GAIA SPECTRAL TYPE

Classification Report

Cyrus Kwan
ID: 25466929

Table of Contents

Data Mining Problem

Business objectives:

The spectral type tells the client a lot of information about the star including temperature, colour, luminosity, etc. Understanding these properties is necessary for aerospace planning and astronomical modelling research.

The scope of this project is limited to the Gaia system. Based on the dataset it considers entries that were present in the recording directly prior to the given dataset as it measures the motion of stars from one point in time to another.

Assess the Situation:

The task is to build a classification model to predict the spectral class of celestial bodies in the Gaia system for the client.

This project will use Python as the main tool for data exploration, preprocessing, model training, and predictions. Only a single researcher will be assigned to each classification model.

There is potential for a single researcher to build an inaccurate model hence, multiple researchers have been assigned to this task with each producing their own classification model.

Data Mining Goals:

The accuracy of each produced model will be tested against the true labels via Kaggle hosted competition. The methodology is outlined in this report and will be evaluated by senior researchers.

Project plan:

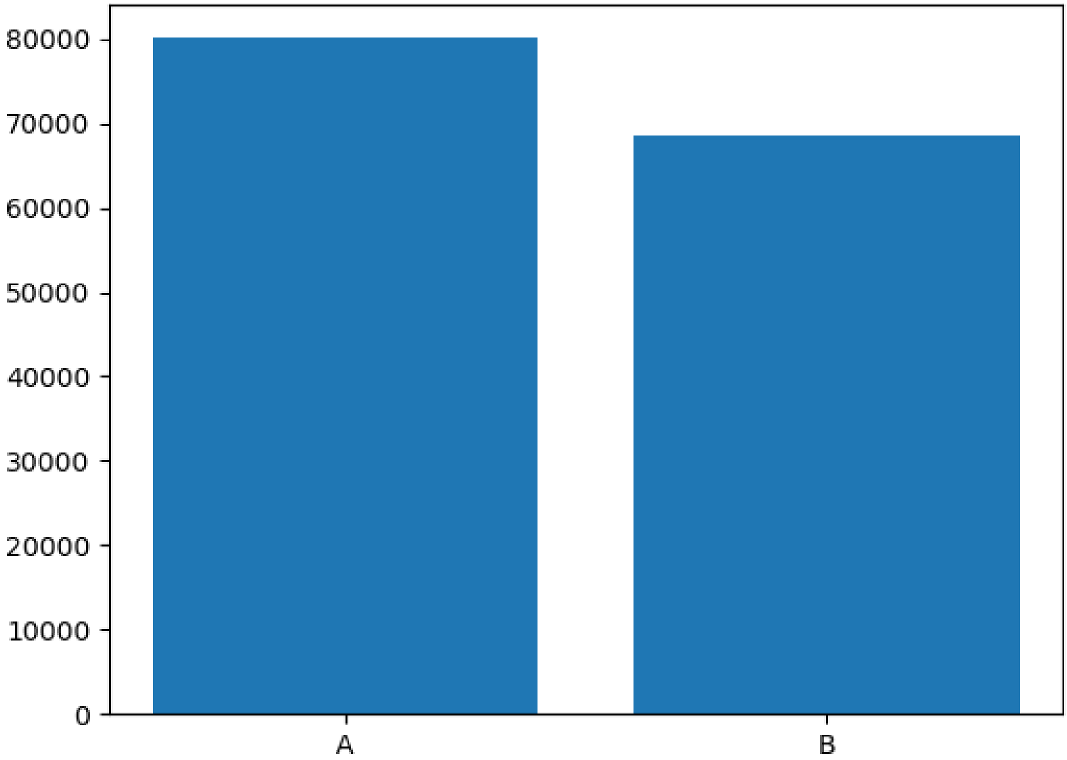
1. Data exploration
2. Data preprocessing
3. Feature selection
4. Model selection
5. Model training
6. Ensembling

Data Exploration

SpType-ELS

SpType-ELS is the target value. There are more A values than B values about a 60-40 split. In this study, A and B labels were recoded as numerical/binary representations for easier model training (A and B were coded as 0 and 1 respectively).

