DS 440 Capstone Project

Exoplanet Pipeline Efficiency Characterization

Yuya Jeremy Ong & Tomoki Takasawa
Team Astro Boys

Penn State University
College of Information Sciences and Technology

Outline

1. Background
2. Data Processing Pipeline
3. Phase 1: PLTI Dataset Modeling
4. Phase 2: TCE Dataset Modeling
5. Model Analysis & Discussions
6. Contributions & Conclusions

Background

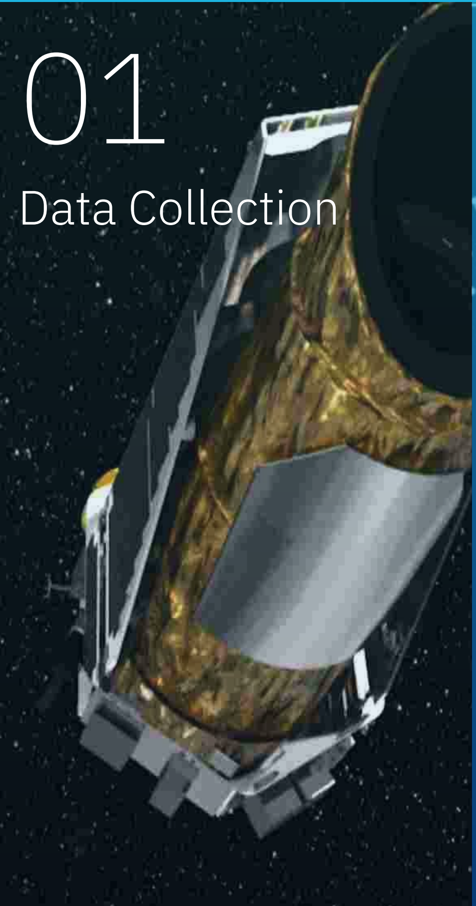
01
Data Collection

02
Manual Analysis

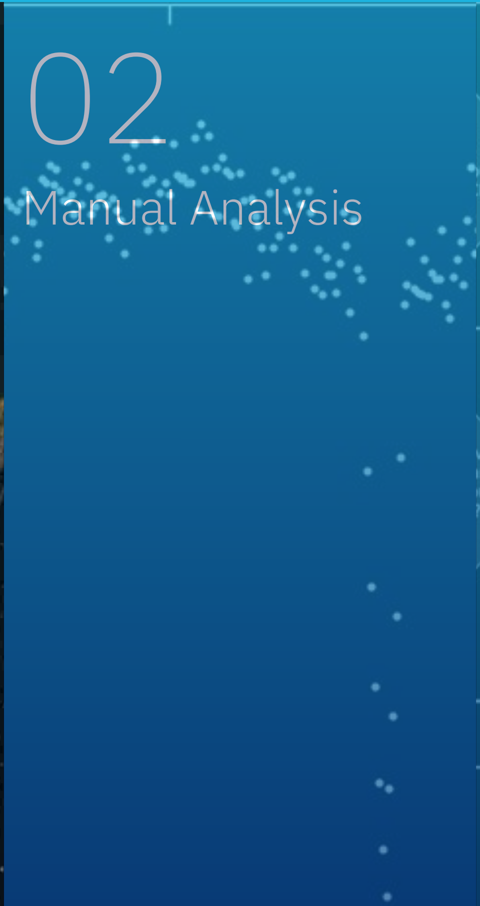
03
Robovetter:
Auto-Heuristics

04
Injection Simulation

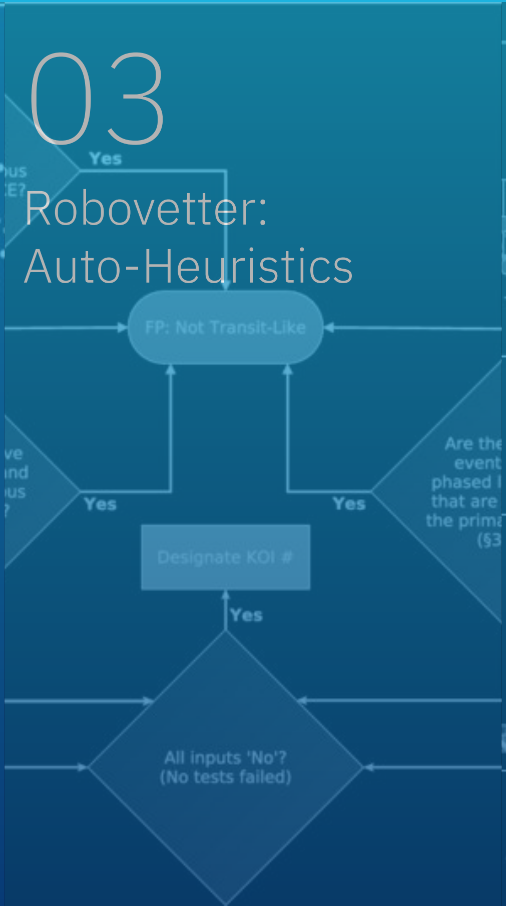
Background



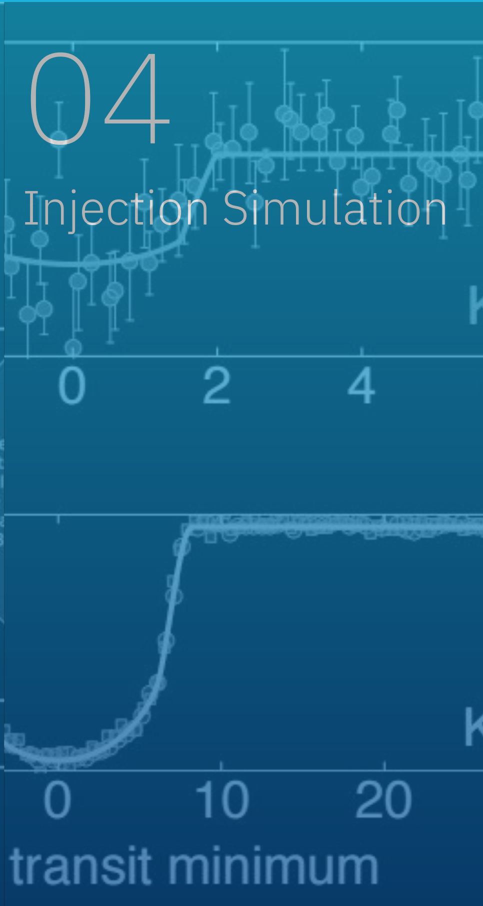
01
Data Collection



02
Manual Analysis

