Machine Learning for Exoplanet Detection Using KOI Cumulative and Stellar Properties DR25 Datasets

Mauricio Ruiz
National College of Ireland
Machine Learning
Dublin, Ireland
x24146102@student.ncirl.ie

Abstract: The main aim of this project is to use five different machine learning algorithms to classify accurately if the celestial objects captured by the Kepler satellite are actual exoplanets. In order to do this, two different datasets have been used to train the models: Kepler Objects of Interest (KOI) cumulative data and Kepler stellar properties stored in Keplers Input Catalog - KIC (DR25). The five algorithms selected for this research are: Random Forest, Extreme Gradient Boosting (XGBoost) , Support Vector Machine, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). The accuracy of these models was assessed with different metrics after performing exploratory data analysis, feature engineering and tuning the algorithms' parameters adequately.

Keywords: supervised machine learning, NASA, exoplanets, kepler.

I. INTRODUCTION

The usage of algorithms is something that has been applied in software development to have computers performing tasks ever since their invention. Over the last years algorithms have been used to create what is now known as machine learning (ML). Machine learning is nothing but programming computers to perform tasks based on data given to train a model and using statistics to obtain the most accurate outcome possible from the data fed to the model. [1]

The Kepler mission launched on March 6th by NASA was an exploration that used a telescope (Kepler) to go into space hunting for earth like planets orbiting other stars outside our solar system. [2]

Exoplanets are any other planet that is outside our solar system and as of today, NASA has confirmed the existence of around 6.000 exoplanets. [3] The Kepler mission observed thousands of objects, known as Kepler Objects of Interest (KOI) and for over nine years the telescope captured images of thousands of stars and information about the KOI orbiting those until it officially retired in October 2018. [4]

The data gathered from Kepler has been made available for the public. The dataset contains different features and has been used to confirm if the KOI are exoplanets or not, or if they are possible candidates based on other observations gathered by the telescope. By using data coming from confirmed exoplanets a machine learning model can be trained to make the process more agile and predict whether the possible candidates in the list are exoplanets.

This project aimed to train five different machine learning models by using two datasets coming from the Kepler mission, KOI Cumulative dataset and KIC (DR25) dataset, to confirm if the KOI is an exoplanet (true positive) or if it is not (false positive).

The chosen models for the project were: Random Forest, XGBoost, Support Vector Machine, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). By using five different models we can assess the robustness of each of them and provide more accurate insights based on cross referencing the models performance and results.

Some of the models used in this project have already been used in other research with similar purposes. Those projects have only used few models. By adding five different algorithms the final results for the predictions can be assessed more in depth and understand better what data is relevant to detecting actual exoplanets.

From another perspective, incorporating the stellar properties (KIC) data to this research, it is possible to understand how the characteristics of stars that the KOIs are orbiting can be relevant to detecting and labelling exoplanets as such.

By adding extra relevant data to the Kepler's cumulative data such as KIC and assessing different ML models, a proper model that produces the most accurate predictions can be selected and deployed to interpret the data and classify the KOIs in an effective and precise manner.

II. RELATED WORK

A. ML in astronomy

ML has been used in different fields over the last few years to classify different types of objects based on gathered data in a more automated manner. In astronomy ML has shown remarkable success when it comes to automating classification processes and has proven to be key to extracting parameters and relationships between themselves from long datasets. [5]

Using ML algorithms and historical data to train a system can allow automatic classification of celestial objects reducing the usage of human resources as well as filtering relevant data that explains the relations between features in extremely long and complicated datasets.

B. ML Models used in exoplanets classification

The usage of different machine learning algorithms is something that has been done already in different research within the astronomy field. These studies have used different approaches, different data with different features and different models that specifically fit the project's purpose.

In relation to exoplanets classification, there have been different projects aiming to understand and create a more accurate and effective way to use ML in the identification of exoplanets. The KOI dataset was used previously to identify the existence of exoplanets by using different trained ML models resulting in successfully validating the existence of exoplanets and also having usability in planet vetting and prioritization. [6]

The usage of ML in identifying exoplanets has been proven to be successful. An example of such success is the discovery of two new exoplanets by using Convolutional Neural Networks, specifically a customized version of the model AstroNet-K2. [7]

Some other approaches have included the usage of AI and ML algorithms designed by ThetaRay Inc which was trained using KOI dataset and validated exoplanets and then applied to Transiting Exoplanets Survey Satellite (TESS) dataset. This resulted in finding three possible exoplanets candidates and 50 targets that need manual vetting. [8]

Taking these publications as a starting point, it is noticeable that many different ML models can be used to classify KOIs into candidates and confirmed exoplanets. These projects have proven that ML is an important tool to reduce the manual work in identifying possible exoplanets considering that the data gathered by the telescopes is extremely large.

C. Contribution

This project has considered the results from previous research and has introduced a new approach. Five ML models were trained to classify KOI into possible candidates or confirmed exoplanets by using KOI cumulative dataset but adding an extra layer using stellar properties captured in KIC.

Using this extra dataset adds more depth to the KOI dataset allowing to identify patterns in how the observed objects by Kepler interact with different features recorded in the stellar dataset.

In summary, the aim of this research is to fill gaps from previous studies by adding the KIC dataset to train the models and produce more accurate results in exoplanets classification.

III. METHODOLOGY

A. Research Design

This study was designed based on CRISP – DM (Cross Industry Standard Process for Data Mining) model. [9]

This methodology is based on the following steps:

- **Business Understanding**: Used to explain the needs and reason for this study. Automate the classification of objects observed by Kepler into confirmed or false positives (FP).
- **Data Understanding:** Includes data collection from Exoplanets archive and Mikulski Archive for Space Telescopes (MAST) and understanding the usability of the data's features in this study.

- **Data Preparation:** Including data cleaning, preparation and creating engineered features in KOI and KIC datasets to prepare them for modelling.
- **Modeling**: Training five different models for classification: Random Forest, XGBoost, SVM, KNN, and MLP.
- **Evaluation:** Analyze the model's accuracy by using different metrics.
- **Deployment:** Not currently to be implemented but these models could be useful if predictions are accurate.

B. Data Collection

1) Kepler Objects of Interest: [10]

- Source: Exoplanets Archive website.
- Features: It contains a total of 49 planetary features.
- Observations: 9564 observations

2) Kepler's Input Catalog: [11]

- Source: Mikulski Archive for Space Telescopes (MAST)
- Features: 11 features recording stellar properties.
- Observations: 13161029 observations

C. Data Preparation and Exploratory Data Analysis (EDA)

Data preparation was done in three stages: before and after merging the two datasets.

1) Pre-merging:

a) *Data filtering:* KIC dataset is very large and was filtered by Keler ID and leaving only the data needed based on Kepler IDs that are included in KOI dataset. KIC dataset went from 13161029 observations to 8214 matching kepler Ids in KOI dataset.

b) Duplicates: Both datasets were checked for possible Kepler ID duplicates. In the KIC dataset there weren't any duplicates, however in the KOI a total of 2288 duplicated Kepler IDs were found. In this case, duplicated values were expected in KOI data as a star (Kepler ID) observed can have one or more celestial objects orbiting. These entries are valuable for the models and were kept.

2) Merging:

The data from the KOI and KIC was merged using a many to one relationship on Kepler ID to include the duplicated values in KOI that are linked to a single star in KIC. The resulting dataset contains 58 features and 9564 entries.

3) Post-merging:

a) Missing Values:

- Two features did not have any data such as koi_teq_err1 and koi_teq_err2 and were dropped.
- kic_parallax had 99.7% missing data and was dropped
- kepler_name had high number of missing values. The names do not bring any relevancy to the models and can be dropped.
- koi_score has 15.7% missing values. This score is calculated to predict the exoplanets and can cause data leakage when training the model. This feature was dropped.
- 39 other features were identified to have missing values. Some of these features are important to train the model. The distribution and missing values ratio were checked for a subset of the variables to confirm if median imputation was the most accurate option. The variables assessed were: koi_prad, koi_depth and kic_teff.
  The median in each case was close to the central tendency of the data as it can be seen in figures 1 to 3 confirming median was a good fit to fill the missing values in the dataset.

