random forest algorithm works by aggregating predictions made by several decision trees each decision tree is trained on a subset of the dataset, called a self-training dataset.

- library(caret): Classification And Regression Training, is designed for combining training and forecasting models.

It is useful to check the distribution of data that is why we do this pre-processing step to check the data before applying any machine learning model.



```
> count_disposition <- table(data$koi_disposition)
> print(count_disposition)

     CANDIDATE      CONFIRMED FALSE POSITIVE 
          2241           2357           4505 
> count_p_disposition <- table(data$koi_pdisposition)
> print(count_p_disposition)

     CANDIDATE FALSE POSITIVE 
          4592           4511 
```

I then create a new column in the “data” dataframe called “exoplanet” and assigning a value of 1 if the value of “koi_disposition” is ==”CONFIRMED” or 0 else, to each observation.

We then create a partition of the data into a training set and a test set. We used the createDataPartition() function to randomly select 80% of the rows of the “data” data frame and assign the row indices to the variable “split_index”.

After that, we create a training set by subsetting the data based on the split index that was previously generated by “createDataPartition()” function. And the reverse for the testing set.

Now we reach one of the most important lines of our code that serves for training a random forest model using the “koi_period”, “koi_impact”, “koi_duration”, “koi_depth”, “koi_teq”, “koi_insol”, “koi_slogg”, “koi_srad”, and “koi_steff” columns as predictors and the “exoplanet” column as the response variable.

We calculate then and assign the feature importances of the predictors used in the random forest model stored in the variable “randomf_model” to the variable “feature_importance”

Following this, we create a new data frame called “feature_importance_df” which will contain the feature importances of the predictors in the random forest model. The “feature” column of this data frame contains the names of the predictors, which are obtained by using the row.names() function on the “feature_importance” variable.

