Kepler Exoplanet Detection

Victoria Brendel, Kenneth Hahn, Moez Hudda, Matthew Paterno

DS207 Final Project

Github: https://github.com/hahnkenneth/mids_w207_fall2024_brendel_hahn_hudda_paterno

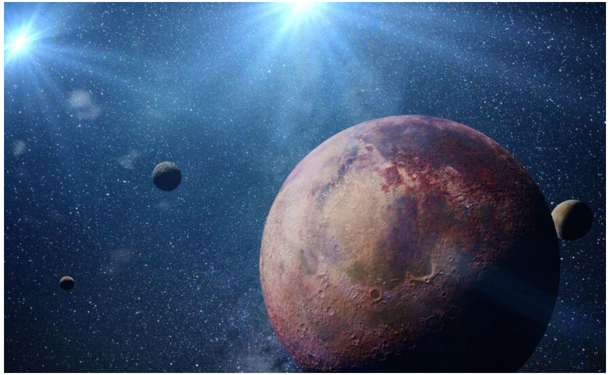
Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

1. Background/Motivation
2. Data source/structure
3. Data pre-processing and feature engineering
4. Baseline model: logistic regression
5. EDA
6. Final selected model
7. Evaluation
8. Altering threshold effect on metrics
9. Conclusion

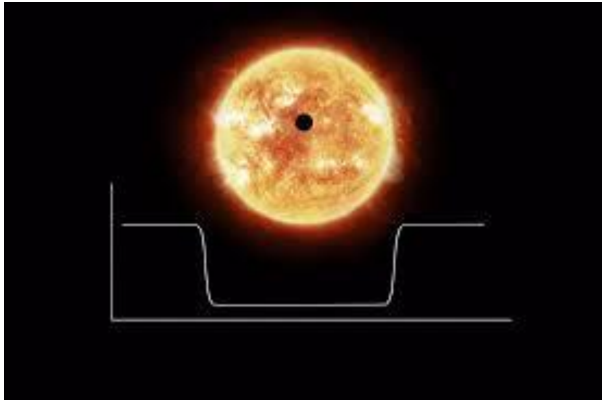
Background

- Exoplanet detection is a cornerstone of modern astronomy
- Provides crucial insights to planetary formation, the potential for extraterrestrial life, and dynamics of distant star systems



- The Kepler Space Telescope operated from 2009 to 2018, which surveyed over 200,000 stars in the Milky Way Galaxy to identify transiting exoplanets - detected by measuring the amount of light it blocks from the star it is orbiting
- Key attributes of confirming an exoplanet detection include the orbital period of the exoplanet, temperature of the star, etc.
- These features form the foundation for a binary classification task: predicting whether an observation corresponds to an exoplanet (1) or not an exoplanet (0).

Motivation



- Scale of data: Kepler telescope dataset contains thousands of observations with numerous features
- Manual analysis is time-consuming, prone to error, and incapable of scaling to datasets from upcoming missions
- Detection accuracy: traditional classification methods may miss subtle patterns in data - advanced feature engineering and optimization can identify non-linear relationships and improve classification accuracy
- Automating the detection process accelerates the pace of discovery, allowing astronomers to focus on further validation and characterization.
- **Goal: We aim to create a ML pipeline that can automate exoplanet detection through binary classification to expedite the process of evaluating exoplanet candidates**

Data Source/Structure

- Kepler Exoplanet Search Results, originates from Kepler Space Telescope observations released by NASA, and accessible on Kaggle
- Dataset includes both confirmed exoplanet and false positives
- 9,576 rows and 49 columns
  - Target variable:
    - Koi_pdisposition: exoplanet (1) or false positive (0)
  - Planetary attributes:
    - Koi_period: orbital period in days of exoplanet candidate
    - Koi_duration: duration (hours) of transit
    - Koi_depth: depth of transit, indicative of planet's size relative to the star
  - Metadata:
    - Kepid: Kepler ID of observed star
    - Koi_score: disposition score indicating confidence level for classification