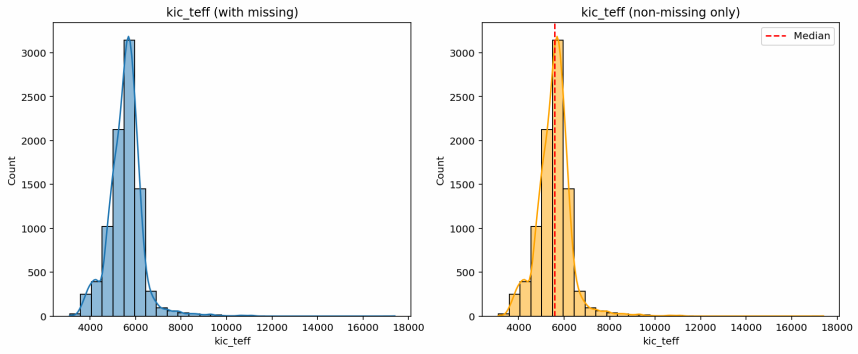

Fig1. Distribution of kic_teff with and without missing values, with median

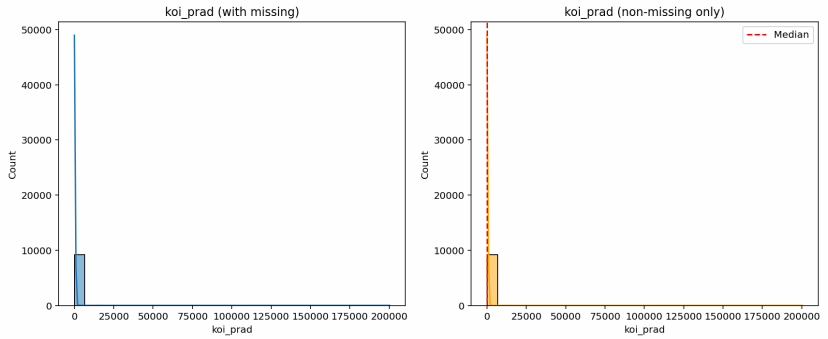

Fig2. Distribution of koi_prad with and without missing values, with median

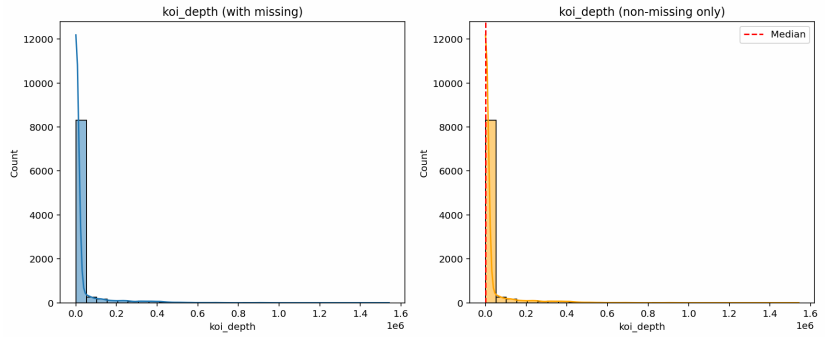

Fig3. Distribution of koi_depth with and without missing values, with median

c) Assessing features variance: This was done to analyse how much information some features can bring to the models, so they are trained on informative features. This was performed as follows:

- By using VarianceThreshold from scikit-learn the features with constant values were removed as they do not bring meaningful information to the models. Two variables were dropped as they are flags.
- By identifying low variance and high coefficient variance (CV) features the model can get more accurate information. The criteria to remove features based on variance was any variable with variance under 0.01 or CV high or negative. A total of 16 features were dropped as they were redundant or correlated with main variables such as the margin of error calculations.

d) *Correlation:* Correlation analysis was done to identify possible correlated features. Highly correlated features create redundancy in the data to train the models. The results can be seen in Figure 4 and Table 1.
Four pairs of features were identified with high correlation (above 0.9). The following features were dropped to avoid multicollinearity:

- kic_kepmag (duplicate of koi_kepmag)

- kepid (identifier only – does not bring relevance to models)
- koi_insol_err1 (error bound for insolation flux)
- kic_teff (duplicate of koi_steff)

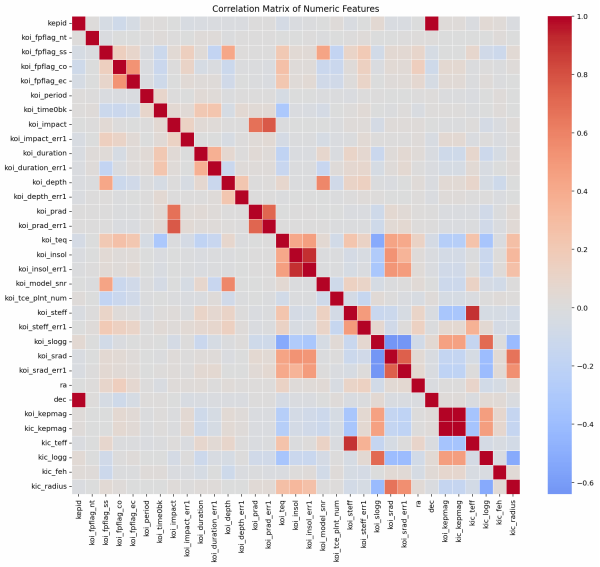

Fig4: Correlation between features in the dataset

Feature 1	Feature 2	Correlation value
koi_kepmag	kic_kepmag	1.000000
kepid	dec	0.993587
koi_insol_	err1 koi_insol	0.904865
kic_teff	koi_steff	0.902339

Table 1. Correlation table between features

e) *Outliers:* 23 features with outliers were identified in the dataset. The outliers were treated as follow:

- *Log Transformation:* Using log1p to highly skewed variables identified based on large ranges.
- *Winsorization:* It was applied to moderate skewed variables capping the extreme value at the 1st and 99th percentile.

f) Feature Scaling: Some models are sensitive to features magnitudes, so scaling was necessary to make sure all features contribute in the same scale to the models training. Z score was applied to three features that

D. Exploratory Data Analysis (EDA)

The EDA was done as follows:

1) Target Variable:

- The target variable's distribution was plotted (Figure 5) to see if there are any imbalances. The data seems to be imbalanced as there are more false positives than candidates or confirmed exoplanets. This was addressed by using Synthetic Minority Over-sampling Technique (SMOTE) as this technique suits all the models and allowed the models to not be biased towards the dominant class and reduce overfitting. This was done after splitting the data into training and test and applied only to the distance based models training data.