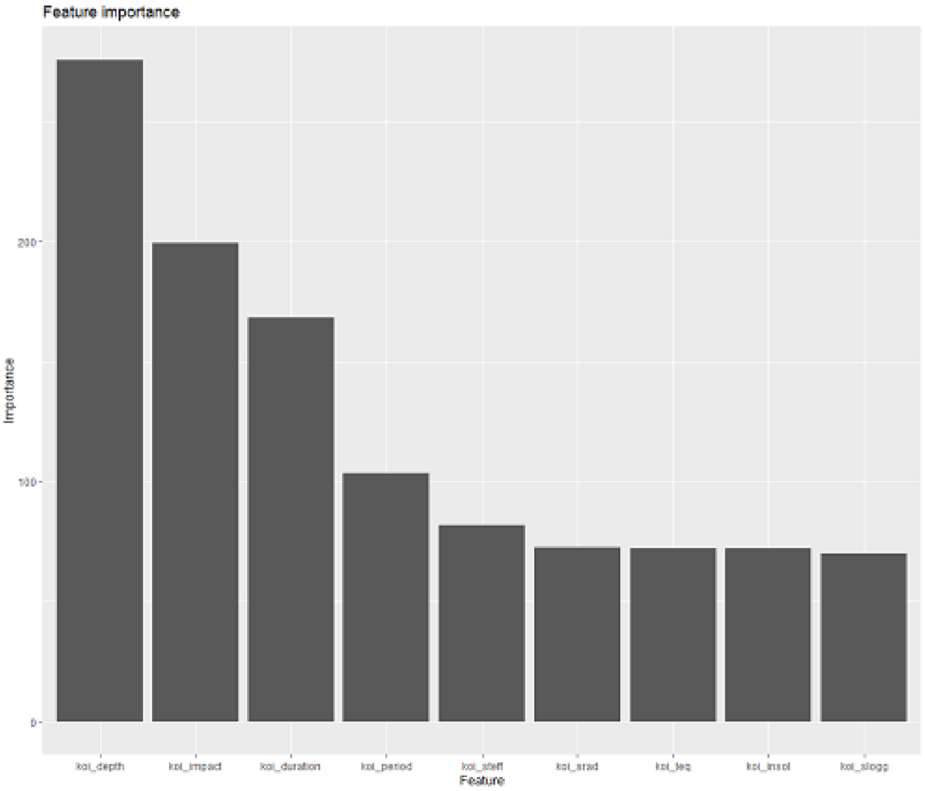


Figure1: Feature importance bar plot

This is a bar chart that shows the feature importance of each variable in the model. The features are sorted by their importance, the x-axis shows the name of the feature, and the y-axis shows the importance value. The importance of a feature is calculated as the average of such foreignness values for all trees, and the lowest foreignness values are considered the most important.

This data frame will be used later in the code to create a bar chart of the feature importances, so that the relative importance of each predictor can be visualized.

Furthermore, we use the random forest model (randomf_model) that was trained earlier in the code to make predictions on the test data (test_data). The "predict()" function returns an output of predicted classes for each observation in the test data, based on the training of the model.

The " `=="` operator is used to compare the predictions with the actual values of the exoplanet variable in the test data, stored in the variable `test_data$exoplanet` . This comparison results in a logical vector, with `TRUE` for correct predictions and `FALSE` for incorrect predictions. Then, the `mean()` function is used to calculate the proportion of correct predictions, which is equivalent to the accuracy of the model.

Afterwards, we use the random forest model (randomf_model) that was trained earlier in the code to predict a score for each observation in the test data (test_data). These scores represent the "anomaly scores" of the test data observations. The type of the argument is set to "response" which means that the function will return the predicted probability of the positive exoplanets class rather than returning the class label itself.

Likewise, whenever the anomaly score, stored in the variable " `anomaly_scores"` , is less than 0.1, we select all the observations from the "test_data" dataframe. So the smaller the score, the more anomalous the observation is. The result of this line is a subset of the “test_data” dataframe containing only the observations that are considered anomalous by the model, these observations are stored in the variable “ `anomalies`” .

The plots, generally, allow us to see patterns and relationships across multiple variables at once, which can be useful for identifying anomalies or outliers in the data, and how they relate to the presence of exoplanets.

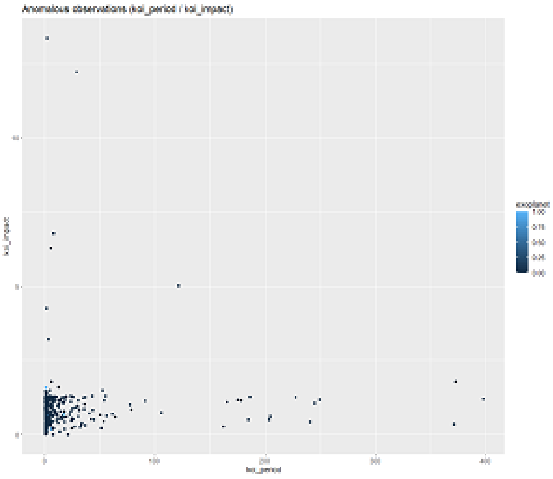


Figure2: Anomalous observations (koi_period / koi_impact)

This plot is showing anomalous observations by plotting, “koi_period’, which is the orbital period of the exoplanet candidate in days, against “koi_impact”, which is the impact parameter of the exoplanet candidate. and coloring the points based on exoplanet variable. The light colored points represents the confirmed or candidate exoplanets while the dark colored points represents the non-exoplanets.

This plot shows how the values of the koi_period and koi_impact are distributed among the anomalous observations, and allows us to see if there are any patterns or trends in the data that might indicate which exoplanet candidates are more likely to be true exoplanets.