Data Pre-Processing Pipeline

- Feature Organization
  - Transit Properties, Threshold-Crossing Event Information, Stellar Parameters, KIC Parameters
  - Binary Target encoding: Candidate (1) / False Positive (0)

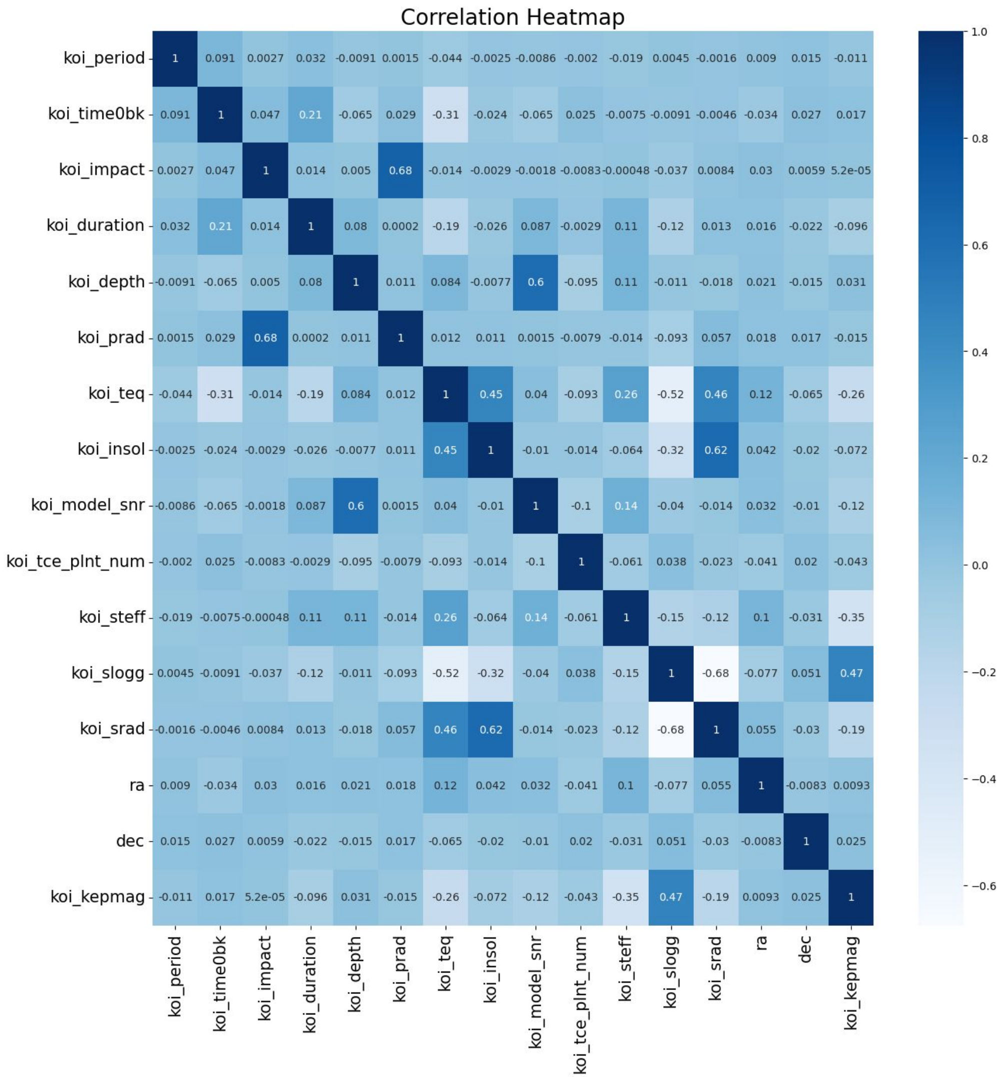
- Train/Val/Test Split (60/20/20)
  - Used random_state = 207 for reproducibility

Final Output: Clean datasets with no missing values. Binary target. Ready for modeling!

Berkeley UNIVERSITY OF CALIFORNIA

Correlation Analysis

- The initial correlation plot showed that nearly all of the error features had ±0.99-±1.00 correlation with the feature.
  - This makes sense as the error should increase proportionally with the measurement.
- As a result, we removed the error columns.
- Right shows the correlation map without error columns.