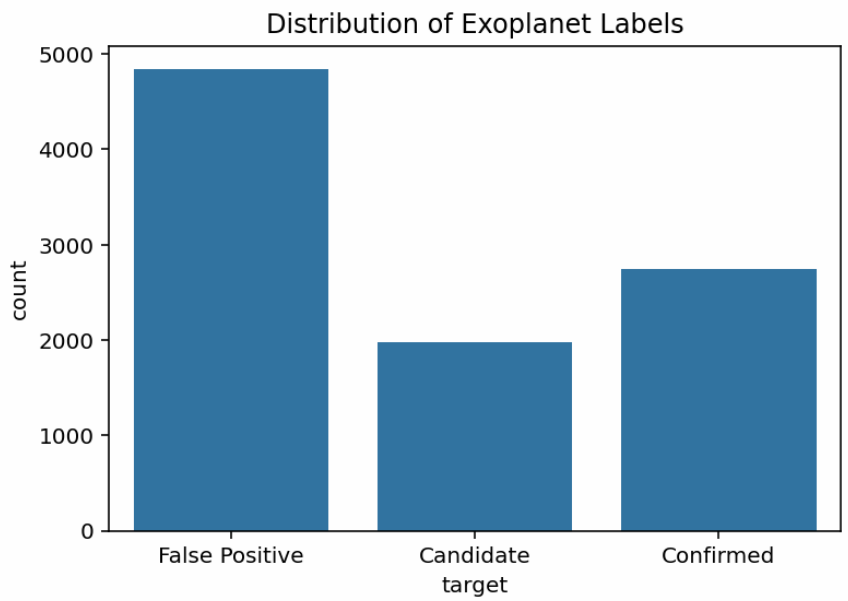

Fig5. Distribution of classes in target variable

3) Top 15 features contributing to the model:

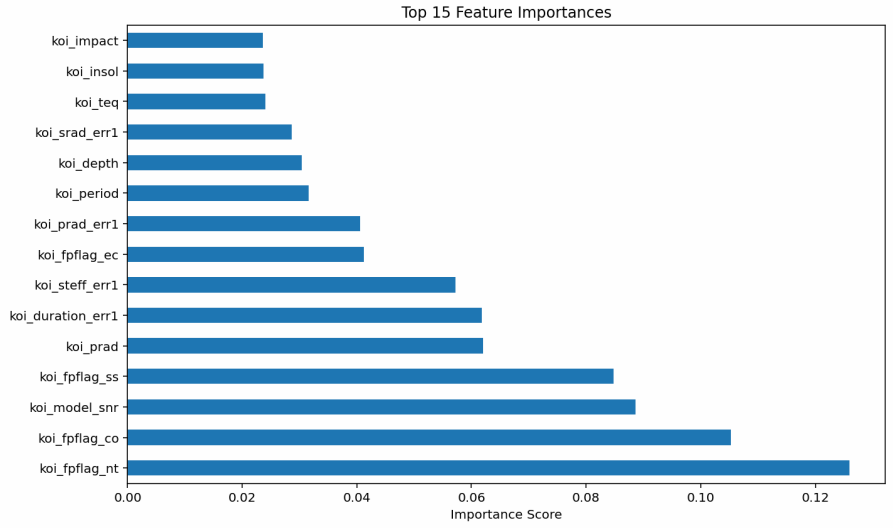

Fig6. Top 5 features contributing to the model

- Non transit like flag (koi_fpflag_nt): This variable is binary, and measures represent if the transit resembles or not a planet (1 = non-transits like a planet). In figure 7 it can be seen that when the variable is 0 it is more likely that the observed object is planet or a candidate.

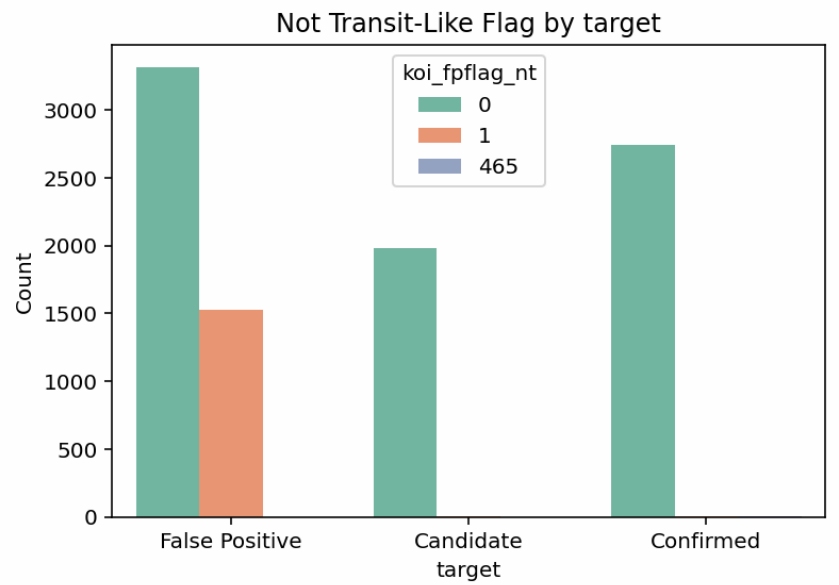

Fig7. Count of Not Transit like per class

- Centroid Offset (koi_fpflag_co): This is a binary variable that checks if the transit is coming from the observed star (values = 0) or not (values =1). The false positives in figure 8 have a significant number of contaminations therefore more 0 values are observed.

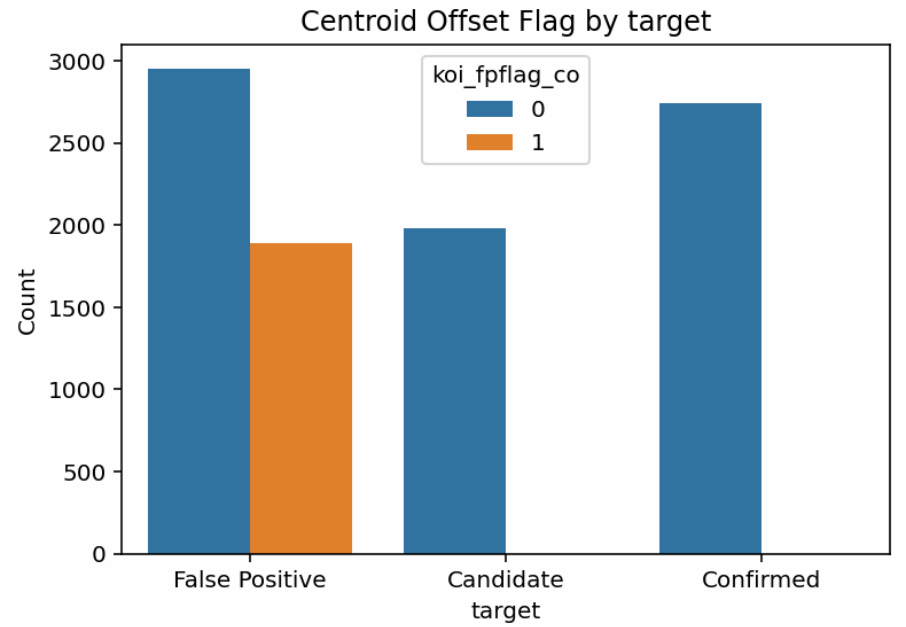

Fig8. Count of Centroid offset per class

- Significant Secondary flag (koi_fpflag_ss): Measure possible secondary eclipses and it is binary. Possible secondary eclipses are represented by value =1. Possible or confirmed exoplanets have a low number of possible secondary eclipses (figure 9).

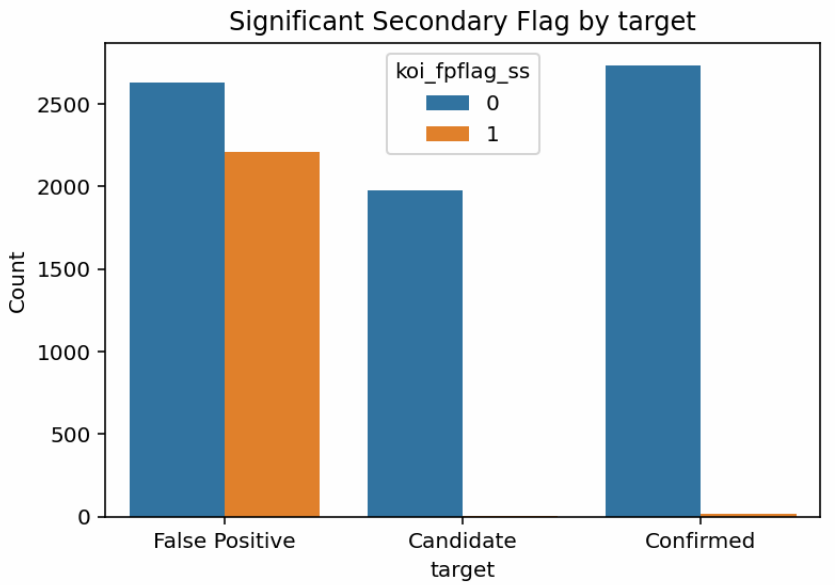

Fig9. Count of Secondary Flag per class

- Noise Ratio (koi_model_snr): Measures noise in the transit. A higher value means the transit is more reliable (less noise). The boxplot plotted (figure 10) shows that candidates or confirmed planets tend to cluster higher than the false positives. It can also be seen that false positives have a high SNR which means that this only feature is not enough to determine the exoplanet's existence.