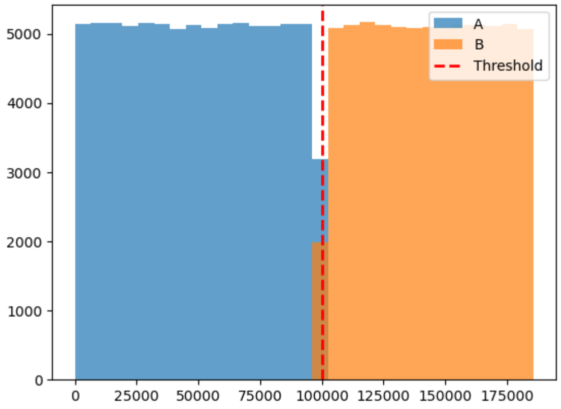
Distribution of A and B values in SpType-ELS



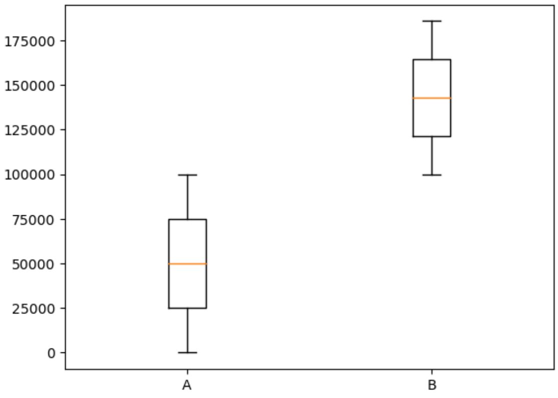
ID

ID seems to be a very clear indicator of types. First impression appears as though the attribute is linearly.

AB Histogram of ID



AB Boxplot of ID

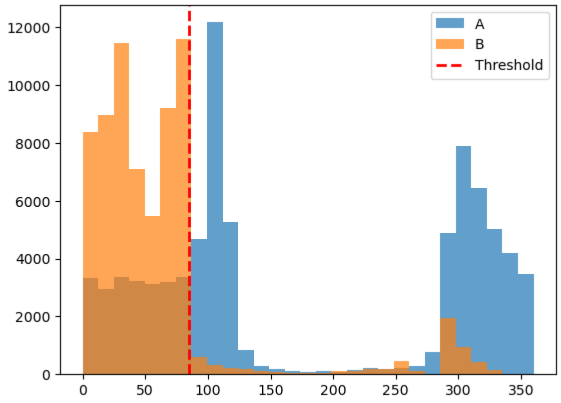


RA_ICRS

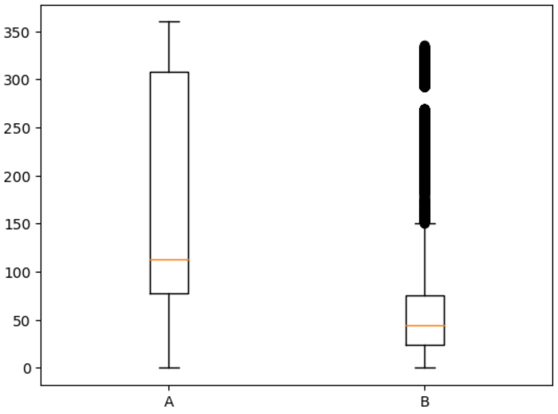
It appears B type is far more right skewed. For both types, there aren't many entries between 150 and 300. Label B has positive outliers from 150 to 350. You can somewhat draw a line at the point of 75 where most values on the left side will be B and on the right A.

However, entries on the left side of the line still have a considerable chance to be A values