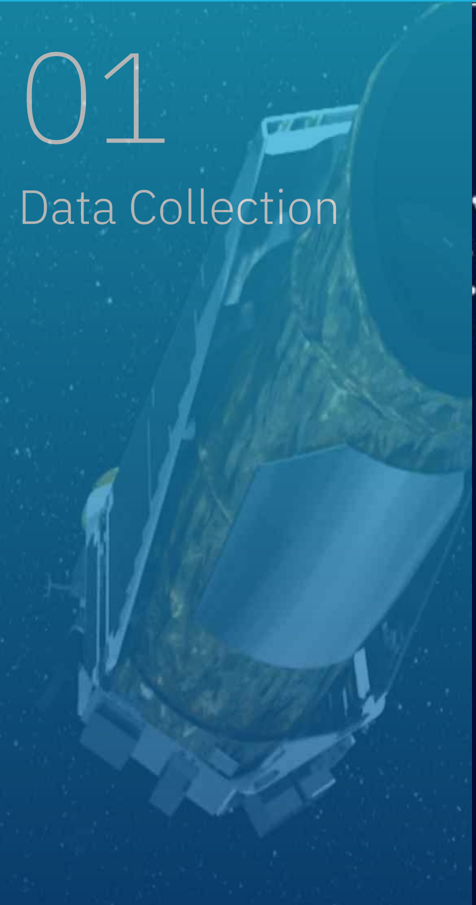


03
Robovetter:
Auto-Heuristics



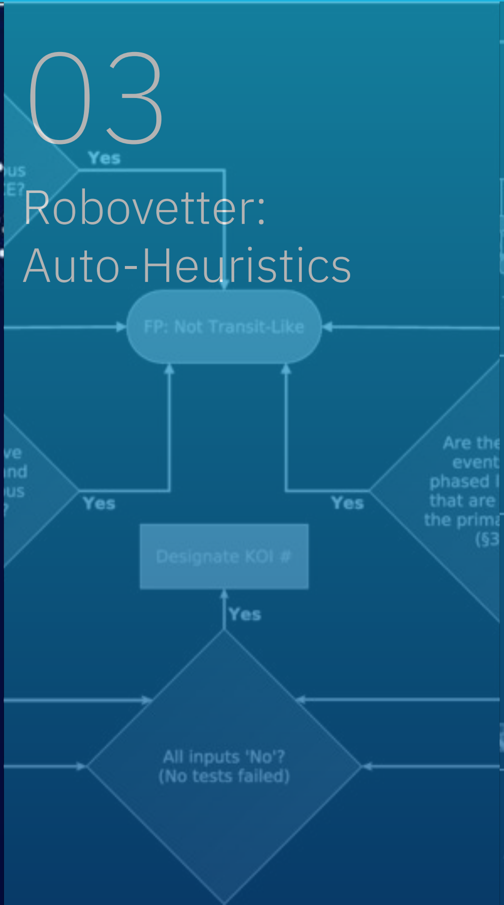
04
Injection Simulation

Background



01
Data Collection

02
Manual Analysis

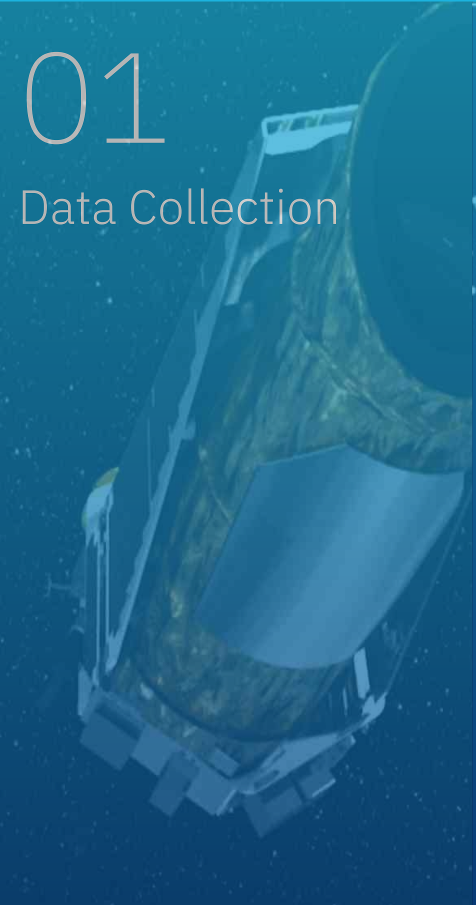


03
Robovetter:
Auto-Heuristics

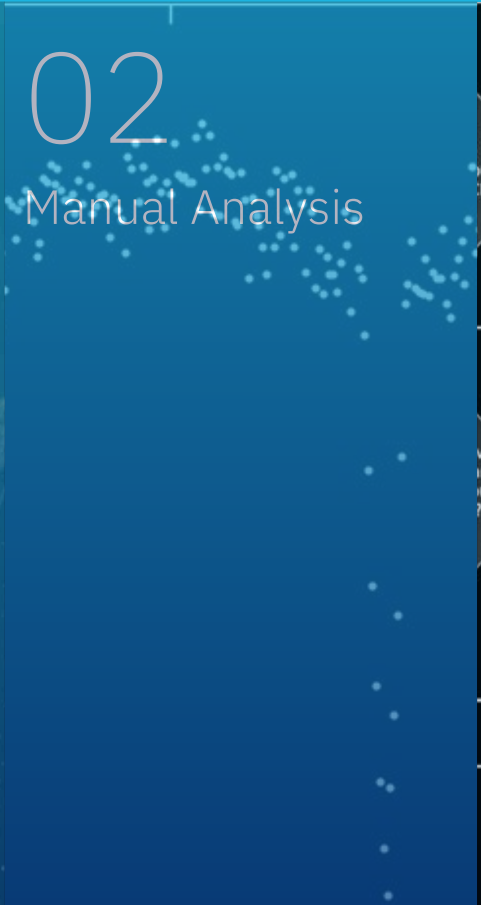
04
Injection Simulation

Background

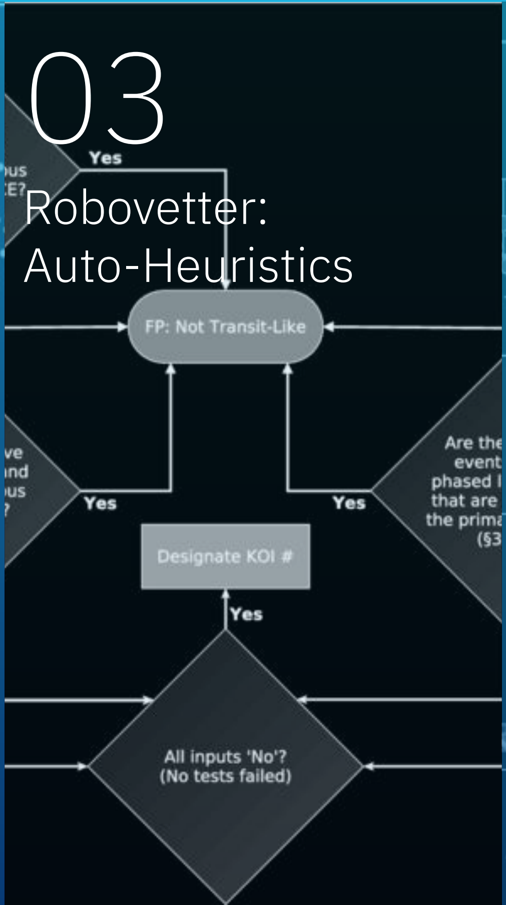
01
Data Collection



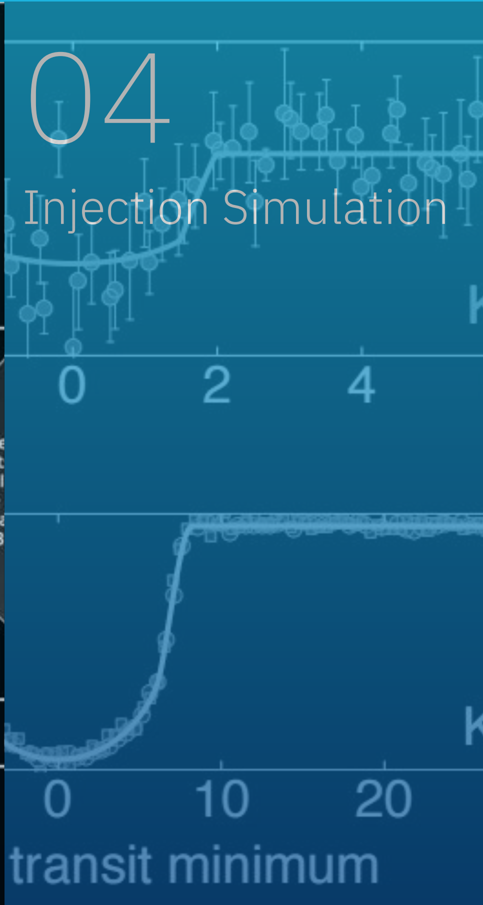
02
Manual Analysis



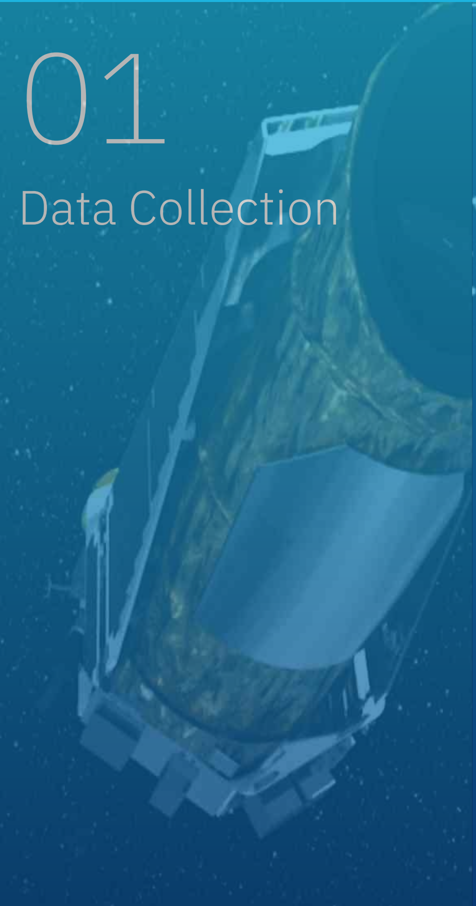
03
Robovetter:
Auto-Heuristics



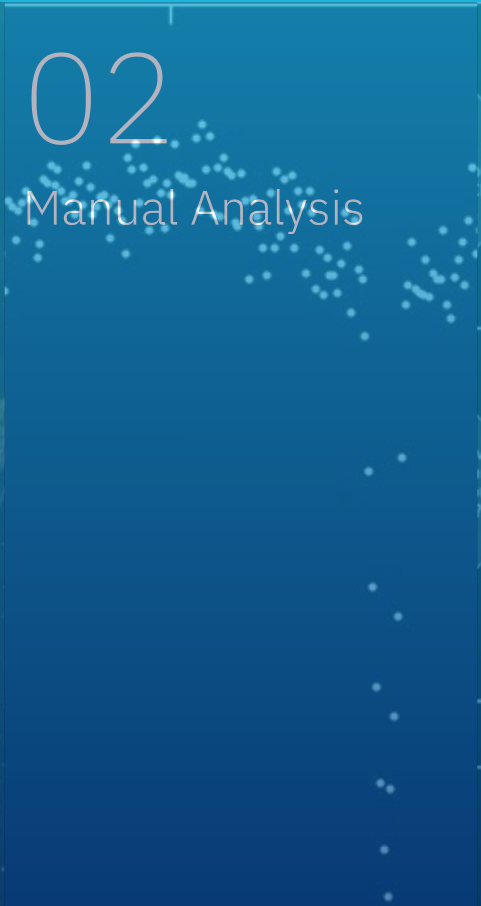
04
Injection Simulation



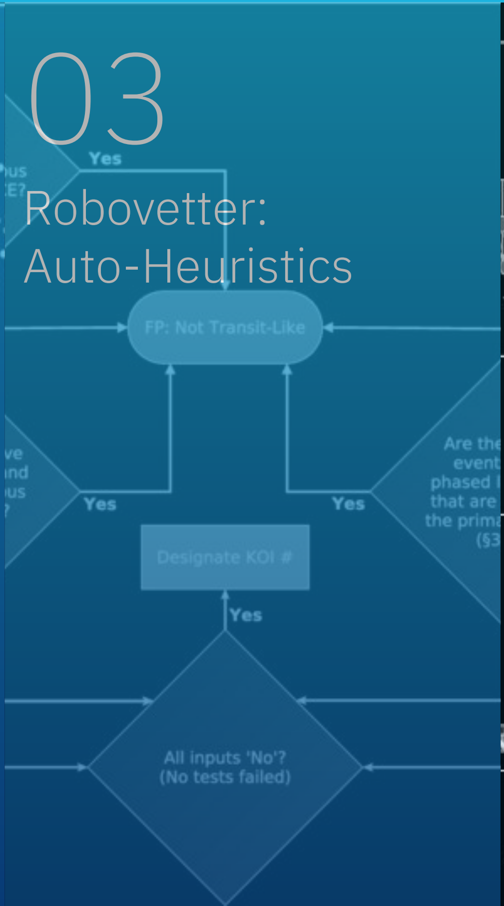
Background



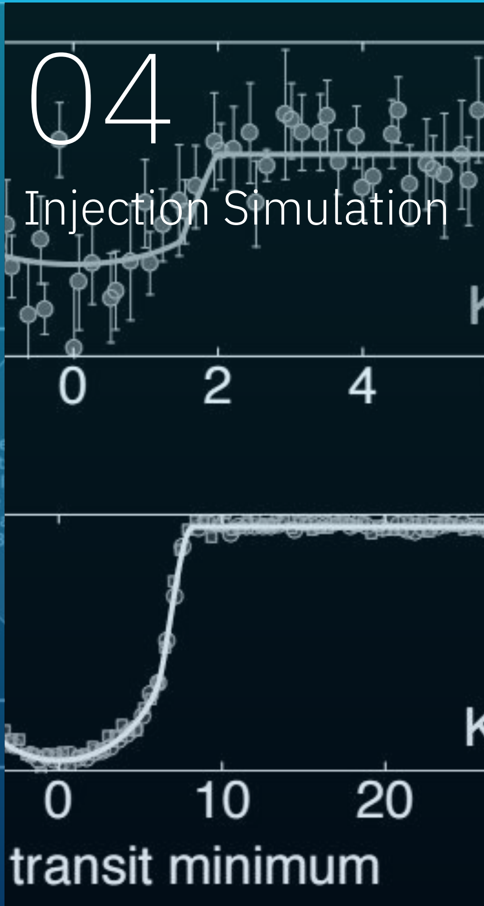