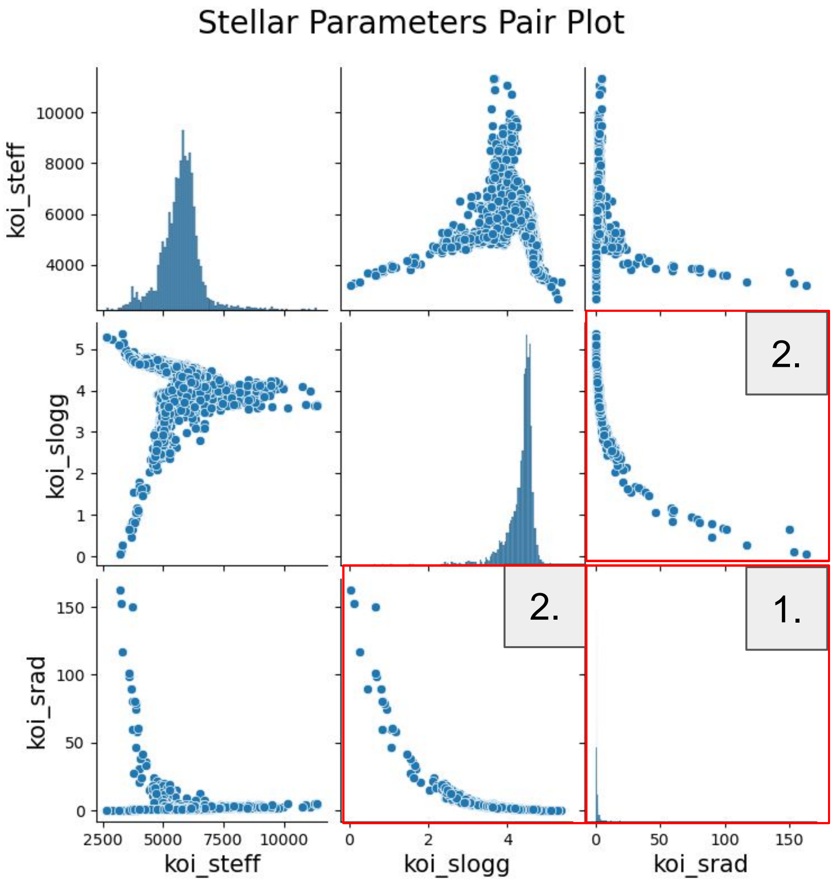
Berkeley
UNIVERSITY OF CALIFORNIA

Pair Plot

- Created pairplots to review distributions of features.
  - Due to large number of features, we grouped them based on the categories of features (stellar parameter features, TCE features, and transit property features).
- Some common patterns amongst features:

1. **Most were heavily right skewed, with large outliers.**
2. **There were four features that displayed exponential/logarithmic growth/decay.**

- As a result, we decided to log transform all of our numeric features and re-evaluate them.