AB Histogram of RA_ICRS



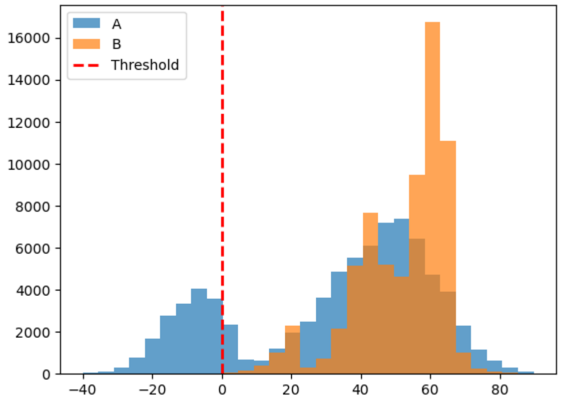
AB Boxplot of RA_ICRS



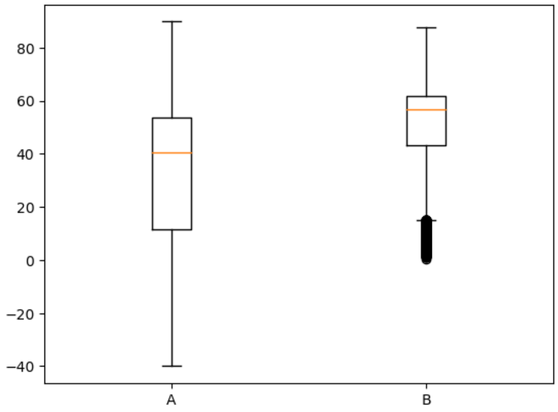
DE_ICRS

Type B has a narrower range compared to A. It would be difficult to distinguish A and B labels using this feature alone as their distributions overlap. There could be a case for entries less than or equal to zero being more likely to be labelled as A.

AB Histogram of DE_ICRS



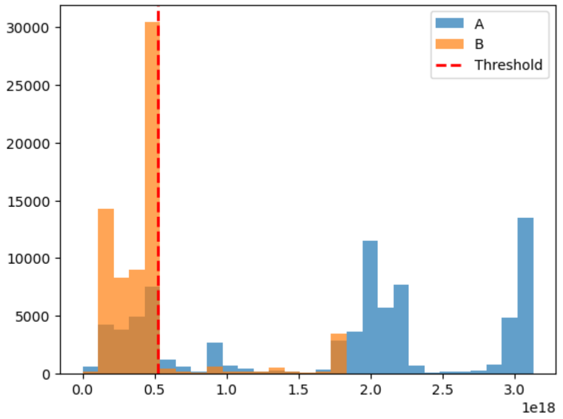
AB Boxplot of DE_ICRS



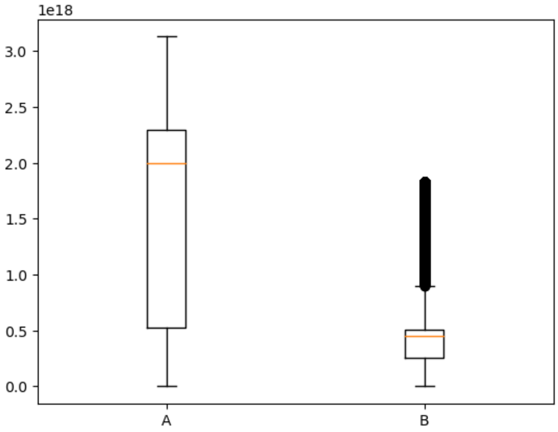
Source

A has a greater range compared to B. Appears separable with a soft margin/

AB Histogram of Source



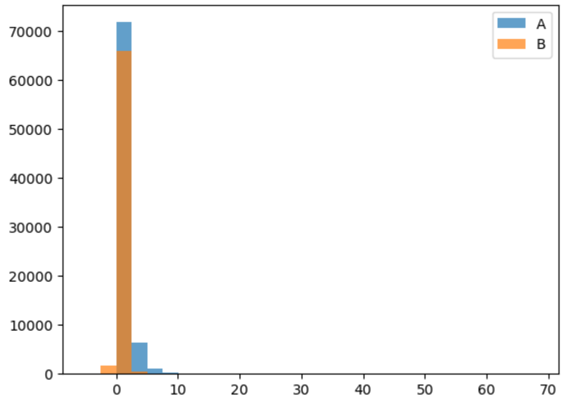
AB Boxplot of Source



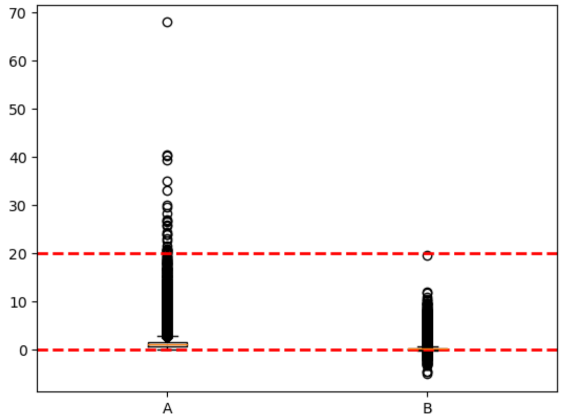
Plx

Most values appear around 0 to 10, it looks like for both types there are quite a few outliers. Most values occur around 0 for both A and B. Boxplot shows that <0 is almost always B and >20 is almost always A