01
Data Collection



02
Manual Analysis



03
Robovetter:
Auto-Heuristics



04
Injection Simulation

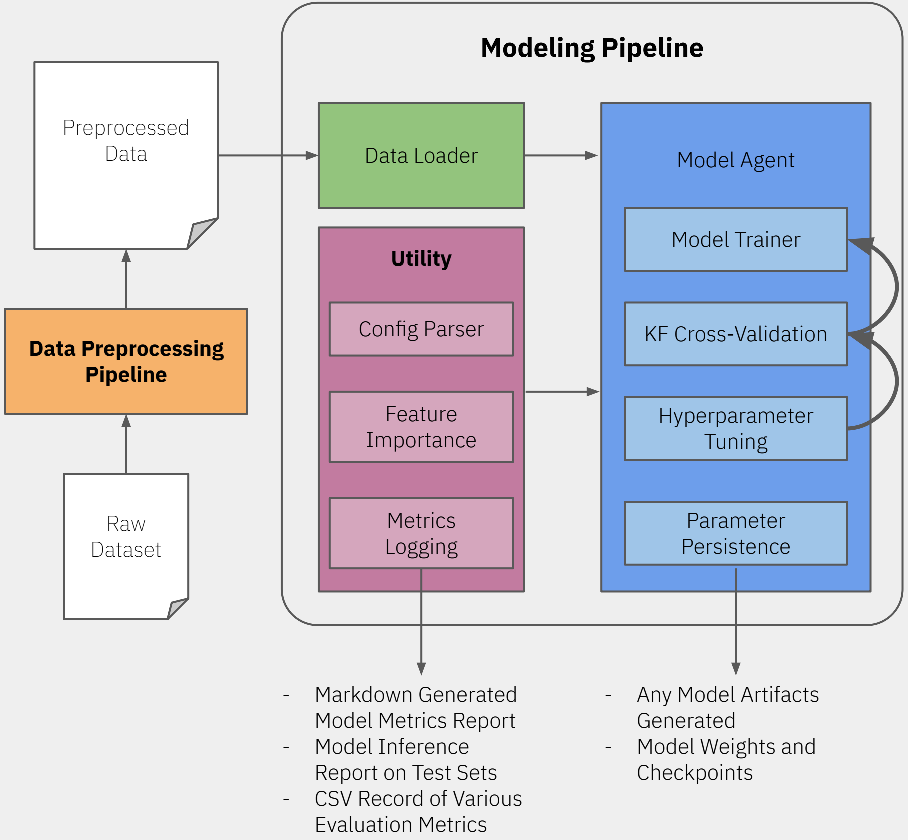
Data Processing Pipeline

- We present our iterative data-driven research platform for **maximizing efficiency and model throughput**.

- **Scalable** and **modular design** for accommodating any supervised modeling tasks (*with reusable components*).

- Auto-generated carbon-copy configuration files, logging, and metrics - allows for **easy debugging**, **interpretability & interoperability**, and **reproducibility**.

- **Key Takeaway**: Mitigate technical debt *as early as possible* for long-term gains in modeling process efficiency.