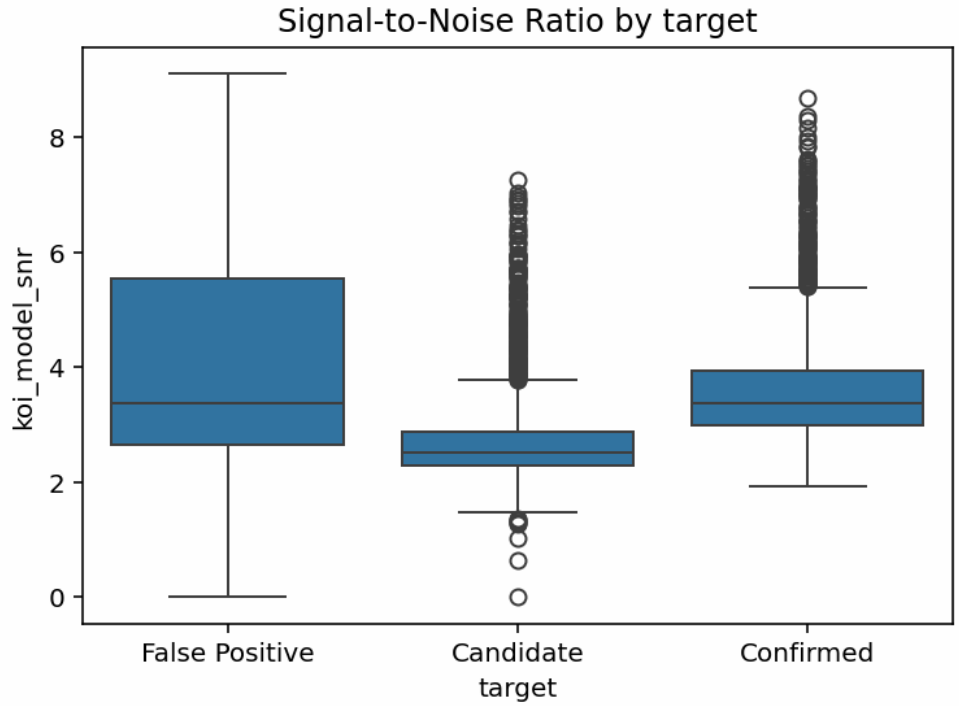

Fig10. Noise ratio per class

Planet Radius (koi_pra): This variable measures the radius of the observed object. As seen in the boxplot (figure 11) the lower the radius the more likely is to be a candidate or confirmed exoplanet.

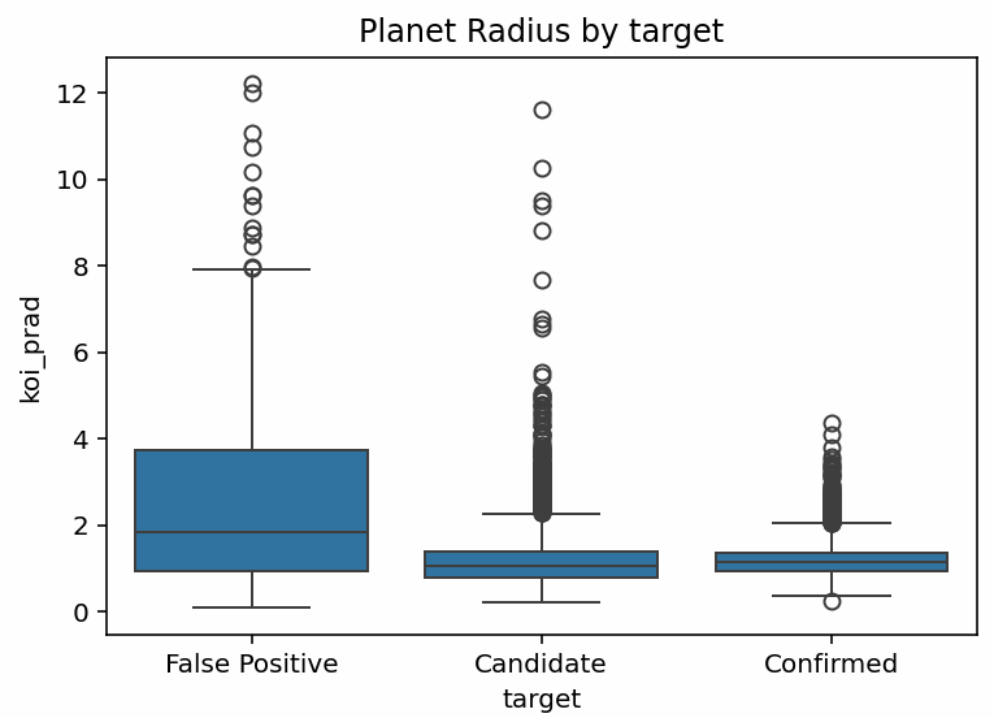

Fig11. Planet radius per class

E. Feature Engineering

To improve the model's accuracy, it was necessary to engineer some features that would capture important information to train the models. Three new variables were calculated:

1) *Planet Ratio to Star:* A new feature was calculated which measures the ratio of the observed object compared to the star. This is meaningful to the models as it allows to understand if the planets are identifiable based on the star's size.
   Formula: Planet ratio = planet radius (koi_prad) / star radius (koi_srad)
2) *Depth ratio to Planet Radius:* It was used to identify the transit based on the planet's radius. It helps the model to detect inconsistencies and identify false positives.
   Formula: Depth ratio = Depth (koi_depth) / Planet radius (koi_prad)
3) *Duration Ratio:* It was calculated the proportion of the duration of the transit compared to the orbital period. It helps the model to understand if the duration of the transit is linked to period and identify more accurately true positives.
   Formula: Duration ratio = Duration (koi_duration) / Period (koi_period)

IV. MODELS

Data still contained a wide difference in scales. This issue was fixed by using Z-Score normalization for the distance based models such as KNN, SVM and MLP.

For the three based models the dataset was kept as it was as these models are not sensitive to variable scales.

The data used to train the models was split as follows:

- Train the model: 80%
- Test the model: 20%

A. XGBoost

Extreme Gradient Boost (XGBoost) was the one of the ML models selected. XGBoost uses gradient boosting which creates decision trees that improve the decision of previous ones to create more accurate predictions. It is a model that has been used for predictions in different fields, and it is robust for classifications as it uses different regularization techniques L1 (Lasso) and L2 (Ridge) to avoid having an overfitting model.