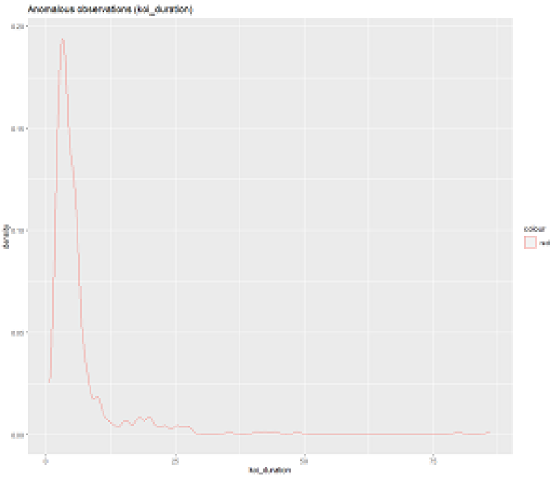


Figure3: Anomalous observations (koi_disposition / koi_period)

This plot is a boxplot showing the relationship between the variable, in the x-axis represents the “koi_disposition”, which is a categorical variable indicating the disposition of this observation, and “koi_period” in the y-axis, which is a numeric variable indicating the orbital period of the object.

This plot shows, in the x-axis the density of the “koi_duration” variable for the observations that were identified as anomalies by the random forest model, and the y-axis represents the “density” of the variable, which is a measure of how frequently different values of “koi_duration” appear in the anomalous observations. This plot is useful for visualizing the distribution of the “koi_duration” variable for the anomalous observations, as well as understanding if there is any specific range of “koi_duration” where most of the anomalies are concentrated. Additionally, it is giving us an idea about how different the anomalous observations are from the non-anomalous observations in terms of “koi_duration”.

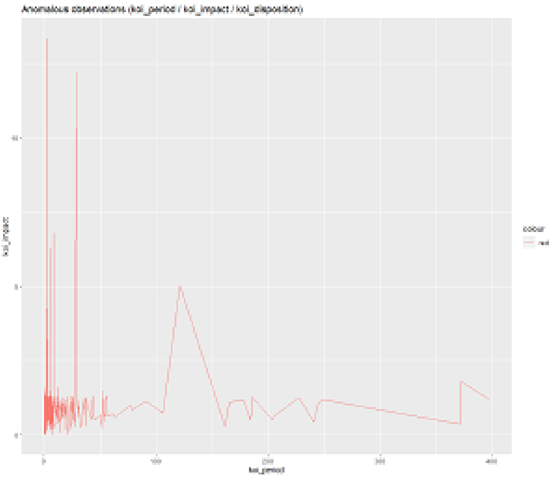


Figure4: Anomalous observations (koi_period / koi_impact / koi_disposition)

This is a line plot that highlights the observations that are identified as exoplanets with a low probability by the random forest model, so the x-axis represents the “koi_period” column and the y-axis represents the “koi_impact” column. The plot shows the relationship between the “koi_period”, “koi_impact” and “koi_disposition” of the anomalous observations, with the koi_disposition as the grouping factor. This plot help us to identify which “koi_disposition” is mostly anomalous and if there are any patterns in the “koi_period” and “koi_impact” values for anomalous observations with different “koi_disposition”.

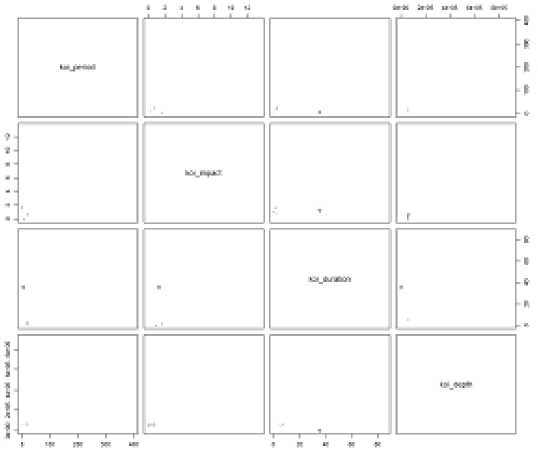


Figure5: scatter plot

The output of the plot is a scatter plot matrix of four variables “koi_period”, “koi_impact”, “koi_duration”, and “koi_depth”, with each point on the plot representing an anomalous observation from the test data indicating whether the observation is an exoplanet or not.