Sample Pipeline Model Artifacts

We auto generate model artifacts for **EVERY** single model ever produced.

```json
{
    "model": "ada",
    "out_dir": "log",

    "data": {
        "plti": "data/raw/plti/kplr_dr25_inj1_plti.txt",
        "noise": "data/raw/misc/DR25_DEModel_NoisyTargetList.txt",

        "features": ["sg_ori_pool", "sg_corner_pool", "i_period", "i_epoch",
            "N_Transit", "i_depth", "i_dur", "i_b", "i_ror", "i_dor", "Expected_MES"],

        "feat_trans": true,
        "ifs": true,
        "pca": false,
        "drop_noise": false
    }
}
```

Figure 1: Reproducible JSON-based carbon-copy configuration files. Throw it back in the pipeline to get the SAME exact results.

ada_score-report.md Preview

ada

Model Performance Score Report

K-Fold Classification Report

K	RMSE	Log Loss	Accuracy	Precision	Recall	F-Measure	AUC	Kappa
1	0.1516086381665932	5.236476230200349	0.8483913618334068	0.8168421205653422	0.8483913618334068	0.8185469044192025	0.5871748480428314	0.2330682193903445
2	0.14103129131776113	4.871145145731478	0.8589687086822388	0.8296322272707404	0.8589687086822388	0.8265676274208562	0.581543272486586	0.226723864335876
3	0.1505068312031732	5.198426099471723	0.8494931687968268	0.819588180672713	0.8494931687968268	0.8143191298142606	0.5792304884302382	0.2201811410608712 8
4	0.13552225650066108	4.680867885827445	0.8644777434993389	0.8367455694143119	0.8644777434993389	0.8312496680607112	0.5787633705740229	0.22306764219693642
5	0.14499779638607316	5.00814725376406	0.8550022036139269	0.8244333497727765	0.8550022036139269	0.8200740513112883	0.57552475509425	0.21214030532820405
6	0.1397091229616571	4.825480125725995	0.8602908770383428	0.8320766585465073	0.8602908770383428	0.8260690661568174	0.578052367666657	0.22055079315970527
7	0.14367562802996914	4.962484171963013	0.8563243719700309	0.8297548982920897	0.8563243719700309	0.8175968978552226	0.5704785722617173	0.20362007513830704
8	0.14238483579457792	4.917899166858549	0.857615164205422	0.8293813344935665	0.857615164205422	0.8210826897307888	0.5734957584875564	0.20987210337407491
9	0.12056424950407758	4.164224134805704	0.8794357504959224	0.8510152123456604	0.8794357504959224	0.8509444640639242	0.5787975597397812	0.22284712747305135
10	0.12827859819263832	4.430673762608704	0.8717214018073617	0.844975266309405	0.8717214018073617	0.8425745720048999	0.5869858021692045	0.24242926753134686

Average Confusion Matrix

	Pred POS	Pred NEG
True POS	3782.6	84.2
True NEG	550.3	120.6

Average Model Performance Metrics

RMSE	LOGLOSS	ACC	PRE	REC	F1	AUC	KAPP
0.1398279248057182	4.829582397695702	0.8601720751942817	0.8314444817683114	0.8601720751942817	0.8269025070837973	0.5790046794952846	0.2214500538988718

AUC/ROC Plot

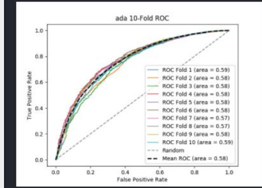


Figure 2: Human-Readable Auto-Generated Markdown Logs

PHASE 01:

Evaluate the Efficiency of Signal Recovery of Robovetter’s Transit Cross Event (TCE) Detection Heuristics

→ *Generate a Probabilistic Model to Predict Efficiency*

PLTI Injection 1 Dataset

Pixel Level Transit Injection: Augmentation of the light-curve data at the raw pixel-level

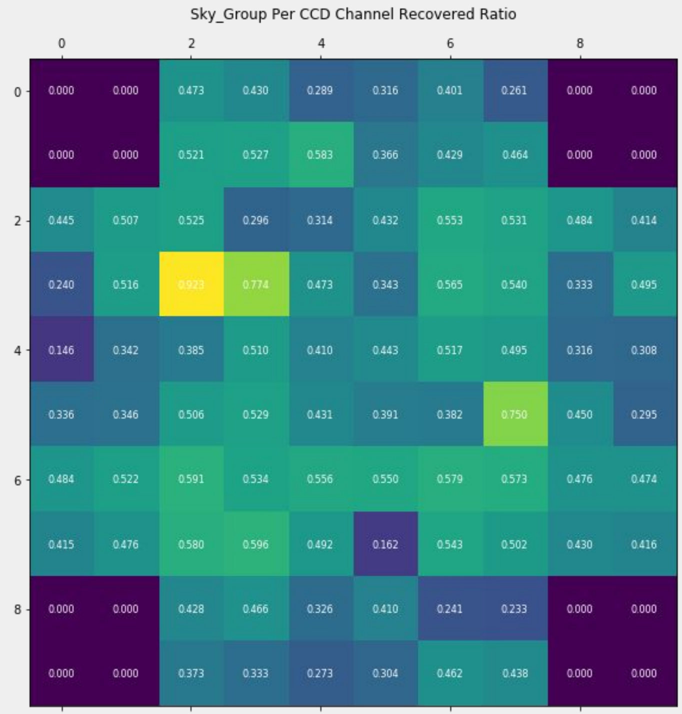
Dimension: 146294 Records, 25 Columns

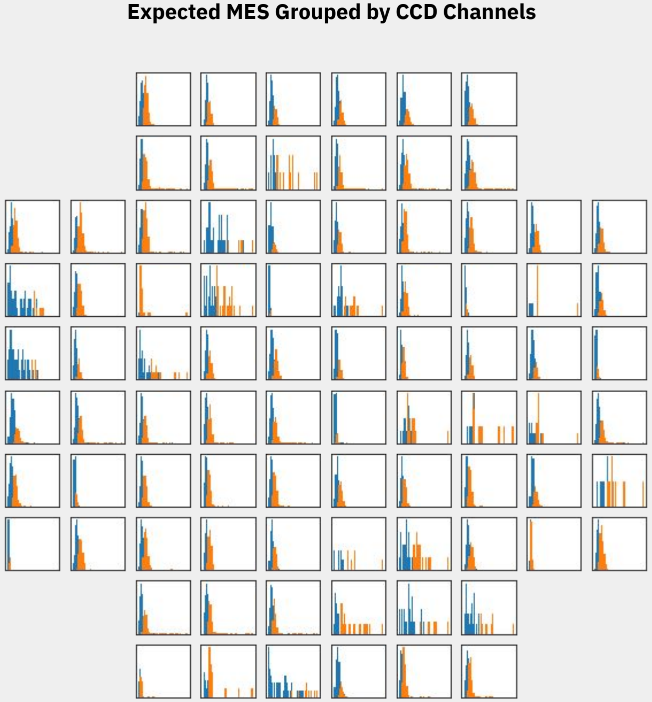
NaN Count

```
df.isna().sum()
```

KIC_ID	0
Sky_Group	0
i_period	0
i_epoch	0
N_Transit	0
i_depth	0
i_dur	0
i_b	0
i_ror	0
i_dor	0
EB_injection	0
Offset_from_source	0
Offset_distance	0
Expected_MES	37
Recovered	0
TCE_ID	100917
Measured_MES	100917
r_period	100917
r_epoch	100917
r_depth	100917
r_dur	100917
r_b	100917
r_ror	100917
r_dor	100917
Fit_Provenance	100917

dtype: int64





PLTI INJ 1 Preprocessing

Three Target Label Issue
- Dataset included three target labels (0, 1, 2)

- According to (Christiasen, 2015) this second value indicates (quoted from Dr. Ford):

"… ***instead of finding a planet with an orbital period close to the period of the injected planet***,… "

- We replace all instances of "2" as "1".

Preprocessing Methods
- Dropped 37 Missing (N/A) Expected_MES records.

- Performed Standardized Scaling over each feature of the dataset.

Feature Importance Evaluation Methods

We evaluated the feature importances against:

1. Random Forests
2. AdaBoost Classifiers
3. Extra Trees Classifiers
4. Gradient Boosting Classifiers
5. Random Trees Embedding
6. Chi-Squared Feature Selection
7. Lasso Feature Selection [Return 0 or 1]

Note:

- Applied Over Entire Dataset (No CV Splits)
- Utilized 1000 Estimators for Classifier Models
- For Lasso, utilized 5-Fold Cross Validation

- *ONLY interested in Feature Importance values!*

Rank Aggregation Algorithm

Given the list of multiple ranks with different metrics for each ranks, we utilized a **Rank Aggregation** method from Information Retrieval proposed by Dwork et al.

Key Takeaway:

Provide an "averaged out" consensus of the feature importance from multiple sources.