B. Random Forest

Random Forest is another chosen ML model and similar to XGBoost, random forest uses a learning method that makes decisions based on different trees and by combining the output makes more robust predictions. RF reduces variance and improve accuracy by aggregating predictions from different trees. [10]

C. KNN

Another model selected was KNN which classifies datapoints based on the majority K class neighbors. KNN models need to be treated carefully and tune the K number properly to avoid overfitting. Small K values can lead to an overfitting model and large K number can have the opposite effect. [10]

D. MLP

Multi-Layer Perceptron is an Artificial Neural Network (ANN) that has been used in different fields for classification. It uses error backpropagation as a training technique. While the model goes forward a training vector is classified whereas during the backward direction the model updates weights. [11]

E. SVM

SVM is another model that had good results in classification models. The model finds the optimal hyperplane to separate data into classes. To obtain better results hyperparameters can be tuned and using cross validation and regularization by using grid search. [11]

F. Tools

The model was implemented using python using Spyder as IDE and a series of libraries such as:

pandas – for data manipulation and cleaning.

numpy – for numerical computations and array operations.

matplotlib – for creating plots and visualizations.

seaborn – for creating plots and visualizations.

scipy.stats.mstats.winsorize – for outlier treatment

sklearn.preprocessing.StandardScaler – for feature scaling and normalization.

sklearn.ensemble.RandomForestClassifier – for tree-based classification modelling.

sklearn.metrics.classification_report, confusion_matrix – for model evaluation.

sklearn.svm.SVC – for Support Vector Machine classification.

sklearn.neighbors.KNeighborsClassifier – for K-Nearest Neighbors classification.

sklearn.neural_network.MLPClassifier – for Multi-Layer Perceptron (ANN) modeling.

imblearn.over_sampling.SMOTE – for handling class imbalance via synthetic oversampling.

xgboost.XGBClassifier – for gradient-boosted decision tree modeling.

V. RESULTS

A. XGBoost:

1) Performance

The model was trained using 80/20 partition. The algorithms' hyperparameters were tuned manually three times to get the most accurate results along with early stopping to avoid overfitting. The algorithm stopped after 668 iterations providing the most optimal results. The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy (table 3). The final most accurate results are reflected in table 2. The model's predictions were accurate 92.8% of the times and the predictions in class 0 (false positives) were accurate 98.8% compared to a 83.4% in class 1 (candidate) and 89% for class 2 (confirmed) based on F-1 score.-

Hyperparameter	1st Round	3rd Round
n_estimators	600	1000
max_depth	5	6
learning_rate	0.05	0.03
subsample	0.9	0.8
colsample_bytree	0.9	0.8
random_state	42	42
early_stopping_rounds	20	20

Table 2. Hyperparameter tunning rounds for classification model (XGBoost)

Classification	Precision	Recall	F-1 Score	Support
0 (FP)	0.99	0.98	0.98	968
1 (Candidate)	0.82	0.84	0.83	396
2 (Confirmed)	0.89	0.88	0.89	549
Accuracy			0.928	1913
Macro Average	0.903	0.905	0.904	1913
Weighted Average	0.928	0.928	0.928	1913

Table 3. Classification Performance Metrics by Class (XGBoost)

4) *Prediction Results:* Using a confusion matrix, it is possible to identify the total of correct results (figure 12). It is shown that the model has high accuracy at predicting false positives and confirmed exoplanets. This supports the F1-Score results that the model is accurate predicting correctly the three classes. The model can clearly differ between false positives and confirmed.

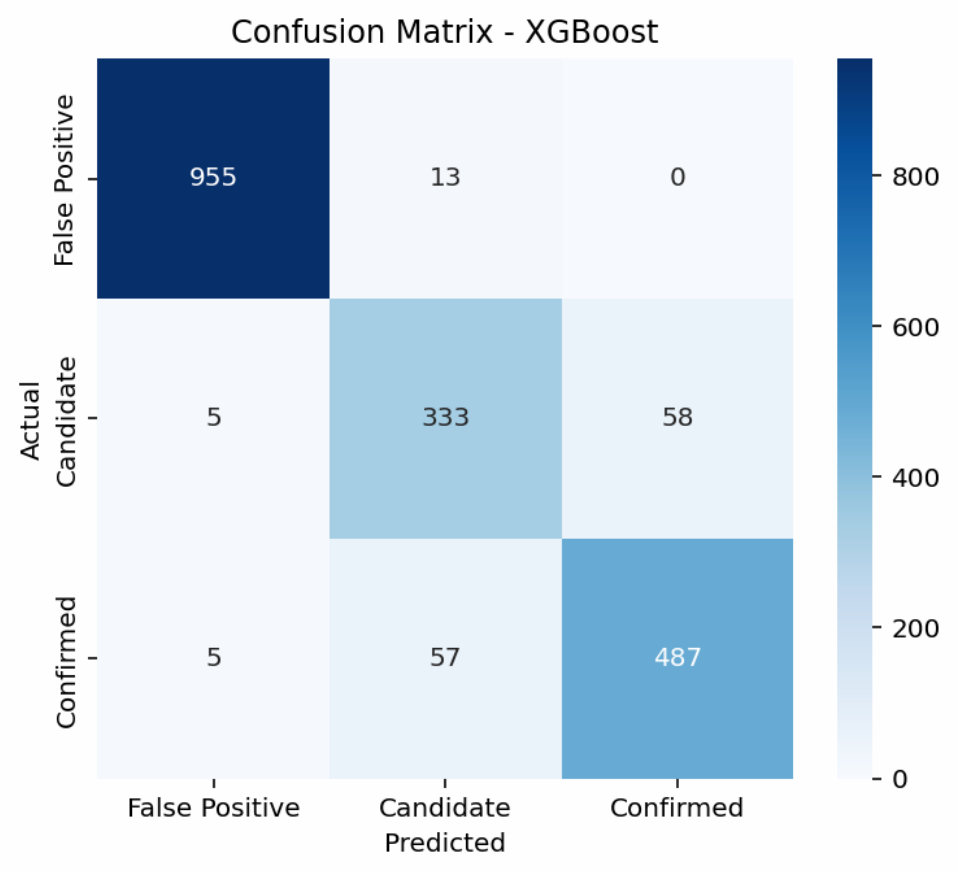

Fig 12. Confusion matrix displays predictions by class (XGBoost)

The confidence vs correctness histogram (figure 13) allows to understand how the model behaved when making decisions predicting classes. The model was very confident and accurate at predicting correctly however, the model predicted some incorrect classes with high confidence.

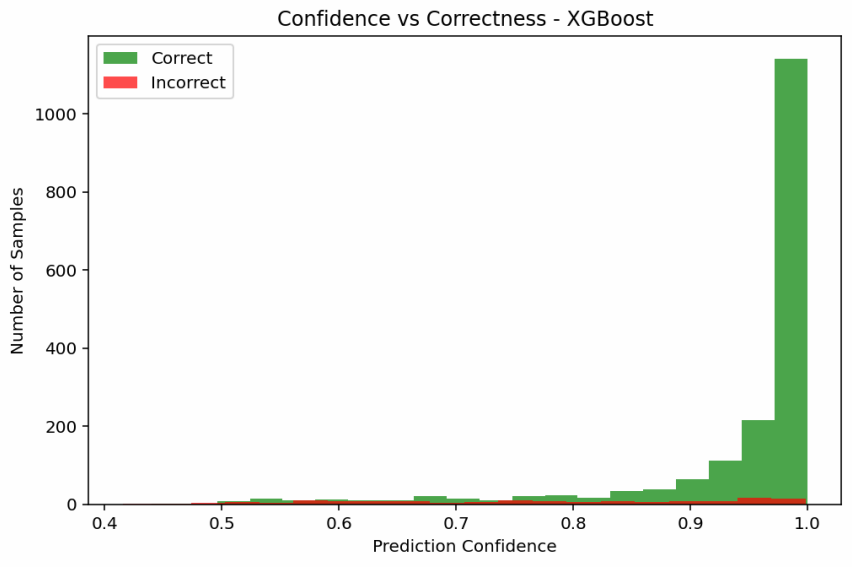

Fig 13. Prediction Confidence vs Correctness for XGBoost classification

B. Random Forest:

1) *Performance:* The model was trained using 80/20 partition. The algorithms hyperparameters were tuned manually three times to get the most accurate results. The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy (table 5). The final most accurate results are reflected in table 4. The model's predictions were accurate 91.9% of the times and the predictions in class 0 (false positives) were accurate 98.9% compared to a 79.2% in class 1 (candidate) and 89.7% for class 2 (confirmed) based on F-1 score.-