AB Histogram of Plx



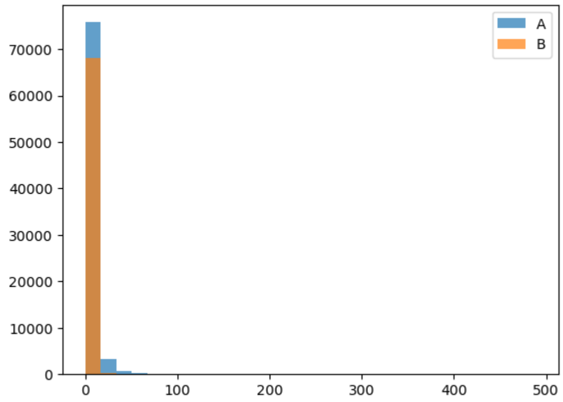
AB Boxplot of Plx



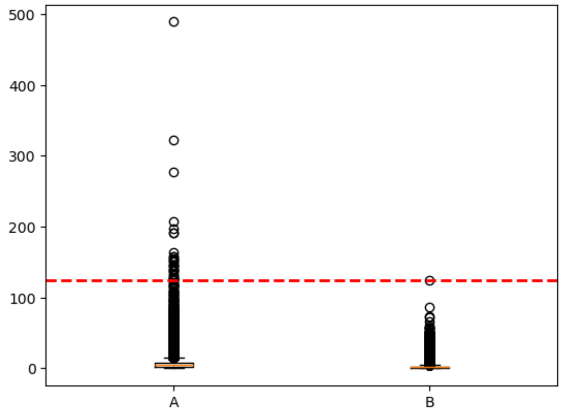
PM

Most values occur at around 0. There are quite a few outliers for both A and B. From the outliers indicated in the boxplots it seems that A values are more likely to have a PM of >125.

AB Histogram of PM



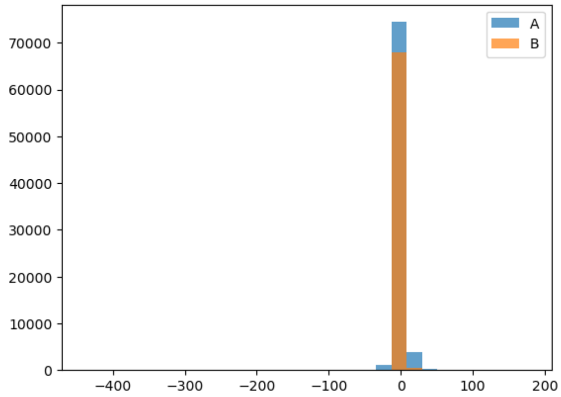
AB Boxplot of PM



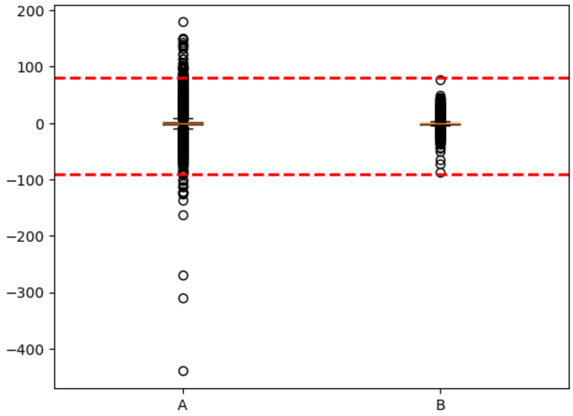
pmRA

B values don't appear to have moved left or right that much compared to A. For most values, neither of them has shifted a large distance. Hence, there are quite a few outliers for values that have moved a greater distance. Even for outliers, B has a limited range. The outliers falling outside this range seem to be only A values.