The observations that are plotted are those that were identified as anomalies by the random forest model. The scatter plot matrix is used to visualize the relationship between the variables and helps to understand the distribution of the anomalous observations in the data space.

This plot used also to identify areas where exoplanets are more likely to be found. And generally, the plot serves to understand the characteristics of the anomalous observations and identify patterns in the data that may be indicative of exoplanets.

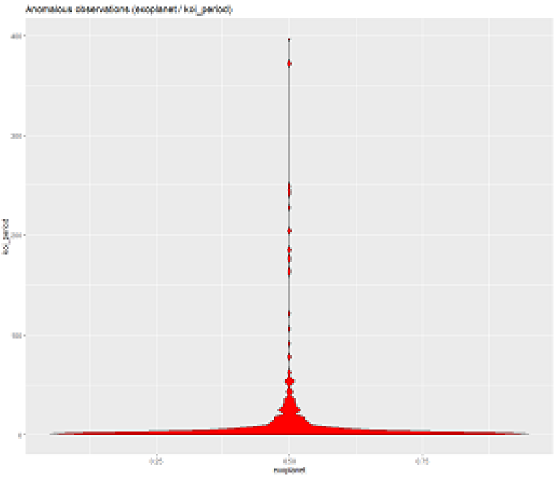


Figure6: Anomalous observations (exoplanet - koi_period)

This plot is a violin plot that shows in the y-axis the distribution of “koi_period” for observations that were identified as anomalies in the x-axis by the random forest model.

This plot allows to understand the characteristics of the anomalous observations in terms of “koi_period” and how they differ from the rest of the observations. It also helps to identify patterns in the data that may be as exoplanets.

It is visualizing the distribution of the data, where a thick line in the center represents the median, and the width of the violin represents the density of the data at different values. In other words, the observations that are plotted are those that were identified as anomalies by the random forest model.

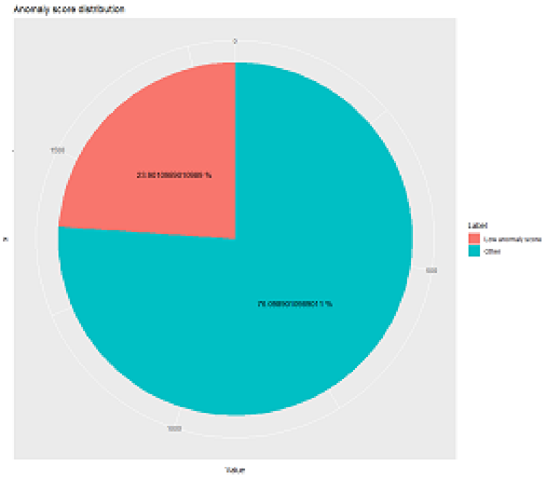


Figure7: Anomaly score distribution

This is a pie chart that shows the distribution of observations by their anomaly scores, the plot is created using the data frame "pie_data" which contains the count and percentage of observations that have an anomaly score less than 0.1. The plot has two sections, one for "Low anomaly score" and another for "Other" observations.

It shows the proportion of observations that were identified as anomalies by the random forest model. The low anomaly score observations are considered as the anomalous observations, this plot is used to visualize the proportion of observations that were identified as anomalies by the model, it shows how many observations are considered as anomalies and how many are not.

Additionally, the plot makes use of the "coord_polar" function to create a polar coordinate system for the chart and "geom_text" to add the percentage of each label, making it easy to understand the proportion of observations in each label.