Hyperparameter	1st Round	3rd Round
n_estimators	400	1500
max_depth	None	40
min_samples_split	2	20
min_samples_leaf	10	6
'max_features':	'log2'	'sqrt'

Table 4. Hyperparameter tunning rounds for classification model (RF)

Classification	Precision	Recall	F-1 Score	Support
0 (FP)	0.989	0.983	0.986	968
1 (Candidate)	0.792	0.854	0.821	396
2 (Confirmed)	0.897	0.854	0.875	549
Accuracy			0.919	1913
Macro Average	0.892	0.897	0.894	1913
Weighted Average	0.921	0.919	0.920	1913

Table 5. Random Forest accuracy

5) *Prediction Results:* Using a confusion matrix, it is possible to identify the total of correct results (figure 14). It is shown that the model has high accuracy at predicting false positives and confirmed exoplanets. This supports the F1-Score results that the model is accurate predicting correctly the three classes. The model can clearly differ between false positives and confirmed.

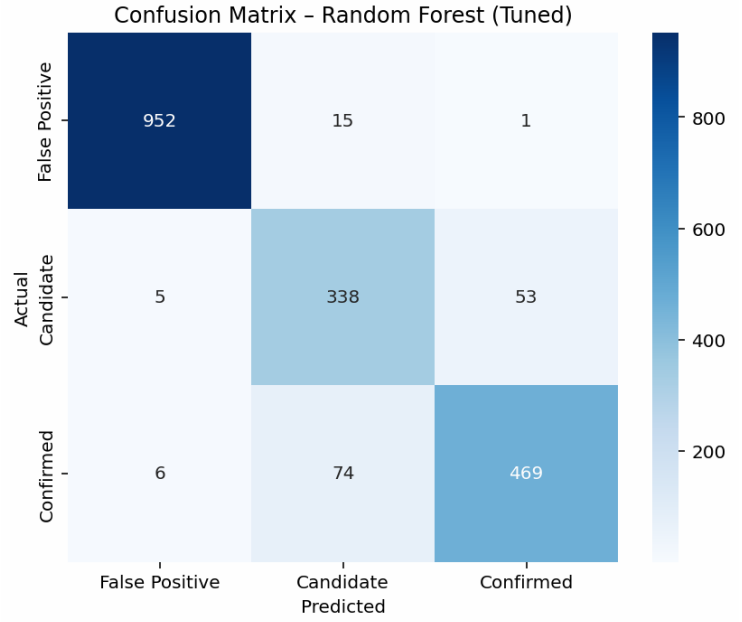

Fig 14. Confusion matrix displays predictions by class (RF)

The confidence vs correctness histogram (figure 15) allows to understand how the model behaved when making decisions predicting classes. The random forest results are similar to XGBoost but this model was more confident and accurate at predicting correctly the classes with lower incorrect predictions done with confidence.

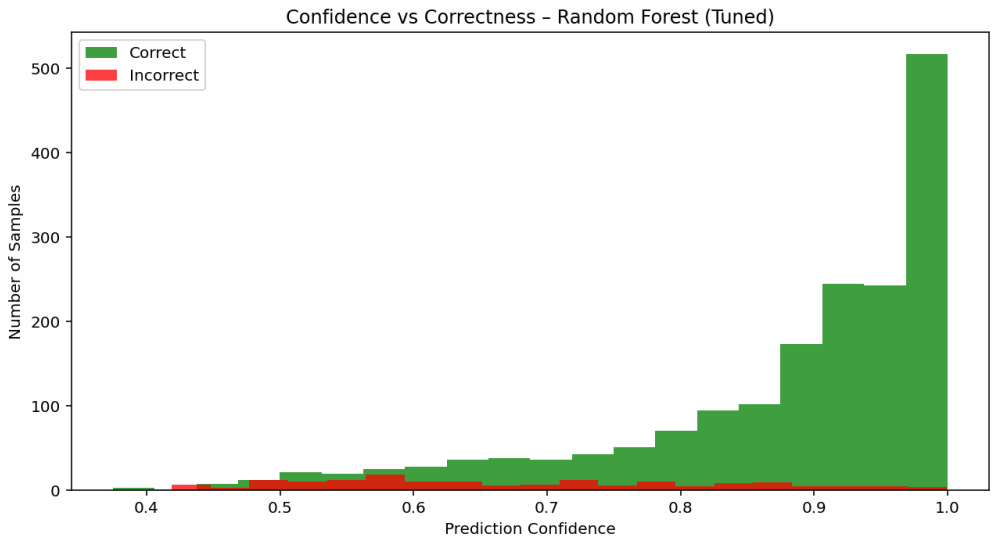

Fig 15. Prediction Confidence vs Correctness for Random Forest classification

C. SVM

1) The model was trained using 80/20 partition. The algorithms' hyperparameters were tuned manually three times to get the most accurate results (table 6). The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy (table 7). The final most accurate results are reflected in table 7. The model's predictions were accurate 89.8% of the times and the predictions in class 0 (false positives) were accurate 96.7% compared to a 77.1% in class 1 (candidate) and 86.5% for class 2 (confirmed) based on F-1 score. The model seems to have some skewness towards class 0 (false positive) with a higher percentage than in the other classes this is possibly due to class imbalance.-

Hyperparameter	1st Round	3rd Round
C	10	3
Gamma	Auto	scale

Table 6. Hyperparameter tunning rounds for classification model (SVM)

Classification	Precision	Recall	F-1 Score	Support
0 (FP)	0.968	0.967	0.967	968
1 (Candidate)	0.772	0.770	0.771	396
2 (Confirmed)	0.864	0.867	0.865	549
Accuracy			0.898	1913
Macro Average	0.868	0.868	0.868	1913
Weighted Average	0.898	0.898	0.898	1913

Table 7. SVM accuracy

6) *Prediction Results:* Using a confusion matrix, it is possible to identify the total of correct results (figure 16). It is shown that the model has high accuracy at predicting false positives and confirmed exoplanets. This supports the F1-Score results that the model is accurate predicting correctly the three classes. The model can clearly differ between false positives and confirmed.

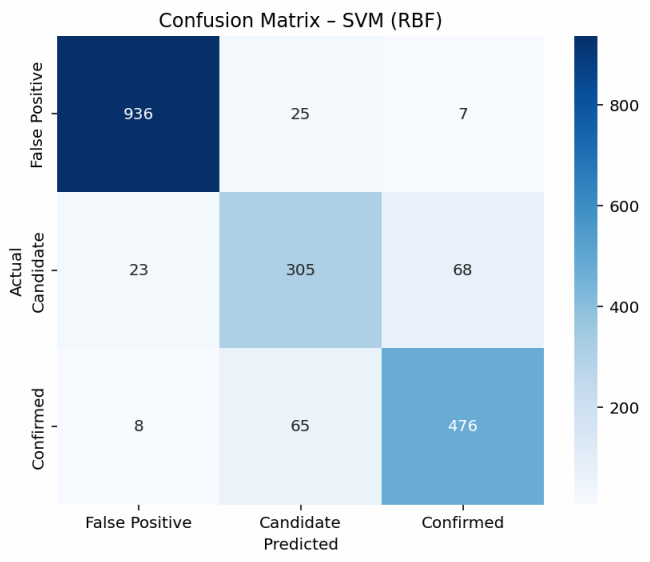

Fig 16. Confusion matrix displays predictions by class (SVM)

The confidence vs correctness histogram (figure 17) allows to understand how the model behaved when making decisions predicting classes. This model predicted with high confidence most of the classes correctly, however there are concerns that it has also predicted incorrectly with the same high level of confidence.