AB Histogram of pmRA



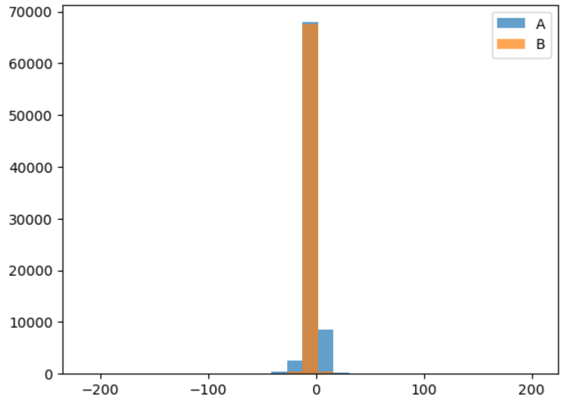
AB Boxplot of pmRA



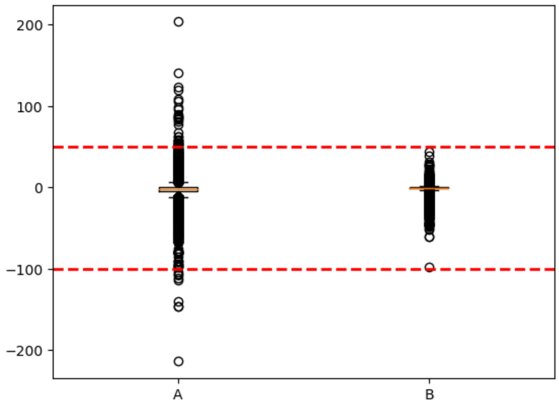
pmDE

A values appear to have moved up and down more compared to B values. For both values, neither appear to have moved a great distance. Hence, both values contain outliers for points that have moved a greater distance, though these outliers are too few to be indicative of any trend.

AB Histogram of pmDE



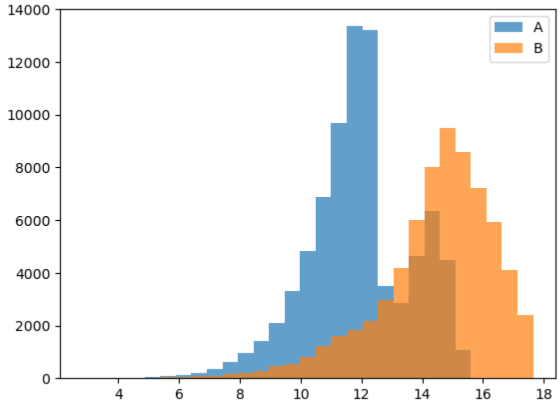
AB Boxplot of pmDE



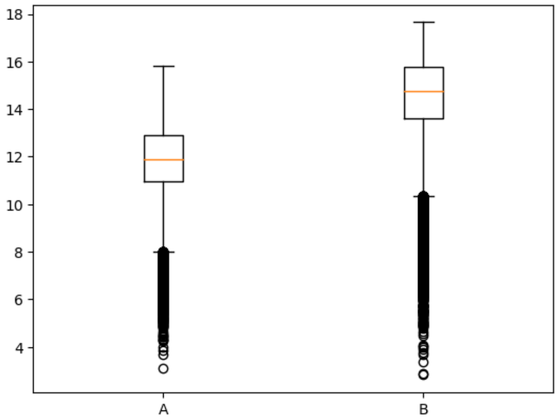
Gmag

B values are left skewed more than A values. B values are distributed further to the right. Appears separable with a soft margin as there are two distinct distributions for A and B values.