CONCLUSION :

In conclusion, the goal of this project is to improve the accuracy and efficiency of exoplanet detection by using data mining. The random forest model is trained using the training data and feature importance. The model is then used to predict on the test data and the accuracy of the model is calculated. The exoplanets are extracted from the test data and anomalous observations are plotted and visualized using different plots such as scatter plots, line plots, density plots, and violin plots. The observations that are anomalies are filtered using the threshold of anomaly scores < 0.1 and the number of objects with an anomaly score below the threshold is also calculated. The author also used the randomForest and caret packages in R to perform the above steps.

REFERENCES :

- Breiman. L, Random Forests, Machine Learning, 45, 5-32, 2001
- Freudenthal, J., von Essen, C., Dreizler, S., Wedemeyer, S., Agol, E., Morris, B.M., Becker, A.C., Mallonn, M., Hoyer, S., Ofir, A. and Tal-Or, L., 2018. Kepler Object of Interest Network-II. Photodynamical modelling of Kepler-9 over 8 years of transit observations. *Astronomy & Astrophysics*, *618*, p.A41.
- Gupta, Bhumika, Aditya Rawat, Akshay Jain, Arpit Arora, and Naresh Dhami. "Analysis of various decision tree algorithms for classification in data mining." *International Journal of Computer Applications* 163, no. 8 (2017): 15-19.
- Hora, K., 2018. Classifying exoplanets as potentially habitable using machine learning. In *ICT Based Innovations* (pp. 203-212). Springer, Singapore.
- Jacob, S.G., 2015. Improved random forest algorithm for software defect prediction through data mining techniques. *International Journal of Computer Applications*, *117*(23).

- Jara-Maldonado, M., Alarcon-Aquino, V., Rosas-Romero, R., Starostenko, O. and Ramirez-Cortes, J.M., 2020. Transiting exoplanet discovery using machine learning techniques: a survey. *Earth Science Informatics*, *13*(3), pp.573-600.
- Jenkins, J.M., Caldwell, D.A., Chandrasekaran, H., Twicken, J.D., Bryson, S.T., Quintana, E.V., Clarke, B.D., Li, J., Allen, C., Tenenbaum, P. and Wu, H., 2010. Overview of the Kepler science processing pipeline. *The Astrophysical Journal Letters*, *713*(2), p.L87.
- Kinemuchi, K., Barclay, T., Fanelli, M., Pepper, J., Still, M. and Howell, S.B., 2012. Demystifying Kepler data: A primer for systematic artifact mitigation. *Publications of the Astronomical Society of the Pacific*, *124*(919), p.963.
- Li, L. and Zhang, X., 2010, June. Study of data mining algorithm based on decision tree. In *2010 International Conference On Computer Design and Applications* (Vol. 1, pp. V1-155). IEEE.
- McGarry, K and McDonald S, Complex network theory for the identification and assessment of candidate protein targets, Computers in Biology and Medicine, Vol 97, 113-123, 2018.
- Quanz, S.P., Absil, O., Benz, W., Bonfils, X., Berger, J.P., Defrère, D., van Dishoeck, E., Ehrenreich, D., Fortney, J., Glauser, A. and Grenfell, J.L., 2021. Atmospheric characterization of terrestrial exoplanets in the mid-infrared: biosignatures, habitability, and diversity. *Experimental Astronomy*, pp.1-25.
- Singh, S.P. and Misra, D.K., 2020. Exoplanet hunting in deep space with machine learning. *International Journal of Research in Engineering, Science and Management*, *3*(9), pp.187-192.
- Sturrock, G.C., Manry, B. and Rafiqi, S., 2019. Machine learning pipeline for exoplanet classification. *SMU Data Science Review*, *2*(1), p.9.
- Thompson, S.E., Fraquelli, D., Van Cleve, J.E. and Caldwell, D.A., 2016. *Kepler Archive Manual* (No. ARC-E-DAA-TN46172).
- Tutorials Point. (n.d.). Classification Algorithms - Random Forest. [online] Available at: https://www.tutorialspoint.com/machine_learning_with_python/classification_algorithms_random_forest.htm
- Verikas A et al, Mining data with random forests: a survey and results of new tests. Pattern Recognit 44:330–349, 2011.
- https://www.stat.berkeley.edu/~breiman/RandomForests
- https://cran.r-project.org/web/packages/randomForest/index.html