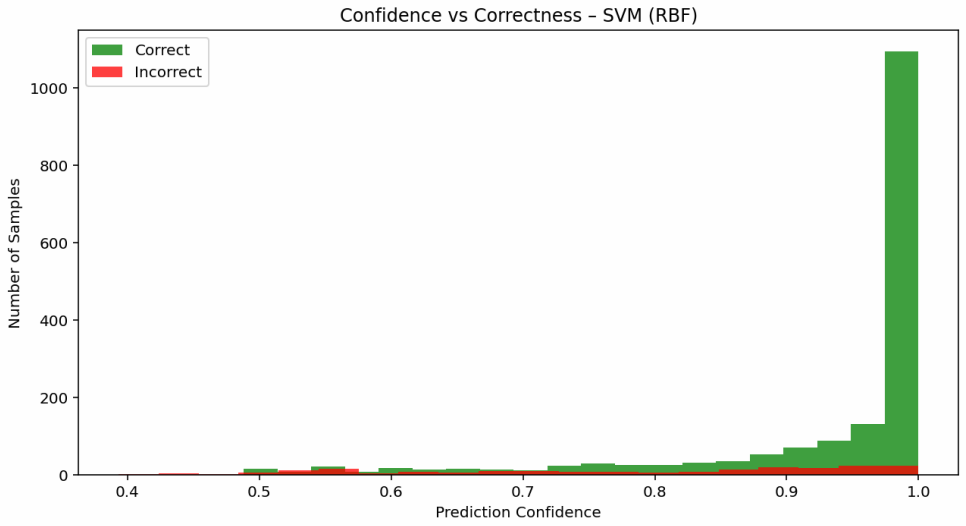

Fig 17. Prediction Confidence vs Correctness for SVM classification

D. KNN

The model was trained using 80/20 partition. The algorithms' hyperparameters were tuned manually three times to get the most accurate results (table 8). The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy (table 9). The final most accurate results are reflected in table 9. The model's predictions were accurate 81% of the time and the predictions in class 0 (false positives) were accurate 89.2% compared to a 60.6% in class 1 (candidate) and 83.2% for class 2 (confirmed) based on F-1 score.

Hyperparameter	1st Round	3rd Round
n_neighbors	15	10
Weights	'distance'	'uniform'

p	2	1

Table 8. Hyperparameter tunning rounds for classification model (KNN)

Classification	Precision	Recall	F-1 Score	Support
0 (FP)	0.938	0.850	0.892	968
1 (Candidate)	0.559	0.662	0.606	396
2 (Confirmed)	0.818	0.845	0.832	549
Accuracy			0.810	1913
Macro Average	0.722	0.786	0.776	1913
Weighted Average	0.825	0.810	0.815	1913

Table 9. KNN accuracy

1) *Prediction Results:* Using a confusion matrix, it is possible to identify the total of correct results (figure 18). It is shown that the model has high accuracy at predicting false positives and confirmed exoplanets but struggles predicting candidates (class 1) accurately. This supports the F1-Score results that the model is accurate predicting correctly the three classes. The model can clearly differ between false positives and confirmed. KNN captured confirmed planet accurately but its accuracy for class 2 (confirmed) is lower than the previous models.

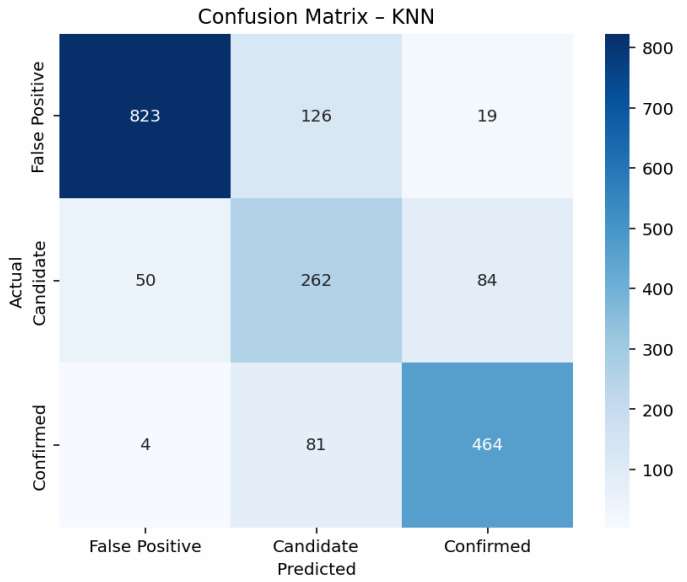

Fig 18. Confusion matrix displays predictions by class (KNN)

The confidence vs correctness histogram (figure 19) shows how the model behaved when making decisions predicting classes. The model was confident in predicting correctly between 0.9 and 1 however it struggled in predicting in the middle range of confidence. This means the model might struggle making decisions compared to the other models. The confidence of the models needs to be reassessed as it shows also high confidence in incorrect predictions.

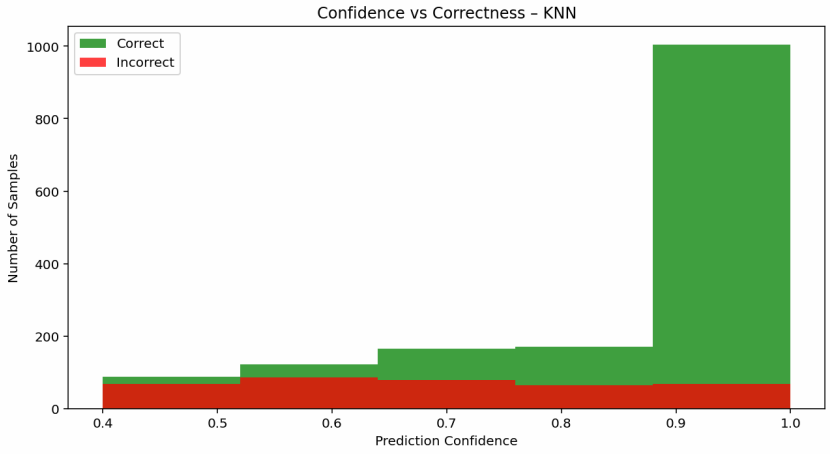

Fig 19. Prediction Confidence vs Correctness for KNN classification

E. MLP

The model was trained using 80/20 partition. The algorithms' hyperparameters were tuned manually three times to get the most accurate results. The algorithms' hyperparameters were tuned three times to get the most accurate results (table 10). The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy. The final most accurate results are reflected in table 11. The model's predictions were accurate 89.5% of the time and the predictions in class 0 (false positives) were accurate 98.1% compared to 77.1% in class 1 (candidate) and 83.6% for class 2 (confirmed) based on F-1 score.

Hyperparameter	1st Round	3rd Round
hidden_layer_sizes	256, 128	128,64
Activation	Tanh	Relu
Alpha	1e-5	1e-4
learning_rate_init	1e-5	1e-3

Table 10. Hyperparameter tunning rounds for classification model (MLP)

Classification	Precision	Recall	F-1 Score	Support
0 (FP)	0.986	0.979	0.981	968
1 (Candidate)	0.741	0.803	0.771	396
2 (Confirmed)	0.860	0.814	0.836	549
Accuracy			0.895	1913
Macro Average	0.861	0.866	0.863	1913
Weighted Average	0.898	0.895	0.896	1913

Table 11. MLP accuracy

1) *Prediction Results:* Using a confusion matrix, it is possible to identify the total of correct results (figure 20). It is shown that the model has high accuracy at predicting false positives and confirmed exoplanets but struggles predicting candidates (class 1) accurately. This supports the F1-Score results that the model is accurate predicting correctly the three classes. The model can clearly differ between false positives and confirmed. MLP captured confirmed planet accurately but its accuracy for class 2 (confirmed) is the lowest of all the previous models.