Log Transformed Heatmap

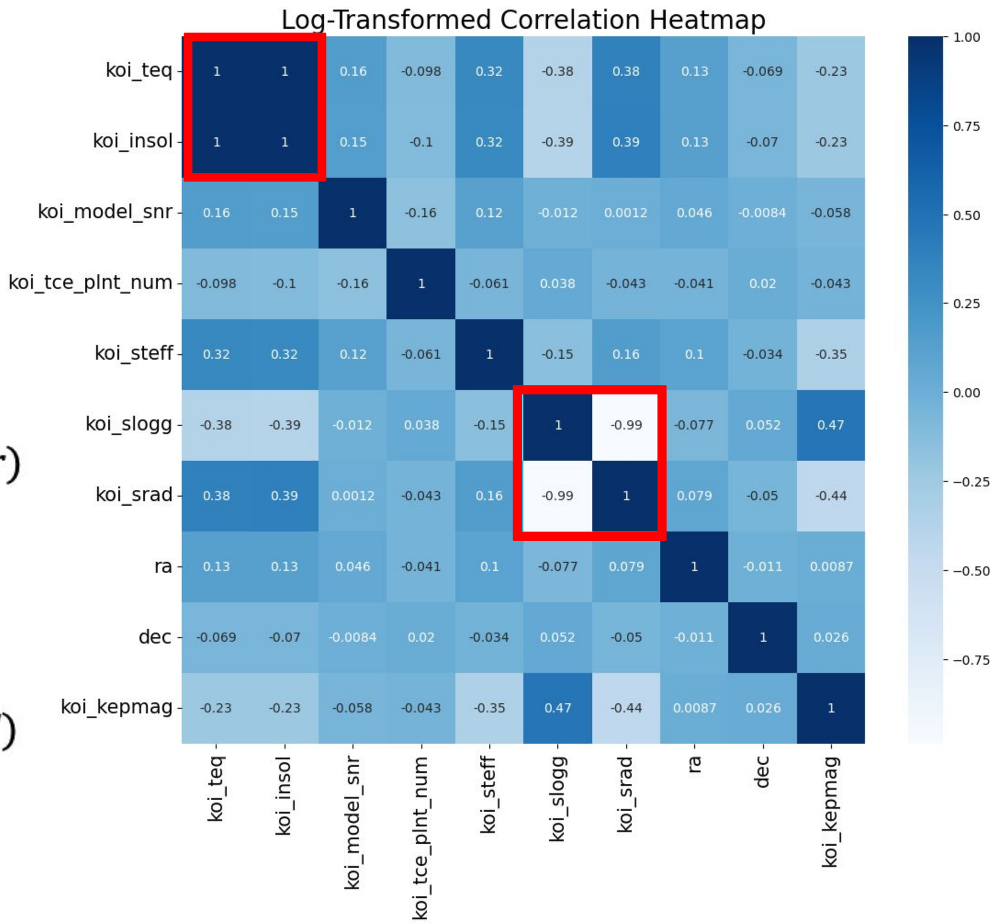
- After Log Transformation, we saw collinearity with 4 features: koi_teq and koi_insol, koi_slogg and koi_srad.
- koi_slogg is the gravitational acceleration of the star and koi_srad is the radius of the star.

$$g = \frac{GM}{r^2} \longrightarrow \log(g) = \log(GM) - 2\log(r)$$

- koi_teq is the equilibrium temperature of the object and koi_insol is the heat flux.

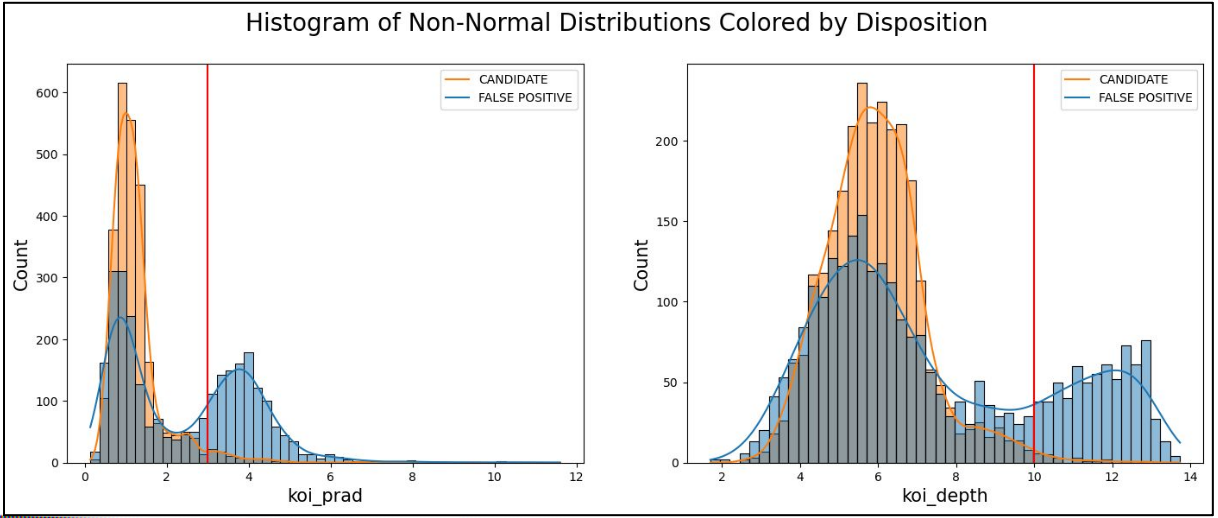
$$q = \sigma\varepsilon T^4 \longrightarrow \log(q) = \log(\sigma\varepsilon) + 4\log(T)$$