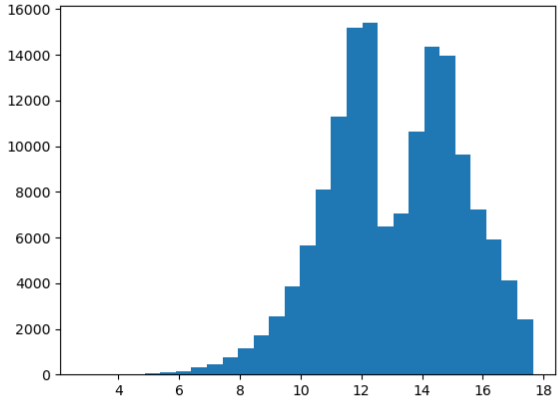
AB Histogram of Gmag



AB Boxplot of Gmag



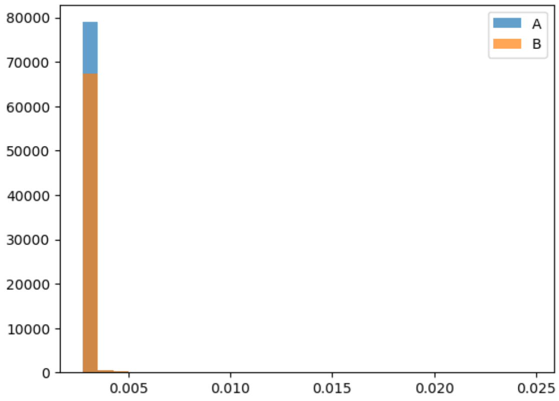
Histogram of Gmag



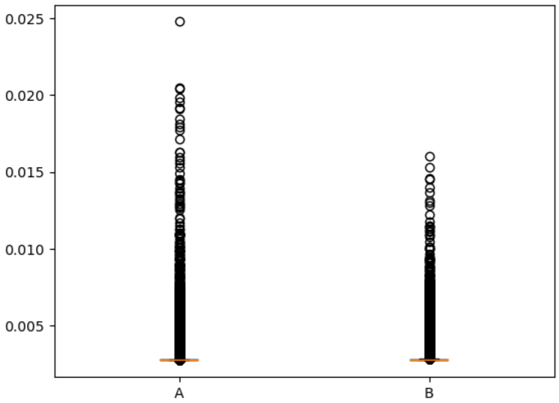
e_Gmag

Most error values occur between 0 and 0.005. Would assume that the error is correlated with the distance, but it doesn't seem like it given the correlation matrix. A has a slightly larger error range, though this is only because of a few outliers and likely isn't indicative of much.

AB Histogram of e_Gmag



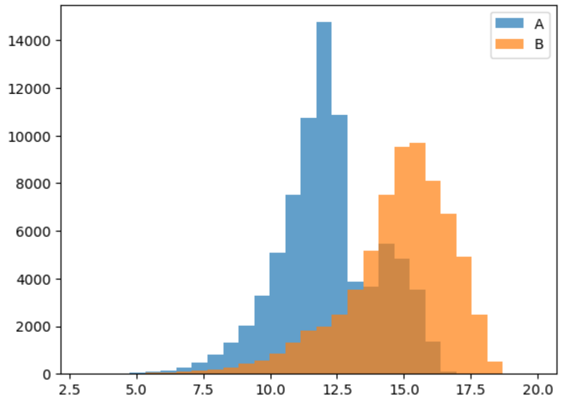
AB Boxplot of e_Gmag



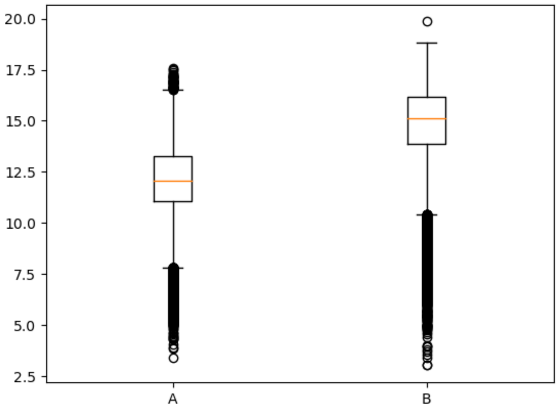
BPmag

Similar to Gmag, B type is more left skewed compared to A values. B values are distributed further to the right. Like Gmag, it appears somewhat separable as there are two distinct distributions of A and B values.