PLTI INJ 1 Feature Importances

Table 1: Derived Feature Importance and Selection Metrics

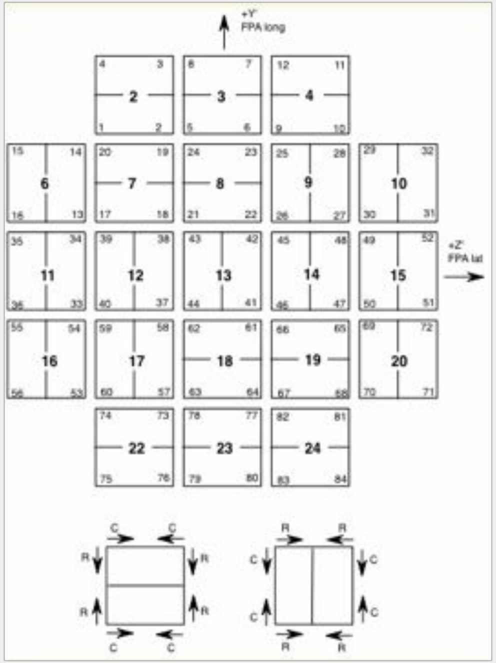
	AdaBoost	Extra Trees	GBM	Lasso	Random Forest	Random Trees	Chi Squared
sky_group	0.009	0.0224	0.0155	0	0.0223	0.0390	1.682131843
i_period	0.109	0.0314	0.0508	1	0.0507	0.0438	24.70575607
i_epoch	0.059	0.0192	0.0511	0	0.0298	0.0476	41.21114159
N_Transit	0.049	0.0609	0.0402	0	0.0735	0.0628	23.67576581
i_depth	0.04	0.0223	0.0315	0	0.0337	0.0569	130.6990697
i_dur	0.06	0.00463	0.0518	0	0.0549	0.0352	249.1171421
i_b	0.051	0.0543	0.0291	0	0.0308	0.0391	26.32305742
i_ror	0.033	0.0264	0.0233	0	0.0323	0.0495	119.2648146
i_dor	0.044	0.0362	0.0456	1	0.0415	0.0369	48.73736431
Expected_MES	0.075	0.3999	0.18	1	0.2751	0.0428	179.6021226

PLTI INJ 1 Aggregated Feature Importances

1. Expected MES
2. i_dur
3. i_period
4. N_Transits
5. i_dor
6. i_depth
7. i_epoch
8. i_ror
9. i_b (Impact Parameter)
10. Sky Group

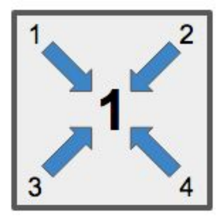
Can We Squeeze More Out of Sky Group?

Question: Can we utilize spatial characteristics of the CCD channel to extract much more useful features?

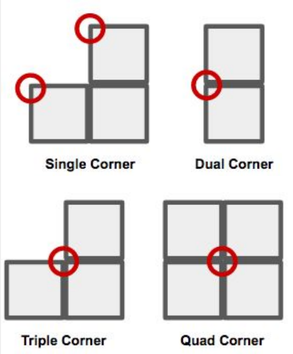


Our Inspiration:

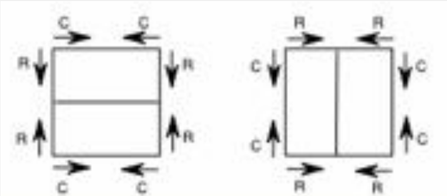
Pooling Operation from Convolutional Neural Networks



Center Pooling



Corner Pooling



Orientation Pooling

PLTI INJ 1 Baseline Models

Expected MES Heuristics

As a lower-bound baseline, we implement the method by Christiansen, 2017.

Defined by the following Cumulative Distribution Function (CDF) of the Gamma Distribution:

$$p = F(x|a,b,c) = \frac{c}{b^a\Gamma(a)}\int_0^x t^{a-1}e^{-t/b}dt$$

Given the following parameters:

a = 30.87

b = 0.271

c = 0.940

Models Implemented

- Logistic Regression
- Decision Tree
- Naive Bayes (Gaussian)
- Naive Bayes (Bernoulli)
- Random Forest
- Stochastic Gradient Descent Classifier
- Multi-Layer Perceptron
- Extreme Gradient Boosting
- Categorical Boosting
- K-NN Classifier
- Ensemble Strategies of Top 3 & 5 Best Models (Voting & Stacking w/ Logistic Meta-Model)

* Used package-defined default parameter as baselines for models used.

Model Evaluation Methodology and Metrics

- For each model we have implemented, we performed a **10-Fold Cross Validation**.

- Fixed PRNG hyperparameter used for all randomized effects.

- For each fold, we correspondingly generate AUC plots (and also persist raw values).

- We also compute an averaged confusion matrix from each of the 10-folds.

Evaluation Metrics

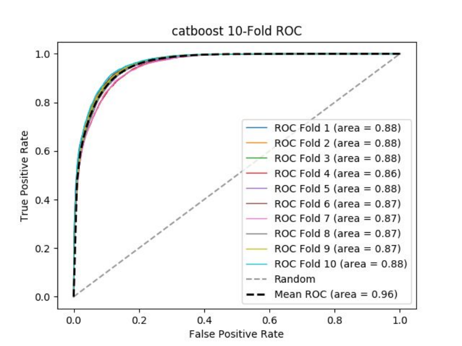
- RMSE
- Log Loss
- Accuracy
- Precision
- Recall
- F-Measure (F1 Score)
- AUC
- Kappa Fleiss Statistics

PLTI INJ 1 Results

Baseline Results

Table 3: Base Model Results

Model	RMSE	Log Loss	Accuracy	Precision	Recall	F1	AUC	Kappa
CatBoost	0.1081	3.7331	0.8919	0.8246	0.8274	0.8259	0.8739	0.7473
Adaboost Classifier	0.1112	3.8391	0.8888	0.8075	0.8425	0.8246	0.8758	0.7430
XGBoost	0.1120	3.8670	0.8880	0.8144	0.8276	0.8209	0.8712	0.7392
Random Forest	0.1200	4.1449	0.8800	0.8303	0.7704	0.7991	0.8496	0.7135
Logistic Regression	0.1306	4.5101	0.8697	0.8261	0.7338	0.7770	0.8320	0.6852
MLP	0.1320	4.5599	0.8690	0.8215	0.7379	0.7773	0.8329	0.6847
NB Bernoulli	0.1446	4.9946	0.8554	0.7311	0.8436	0.7832	0.8521	0.6753
SGD Classifier	0.1511	5.2184	0.8377	0.7995	0.6920	0.7141	0.7953	0.6042
Decision Tree	0.1562	5.3952	0.8447	0.7492	0.7498	0.7494	0.8184	0.6366
K-NN Classifier	0.1684	5.8178	0.8415	0.8299	0.6178	0.7043	0.7808	0.6008
Baseline Gamma	0.3083	10.6496	0.6919	0.7851	0.0092	0.0182	0.5040	0.0111
NB Gaussian	0.5902	20.3863	0.4098	0.3389	0.9503	0.4992	0.5585	0.0781



	Pred POS	Pred NEG
True POS	3756.2	781.5
True NEG	799.7	9292.0

Key Takeaway: Categorical Boosting model performed the best

PLTI INJ 1 Results

Sky Group-Feature Results

Model	RMSE	Log Loss	Accuracy	Precision	Recall	F1	AUC	Kappa
CatBoost Baseline	0.1080836938	3.733122242	0.8919163062	0.8246439868	0.8273564727	0.8259440368	0.8739162617	0.7472974926
CatBoost (SG Feat)	0.1082067404	3.737372348	0.8917932596	0.8240345302	0.8278986607	0.8258674712	0.8739818182	0.7471120454
CatBoost (No SG)	0.1075778623	3.715651425	0.8924221377	0.8250630442	0.8289142661	0.8269166362	0.8747008214	0.748602517

Key Takeaway: Performed worse than previous Categorical Boosting baseline.
Removing Sky_Group performs the BEST

PLTI INJ 1 Results

Ensemble Models

Model	RMSE	Log Loss	Accuracy	Precision	Recall	F1	AUC	Kappa
Vote Ensemble (T3)	0.1092594084	3.773731252	0.8907405916	0.82071348	**0.8287005423**	0.8246174013	0.8734364656	0.7450039851
Vote Ensemble (T5)	0.110277901	3.80890621	0.889722099	**0.826808082**	0.8152154887	0.8208816793	0.868988676	0.7409565655
Stack Ensemble (T3)	**0.1080836938**	**3.733122242**	**0.8919163062**	0.8246439868	0.8273564727	**0.8259440368**	**0.8739162617**	**0.7472974926**
Stack Ensemble (T5)	0.1208525188	4.174137874	0.8791474812	0.8279507075	0.7701657095	0.7979569938	0.8489175992	0.7116208242

Key Takeaway: Similar performance to best performing model (CatBoost).
Categorical Boosting was a bottleneck in our ensemble.