- We removed koi_srad and koi_insol to avoid collinearity and redundancy.



Bimodal Distributions

- We observed some bimodal distributions for several of our features and we attempted to bin the features such that the the peaks were in their own bin.
  - Red line is where we set the bin boundaries.
- However, it led to ~1-2% decrease in accuracy for our models, so we did not go down this route.



Missing Values, Scaling & Categorical Values

Missing Values:

- Calculated percentage of missing values
- Dropped rows with missing values in astronomical features (~%4 of data in train, ~3% in validation, ~2% in test)
    - Transit properties (koi_impact, koi_teq)
    - Stellar parameters (koi_steff, koi_slogg, koi_srad)
    - Other (koi_model_snr, koi_tce_plnt_num, koi_tce_delivname)

Scaling:

- Applied log transform to transit properties and specific features
- RobustScaler for numeric columns to handle outliers
- Binned features for non-normal distributions

Missing Values, Scaling & Categorical Values (cont.)

Categorical Values:

- Binary encoded target (Candidate:1, False Positive: 0)
- One-hot encoding categorical variables
    - Koi_tce_delivname (3 categories)
    - Koi_tce_plnt_num (7 categories)

All transformations were applied consistently across train/validation/test sets*

Baseline Model

- We began with exploring a logistic regression model as our baseline
  - Simple linear classifier
  - 5-fold cross validation to assess performance
  - Default parameters
- As a second iteration to our baseline model, we ran a random forest model
  - 500 trees, 5-fold cross validation

Can we improve these with a neural network?

Baseline Model

We will start with a baseline model our more complicated model will compare. We will use a logistic regression as our baseline.

```python
# create logistic regression model
linear_model = LogisticRegression(max_iter=1000)
linear_model.fit(X_train_final, y_train_final)


# use cross validation for determining scores
cv_scores = cross_val_score(linear_model, X_train_final, y_train_final, cv=5)

print('Scores;', cv_scores)
print('Mean Score:', np.mean(cv_scores))

val_pred_lr = linear_model.predict(X_val_final)
val_accuracy_lr = accuracy_score(y_val_final, val_pred_lr)
print('Validation Accuracy:', val_accuracy_lr)
```

✓ 0.2s

```
Scores; [0.80934579 0.80373832 0.8046729  0.81775701 0.78785047]
Mean Score: 0.8046728971962617
Validation Accuracy: 0.8053691275167785
```

Random Forest Model

```python
rf_model = RandomForestClassifier(n_estimators=500)

rf_model.fit(X_train_final, y_train_final)

rf_cv_scores = cross_val_score(rf_model, X_train_final, y_train_final, cv=5)

print('Scores:', rf_cv_scores)
print('Mean Score:', np.mean(rf_cv_scores))

# Validation accuracy
val_pred_rf = rf_model.predict(X_val_final)
val_accuracy_rf = accuracy_score(y_val_final, val_pred_rf)
print('Validation Accuracy:', val_accuracy_rf)
```

✓ 15.1s

```
Scores: [0.8411215  0.8364486  0.84205607 0.85046729 0.82429907]
Mean Score: 0.8388785046728972
Validation Accuracy: 0.8501118568232662
```

Baseline Model Metrics

Logistic Regression Metrics:

CV Scores:
[0.809 0.803 0.804 0.81775701 0.787]
Mean CV Score: 0.804
Validation Accuracy: 0.805

Random Forest Model Metrics:

CV Scores:
[0.841 0.836 0.842 0.850 0.8242]
Mean CV Score: 0.838
Validation Accuracy: 0.835

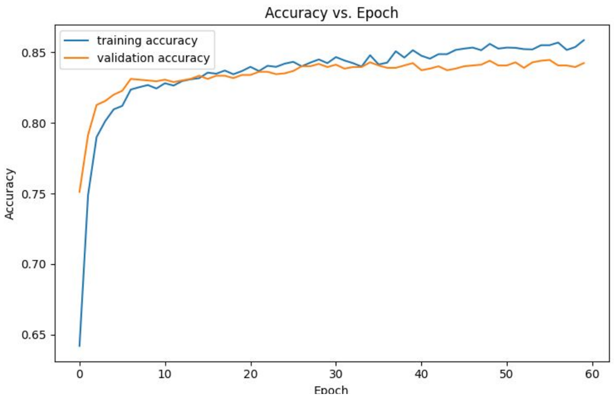
Final Model: Neural Network

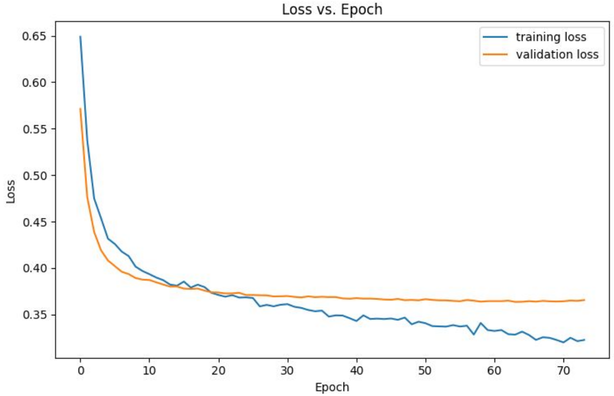
Why we chose a neural network

- Theoretically could capture more complex relationships with our dataset.
- More flexibility in tuning the model.
- Ultimately would scale better with more data.
- Achieved higher accuracy on training and validation sets.

Final validation accuracy: **0.842**

Log. Regression / RFC baseline: **0.804 / 0.835**





Neural Network Hyperparameter Selection

Neural Network Structure:

- **Input Layer:** 24 features
- **4 Dense Hidden Layers:** 256, 128, 64, 32 Neurons
- **3 Dropout Layers:** dropout rate = 0.3 for regularization
- **Output Layer:** Softmax activation

Other Hyperparameters:

- **Learning Rate:** 1e-5
- **Batch Size:** 32
- **Optimizer:** Adam
- **Activation Functions:** ReLU
- **# Epochs:** 1000 (with EarlyStopping callback)

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 24)	0
dense (Dense)	(None, 256)	6,400
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
dense_4 (Dense)	(None, 1)	33

Berkeley
UNIVERSITY OF CALIFORNIA

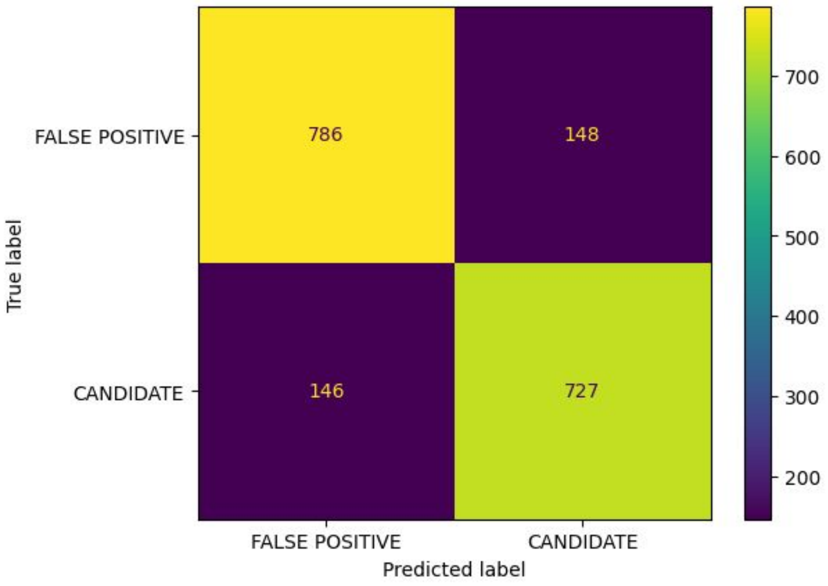
Model Evaluation

Final test accuracy: 83.7% (compared to 84% validation score)