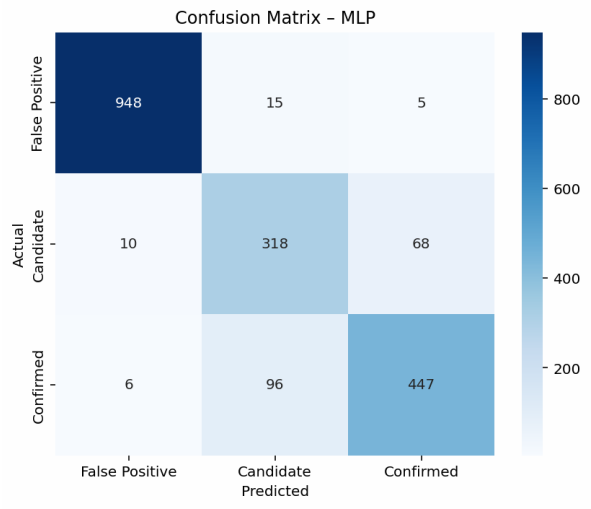

Fig 20. Confusion matrix displays predictions by class (MLP)

The confidence vs correctness histogram (figure 21) shows how the model behaved when making decisions predicting classes. The model was confident in predicting correctly between 0.9 and 1. MLP has show high confidence in the decisions. It still has issues detecting accurate candidates and predicting incorrectly with high confidence.

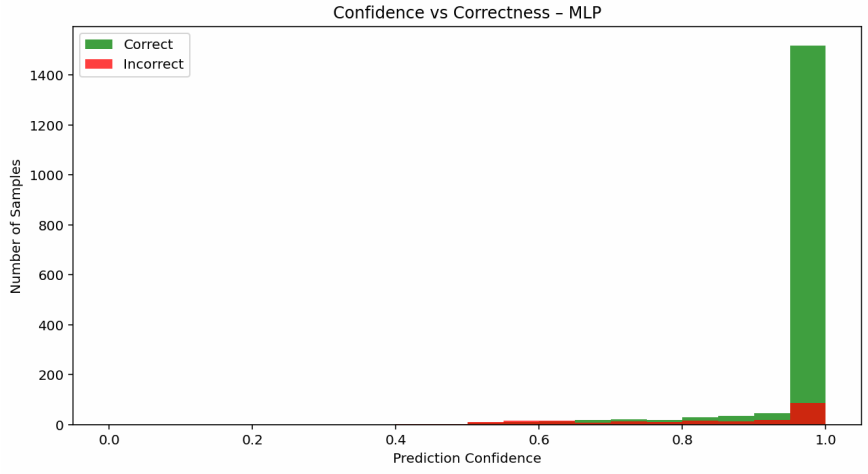

Fig 20. Prediction Confidence vs Correctness for MLP classification

The main objective of this project was to identify accurately possible exoplanets, candidates or false positives by using two related datasets that captured information from different stars. All five models have shown some consistency in the results but showing some are more accurate than others.

VI. CONCLUSIONS

A. Conclusions

Five different ML models were implemented to classify celestial objects observed by the Kepler satellite into false positives, candidates and confirmed exoplanets. The models used were XGBoost, Random Forest, Support Vector Machine (RBF), K-Nearest Neighbors, and Multi-Layer Perceptron. Each of them provided different results in classification and accuracy.

The tree-based models achieved the most accurate results overall, showing a more balanced performance (despite using only for these models imbalanced data) and predicting more accurately confirmed exoplanets as well as false positives.

SVM performed well overall but struggled in classifying possible candidates (class 1) and struggling to make accurate decisions in the mid-range of confidence levels as seen in the confidence plot.

KNN showed moderate accuracy performance but due to the high multidimensionality did not perform as well as other models. This model also predicted a high number of cases incorrectly with high confidence reducing reliability in its results.

MLP achieved good accuracy, however, showed high confidence which leads to considering that needs to be calibrated in deeper level. This model also predicted incorrectly with high confidence which makes its results less reliable.

Tree based models performed much better than the other models demonstrating their robustness and ability to perform with the current data. However, all the models seemed to have a higher accuracy in class 0 (FP) due to higher sample of that specific class.

B. Future work

Alternative techniques to balance data: These models can benefit from using different class balancing rather than SMOTE. This is recommended for future research using the same datasets.

Models' optimization: Hyperparameter tuning can be optimized by implementing other tools which could lead to improve accuracy, robustness and more confident decisions.

Domain expertise: Collaborate with subject matter experts to improve feature engineering that can provide more meaningful information to the models to improve accuracy.

VII. REFERENCES

[1] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2020.

[2] NASA, “Kepler / K2,” NASA, 2018. [Online]. Available: https://science.nasa.gov/mission/kepler/. [Accessed 01 08 2025].

[3] NASA, “Exoplanets,” NASA, [Online]. Available: https://science.nasa.gov/exoplanets/. [Accessed 01 08 2025].

[4] NASA, “What Did Kepler Teach Us? Celebrating the Space Telescope, 10 Years after Launch,” 06 03 2019. [Online]. Available: https://science.nasa.gov/universe/exoplanets/what-did-kepler-teach-us-celebrating-the-space-telescope-10-years-after-launch/. [Accessed 01 08 2025].

[5] Z. L. J. W. a. Z. W. Guangping Li, “Machine Learning in Stellar Astronomy: Progress up to 2024,” 21 02 2025. [Online]. Available: https://arxiv.org/abs/2502.15300. [Accessed 02 08 2025].

[6] J. G. ,. T. D. David J Armstrong, “Exoplanet validation with machine learning: 50 new validated Kepler planets,” 07 2021. [Online]. Available: https://academic.oup.com/mnras/article/504/4/5327/5894933. [Accessed 02 08 2025].

[7] A. V. C. J. S. A. W. M. P. B. A. B. M. L. C. G. A. E. M. E. E. S. B. H. D. W. L. N. J. S. L. Y. Anne Dattilo, “Identifying Exoplanets with Deep Learning II: Two New Super-Earths Uncovered by a Neural Network in K2 Data,” 25 03 2019. [Online]. Available: https://arxiv.org/pdf/1903.10507. [Accessed 02 08 2025].

[8] A. A. A. S. I. B. D. S. A. R. Leon Ofman, “Automated identification of transiting exoplanet candidates in NASA Transiting Exoplanets Survey Satellite (TESS) data with machine learning methods,” 02 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1384107621001196#d1e1862. [Accessed 02 08 2025].

[9] IBM, “Introduction to CRISP-DM,” IBM, 17 08 2021. [Online]. Available: https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-introduction-crisp-dm. [Accessed 16 07 2025].

[10] NASA, “NASA Exoplanet Archive,” 10 02 2021. [Online]. Available: https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative. [Accessed 10 05 2025].

[11] M. A. f. S. Telescopes, “Kepler,” 20 01 2017. [Online]. Available: https://archive.stsci.edu/pub/kepler/catalogs/kic.txt.gz. [Accessed 10 07 2025].

[12] A. Jain, Machine Learning in 24 Hours: The Ultimate Beginner’s Guide, 2025.

[13] V. S. Anke Meyer-Baese, “Chapter 7 - Foundations of Neural Networks,” in *Pattern Recognition and Signal Analysis in Medical Imaging*, 2014, p. 197.

[14] IBM, “What are support vector machines (SVMs)?,” [Online]. Available: https://www.ibm.com/think/topics/support-vector-machine. [Accessed 03 08 2025].

[15] V. Rahangdale, XGBoost Explained: A beginner's Journey into Machine Learning, 2025.