PHASE 02:

Evaluate the Efficiency of Signal Recovery of Robovetter’s Detection for False Positive Candidates

→ *Generate a Probabilistic Model to Predict Efficiency*

TCEs Dataset

Pixel Level Transit Injection: Augmentation of the light-curve data at the raw pixel-level

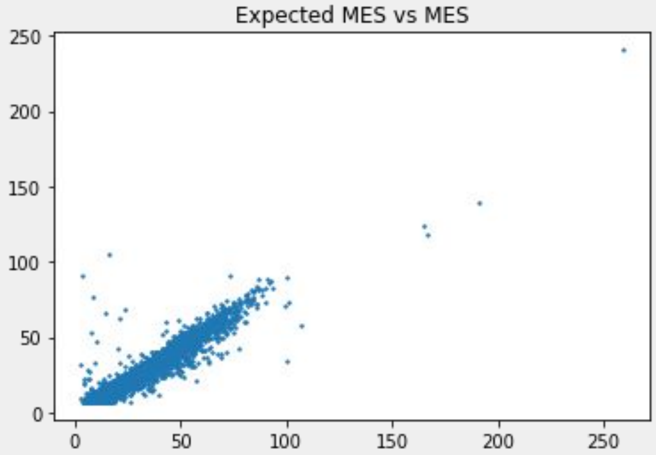
Dimension: 146294 Records, 25 Columns

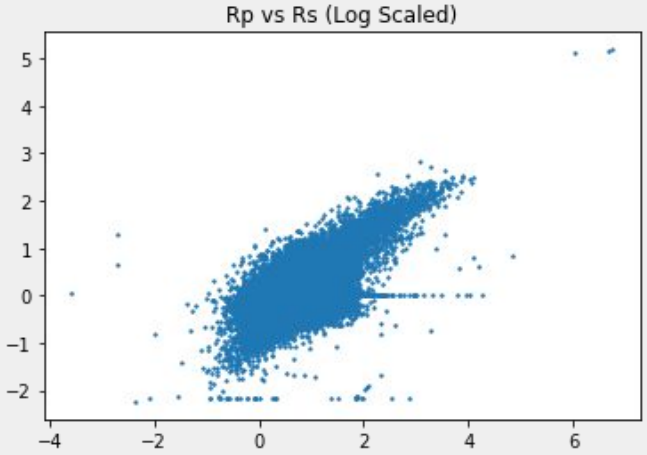
NaN Count

Identify the count of NaN values in the dataset.

```
df.isna().sum()
```

```
TCE_ID          0
KIC             0
Disp            0
Score           0
NTL             0
SS              0
CO              0
EM              0
period          0
epoch           0
Expected_MES    0
MES             0
NTran           0
depth           0
duration        0
Rp              0
Rs              0
Ts              0
logg            0
a               0
Rp/Rs           0
a/Rs            0
impact          0
SNR_DV          0
Sp              0
Fit_Prov        0
dtype: int64
```





TCEs Dataset

Pixel Level Transit Injection: Augmentation of the light-curve data at the raw pixel-level

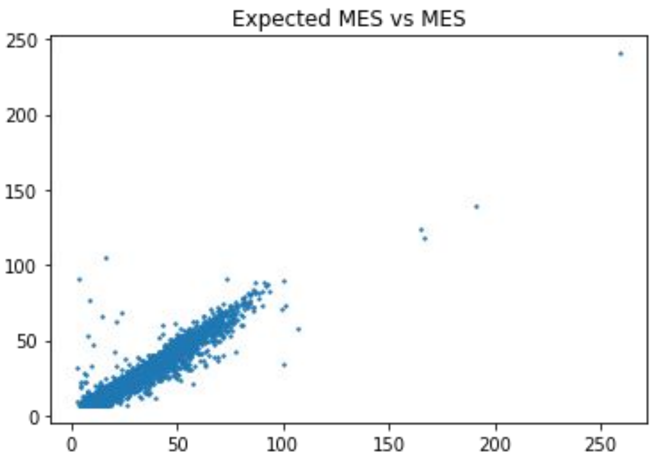
Dimension: 146294 Records, 25 Columns

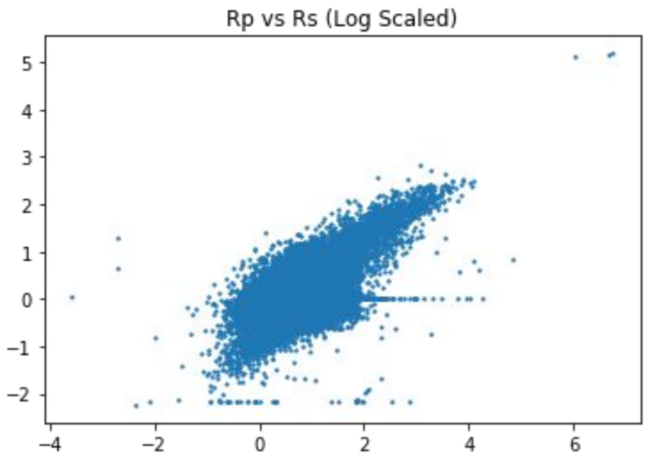
NaN Count

Identify the count of NaN values in the dataset.

```
df.isna().sum()
```

```
TCE_ID          0
KIC             0
Disp            0
Score           0
NTL             0
SS              0
CO              0
EM              0
period          0
epoch           0
Expected_MES    0
MES             0
NTran           0
depth           0
duration        0
Rp              0
Rs              0
Ts              0
logg            0
a               0
Rp/Rs           0
a/Rs            0
impact          0
SNR_DV          0
Sp              0
Fit_Prov        0
dtype: int64
```





TCEs Log Transforms

We perform some log-based transformations to the dataset for improved scaling of our data.

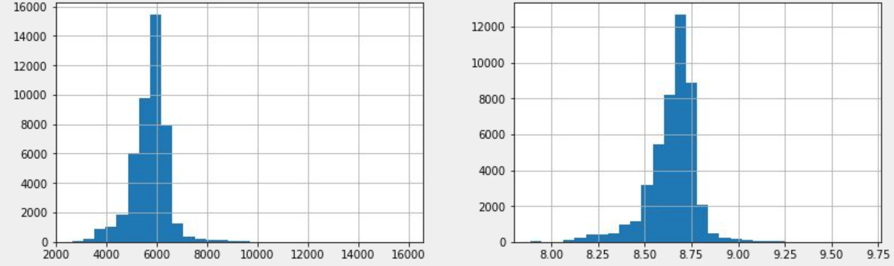
We independently evaluated our transformations for each feature and observed its performance.

Used a Logistic Regression based model (similar modeling pipeline to Phase 1).

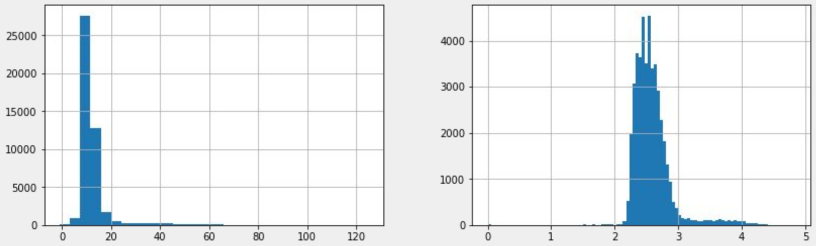
Figure 2: List of Features Based on Accuracy (Higher the Better)

1.	Ts	[0.8593]
2.	SNR_DV	[0.8580]
3.	Depth	[0.8576]
4.	NTran, Rs	[0.8575]
5.	a/Rs, impact	[0.8574]
6.	Baseline	[0.8573]

TS Log Transforms



SNR_DV Log Transforms



TCEs Models

Models Implemented

- Logistic Regression
- Decision Tree
- Naive Bayes (Gaussian)
- Naive Bayes (Bernoulli)
- Random Forest
- Stochastic Gradient Descent Classifier
- Multi-Layer Perceptron
- Extreme Gradient Boosting
- Categorical Boosting
- K-NN Classifier

* Used package-defined default parameter as baselines for models used.

Model Variants Implemented

- Regular Baseline

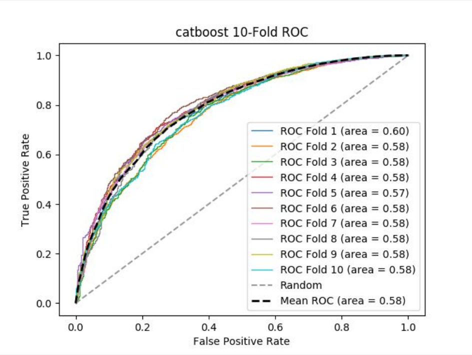
- Log Transformed Baseline

Currently in the process of working with more models and feature engineering methods.