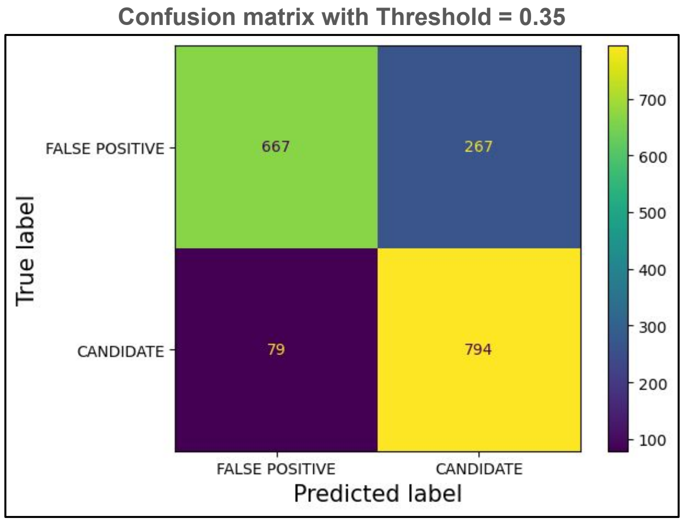
- We see good generalization from training to test.
- Consistent performance across classes.

	Precision	Recall	F1-score
False Positive	0.84	0.84	0.84
Candidate	0.83	0.83	0.83

Confusion matrix w/ threshold 0.5



Threshold Impact on Metrics



Metrics	Threshold = 0.5	Threshold = 0.35
Accuracy	0.84	**0.81**
Candidate Precision	0.82	0.75
Candidate Recall	0.82	**0.91**
False Positive Precision	0.83	**0.89**
False Positive Recall	0.83	0.71

Conclusion

- Summary
  - Started with baseline logistic regression + random forest
  - EDA + feature engineering
  - Final model selection + evaluation
  - Threshold impact on metrics
- Final model
  - Increase in test accuracy by ~1-4%from baseline model
  - Random forest performance comparable to NN
  - Good generalization - similar metrics on train/test set
- Next steps:
  - Better understanding of NASA's original classification methods
  - Better feature engineering
  - More time spent understanding each column

Berkeley
UNIVERSITY OF CALIFORNIA

References

Bilogur, Aleksey, and NASA. “Kepler Exoplanet Search Results.” Kaggle, Google LLC, 10 Oct. 2017, www.kaggle.com/datasets/nasa/kepler-exoplanet-search-results.

NASA Exoplanet Archive. NASA, exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html. Accessed 20 Oct. 2024.

Contributions

- Victoria Brendel: Baseline logistic regression model, background and motivation, data source/structure, conclusion
- Kenneth Hahn: EDA, feature engineering, hyperparameter selection, and threshold optimization
- Moez Hudda: Missing, Categorical & Scaling. Baseline Model
- Matthew Paterno: Training, hyperparameter optimization, final model evaluation

NeurIPS 2021 Paper Checklist Guidelines

1. For all authors...
    a. (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
    b. (b) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
    c. (c) Did you discuss any potential negative societal impacts of your work? Yes
    d. (d) Did you describe the limitations of your work? Yes
2. If you ran experiments...
    a. (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes
    b. (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes
    c. (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? N/A
    d. (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? N/A
    e. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    f. (a) If your work uses existing assets, did you cite the creators? Yes
    g. (b) Did you mention the license of the assets? N/A
    h. (c) Did you include any new assets either in the supplemental material or as a URL?No
    i. (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? Yes
    j. (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes
    k. If you used crowdsourcing or conducted research with human subjects… N/A