R-CODE :

```
library(randomForest)
library(caret)

data <- read.csv("cumulative_2020.12.30_14.14.11.csv")

# data pre-processing
count_disposition <- table(data$koi_disposition)
count_p_disposition <- table(data$koi_pdisposition)

# exoplanet to precise if an observation is an exoplanet
```

```
data$exoplanet <- ifelse(data$koi_disposition == "CONFIRMED" | data$koi_disposition ==
"CANDIDATE", 1, 0)

set.seed(123)

# partitioning the data into training and testing sets
split_index <- createDataPartition(data$exoplanet, p = 0.8, list = FALSE)

train_data <- data[split_index,]
test_data <- data[-split_index,]

# training the random forest model
randomf_model <- randomForest(exoplanet ~ koi_period +
koi_impact + koi_duration + koi_depth + koi_teq + koi_insol +
koi_slogg + koi_srad + koi_steff, data = train_data, importance
 = TRUE, ntree = 1000)

# extract the feature importance
feature_importance <- importance(randomf_model)

# create a data frame with the feature importance only
feature_importance_df <- data.frame(feature = row.names(feature_importance),
importance = feature_importance[,1])

# plot the feature importance
ggplot(feature_importance_df, aes(x = reorder(feature, -importance), y = importance)) +
geom_bar(stat = "identity") +
xlab("Feature") +
ylab("Importance") + ggtitle("Feature importance")

# predict on the test data
prediction <- predict(randomf_model, test_data)

# the accuracy of the model
accuracy <- mean(prediction == test_data$exoplanet)

# exoplanets extraction
exo_p <- test_data[prediction == 1,]

# the test data anomaly scores
anomaly_scores <- predict(randomf_model, test_data, type = "response")

# the observations that are anomalies
anomalies <- test_data[anomaly_scores < 0.1,]

# plots
ggplot(anomalies, aes(x = koi_period, y = koi_impact, color =
exoplanet)) +
geom_point() +
ggtitle("Anomalous observations
(koi_period / koi_impact)")

ggplot(anomalies, aes(koi_duration)) + geom_density(aes(color = "red")) +
ggtitle("Anomalous observations (koi_duration)")

ggplot(anomalies, aes(x = koi_period, y = koi_impact, group = koi_disposition)) +
geom_line(aes(color = "red")) +
ggtitle("Anomalous observations (koi_period / koi_impact / koi_disposition)")

# creating scatter plot matrix using the pairs()
pairs(anomalies[,c("koi_period", "koi_impact", "koi_duration", "koi_depth")],
col = anomalies$exoplanet)

# create a violin plot using the geom_violin()
ggplot(anomalies, aes(x = exoplanet, y = koi_period)) +
geom_violin(fill = "red") +
ggtitle("Anomalous observations (exoplanet / koi_period)")

# count the number of objects with an anomaly score below the threshold < 0.1
low_score_count <- sum(anomaly_scores < 0.1)

# count the number of these objects
total_count <- nrow(test_data)

# calculate the percentage of objects with a low anomaly scores
low_score_percentage <- low_score_count / total_count

# creating a data frame
pie_data <- data.frame(Label = c("Low anomaly score", "Other"),
Value = c(low_score_count, total_count - low_score_count),
Percentage = c(low_score_percentage, 1 - low_score_percentage))

# the pie chart
ggplot(pie_data, aes(x = "", y = Value, fill = Label)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start = 0) +
geom_text(aes(label = paste(Percentage * 100, "%")),
position = position_stack(vjust = 0.5)) +
ggtitle("Anomaly score distribution")
```