TCEs Results

Baseline Results

Model	RMSE	Log Loss	Accuracy	Precision	Recall	F1	AUC	Kappa
AdaBoost	0.1398059	4.8288213	0.8601941	0.8314928	0.8601941	0.8269183	0.5790176	0.2215057
CatBoost	0.1348033	4.6560388	0.8651967	0.8427900	0.8651967	0.8302303	0.5811168	0.2324654
Decision Tree	0.2157695	7.4524965	0.7842305	0.7922355	0.7842305	0.7880072	0.5903936	0.1740383
Random Forest	0.1349797	4.6621302	0.8650203	0.8422015	0.8650203	0.8303425	0.5815998	0.2329431
K-Nearest Neighbors	0.1441032	4.9772619	0.8558968	0.8325828	0.8558968	0.7979631	0.5209794	0.0681392
Logistic Regression	0.1426707	4.9277810	0.8573293	0.8312126	0.8573293	0.8052391	0.5329934	0.1041091
MLP	0.1470783	5.0800218	0.8529217	0.7471708	0.8529217	0.7873895	0.5048019	0.0154006
Naive Bayes (Bernoulli)	0.1823825	6.2993580	0.8176175	0.7962029	0.8176175	0.8052425	0.5816848	0.1839217
Naive Bayes (Gaussian)	0.1498550	5.1759100	0.8501450	0.8112269	0.8501450	0.8150524	0.5611260	0.1692185
Random Forest	0.1363902	4.7108471	0.8636098	0.8391171	0.8636098	0.8281957	0.5777990	0.2228036
SGDC	0.2032387	7.0197240	0.7967613	0.7733614	0.7967613	0.7394661	0.5089710	0.0103582
XGBoost	0.1361036	4.7009512	0.8638964	0.8426802	0.8638964	0.8245737	0.5686242	0.2024474



	Pred POS	Pred NEG
True POS	3806.7	60.1
True NEG	551.6	119.3

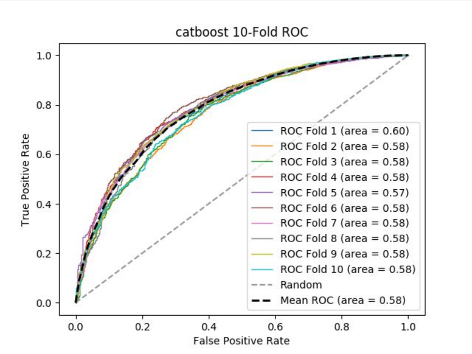
Key Takeaway: Categorical Boosting model performed the best, However Decision Tree's AUC is higher... (why?)

TCEs Results

Log Transformed Results

Figure 3: Benchmarks for Log Based Transformed Features

	RMSE	Log Loss	Accuracy	Precision	Recall	F1	AUC	Kappa
AdaBoost	0.13983	4.82958	0.86017	0.83144	0.86017	0.82690	0.57900	0.22145
CatBoost	0.13480	4.65604	0.86520	0.84279	0.86520	0.83023	0.58112	0.23247
Decision Tree	0.21520	7.43271	0.78480	0.79290	0.78480	0.78865	0.59178	0.17664
Random Forest	0.13595	4.69562	0.86405	0.84012	0.86405	0.82905	0.57955	0.22732
KNC	0.14093	4.86765	0.85907	0.84005	0.85907	0.80744	0.53623	0.11436
Logistic Reg.	0.13844	4.78164	0.86156	0.84653	0.86156	0.81288	0.54518	0.14106
MLP	0.13780	4.75956	0.86220	0.83864	0.86220	0.82118	0.56263	0.18591
NB (Bernoulli)	0.17921	6.18975	0.82079	0.79758	0.82079	0.80711	0.58207	0.18742
NB (Gaussian)	0.14792	5.10893	0.85208	0.81320	0.85208	0.81456	0.55744	0.16278
Random Forest	0.13608	4.70019	0.86392	0.83981	0.86392	0.82859	0.57842	0.22464
SGDC	0.20790	7.18062	0.79210	0.82990	0.79210	0.74776	0.55428	0.15526
XGBoost	0.13610	4.70095	0.86390	0.84268	0.86390	0.82457	0.56862	0.20245



	Pred POS	**Pred NEG**
True POS	3806.7	60.1
True NEG	551.6	119.3

Key Takeaway: Marginally Improved Results - Still Same Behavior with Categorical Boost vs Decision Tree

Key Questions

Based on the documentation from Christiansen, 2017, we attempt to investigate some of the following key questions/observations raised:

1. Should we *exclude duration times over 15 hours* from the data pipeline?

2. What is the ideal threshold value for the *number of valid transits*?

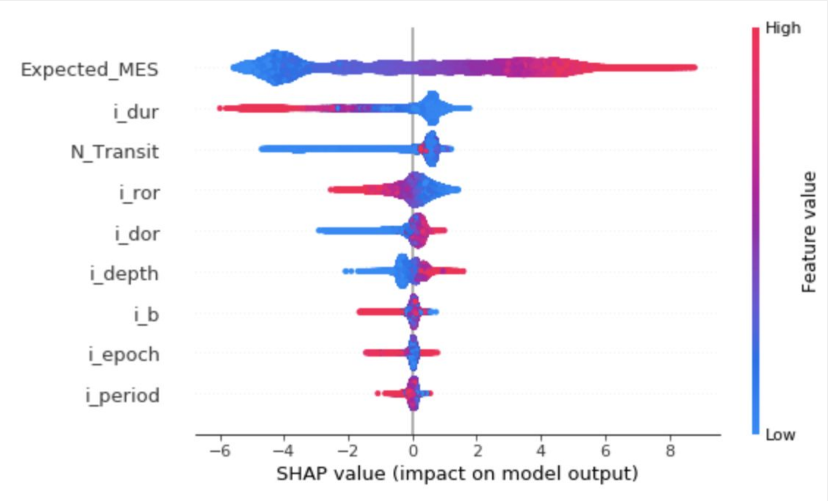
SHAP Analysis

- **Shapely Additive Explanation Values** help to explain how features contribute to the outcome of the model. *Helps with model interpretability.*

- Perform analysis over our best performing model, Categorical Boosting, using 10-Fold CV.

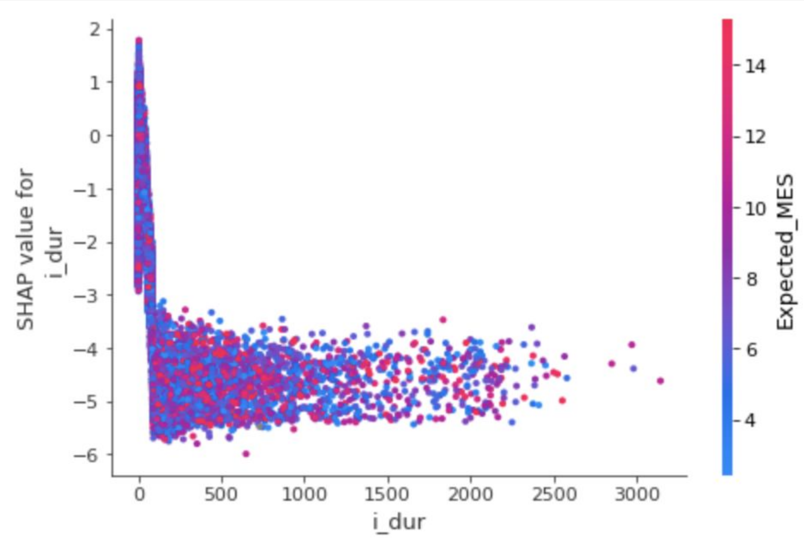
- Performed analysis including and excluding Expected MES to see the effects of the other features contributions to the model.

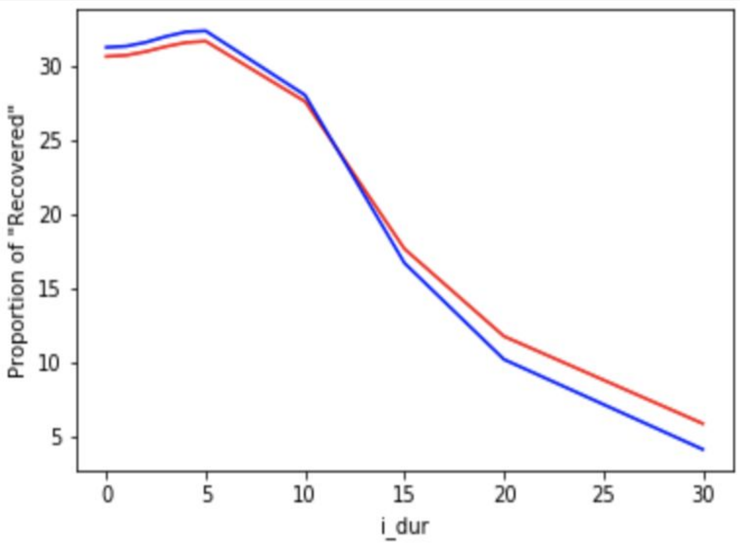
- We use SHAP Analysis to address some of the questions raised in (Christiansen, 2017)'s work.



Should We Exclude Transition Times Over 15 Hours?

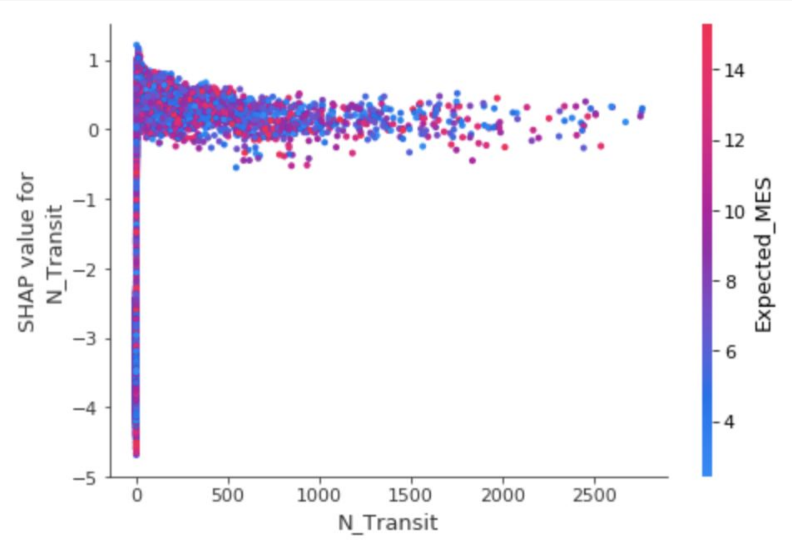
We evaluate whether or not one data instance with transit duration over 15 hours should be omitted

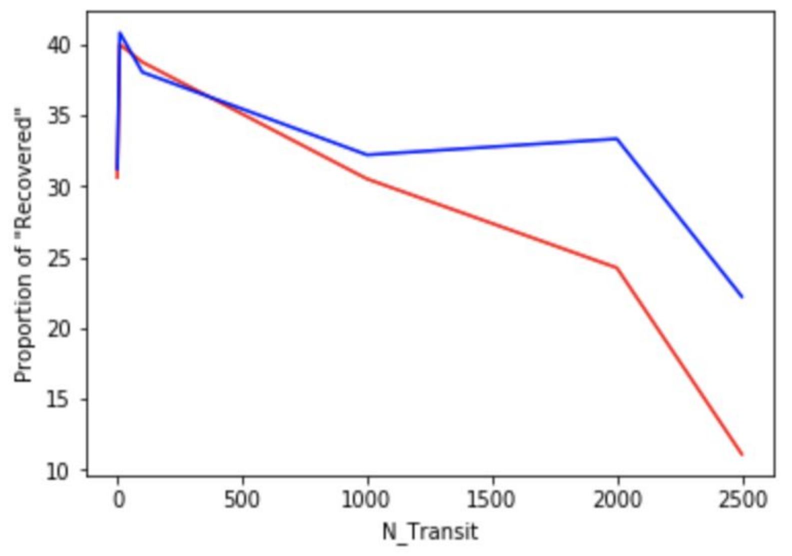




Threshold for the number of valid transits

Should we have the cut-off point for the number of valid transits?

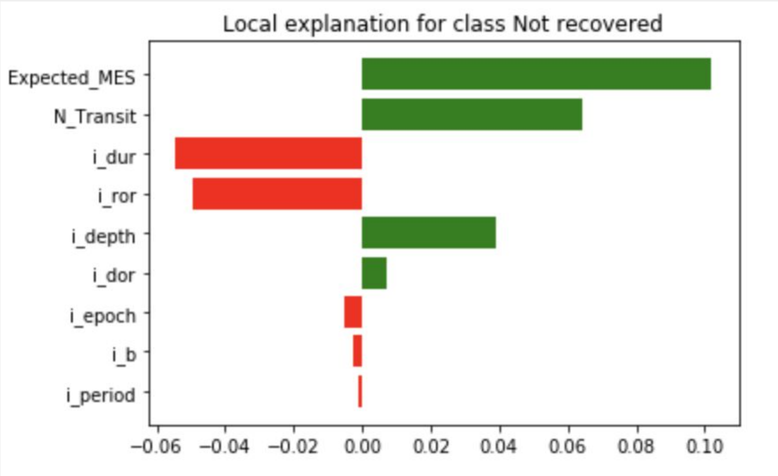




Additional Interpretation Methods

LIME Analysis

Explains the model by learning an interpretable model locally around the prediction.



Anchors (Influencer Scores)

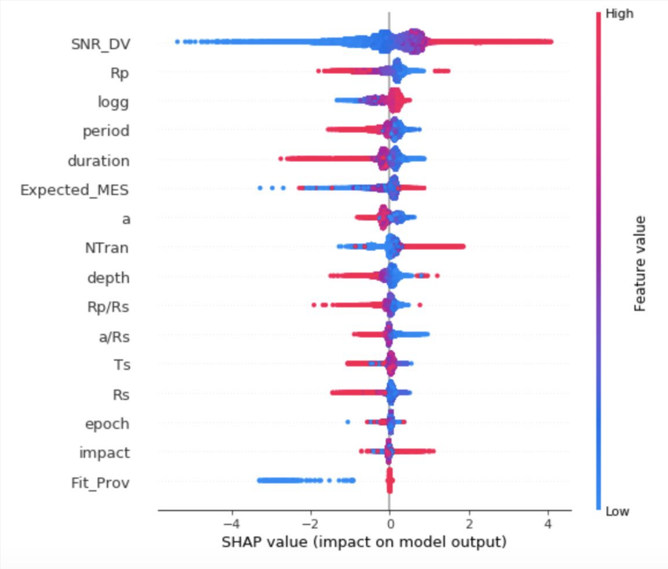
This explains which features are enough for the model to come up with the outcome in each instances of training data.

```
Partial anchor: N_Transit <= 3.04 AND i_ror > 0.02
Partial precision: 0.91
```

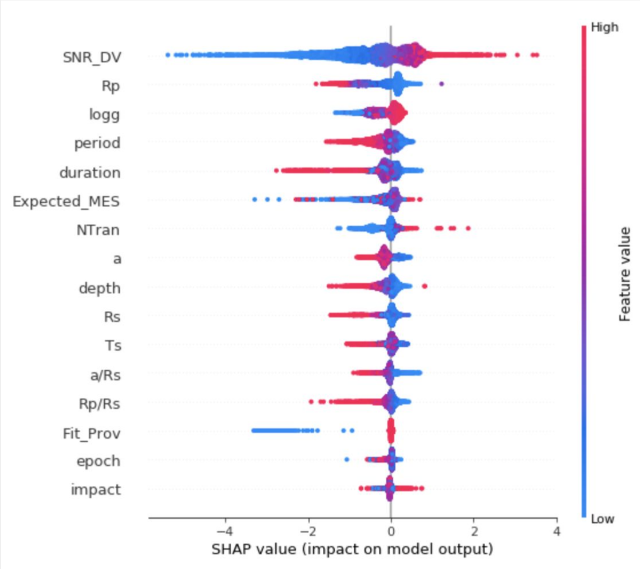
TCES Robovetter SHAP Analysis

We use SHAP to analyze which features correspond to specific FP types which are detected.

Planet Candidate vs All False Positives



Classification of False Positives



NOTE: Our priors used are based on Ground Truth and NOT from a hierarchical model.

Our Contributions

1. Efficient Data-Driven Data Processing Pipeline
2. PLTI Injection Predictive Model
3. TCES Predictive Model
4. Various Model Interpretation + Analysis

Thank You

Questions, Comments, or Suggestions?