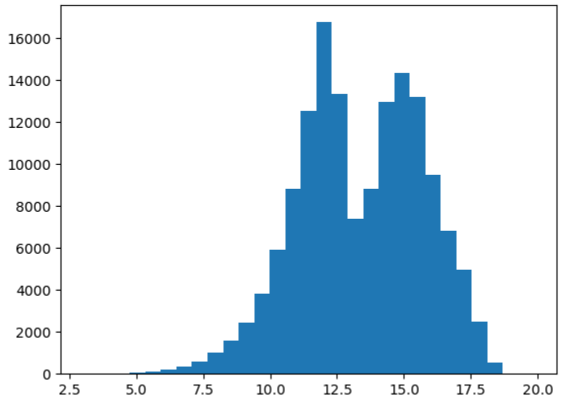
AB Histogram of BPmag



AB Boxplot of BPmag



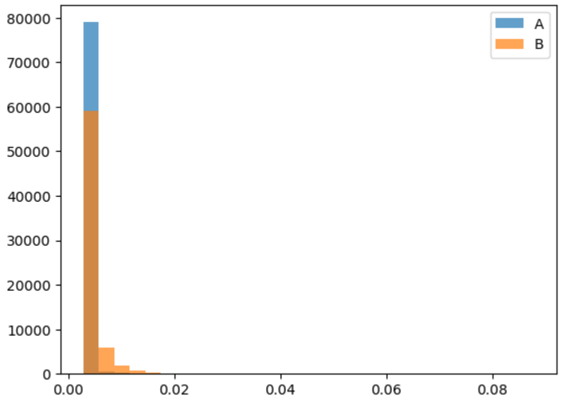
Histogram of BPmag



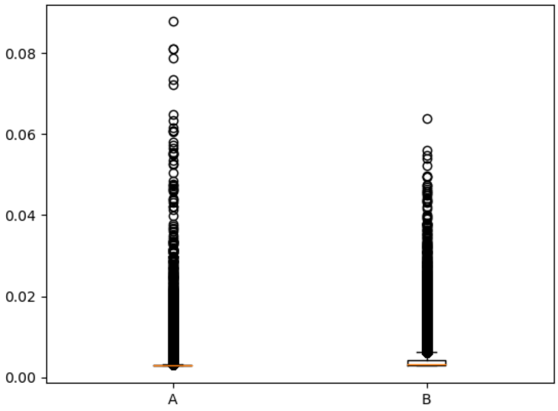
e_BPmag

Has a greater error range compared to Gmag 0 to 0.02. Does not appear particularly indicative of A and B labels.

AB Histogram of e_BPmag



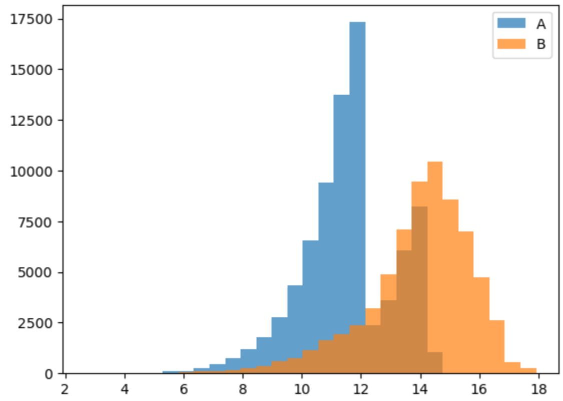
AB Boxplot of e_BPmag



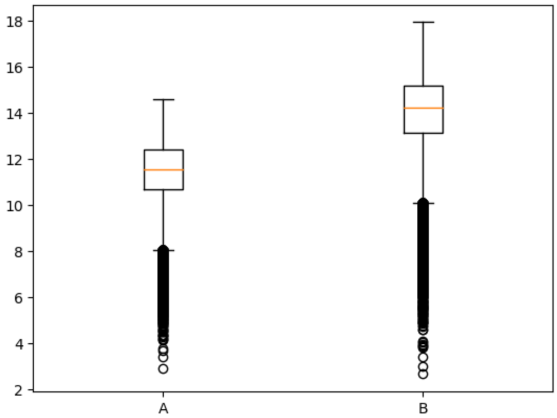
RPmag

Following the same trend, B values are more left skewed compared to A values. B values are distributed further to the right. Same with previous mag scores there appears to be two distinct distributions.

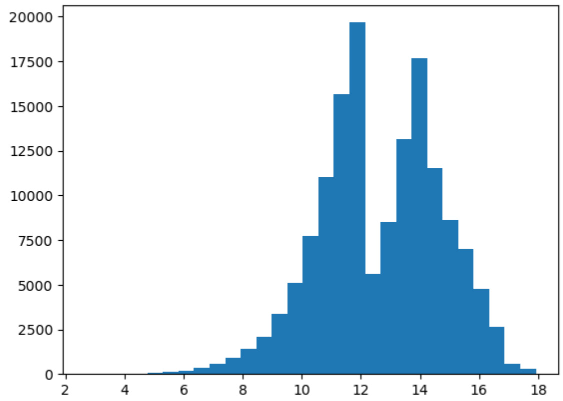
AB Histogram of RPmag



AB Boxplot of RPmag



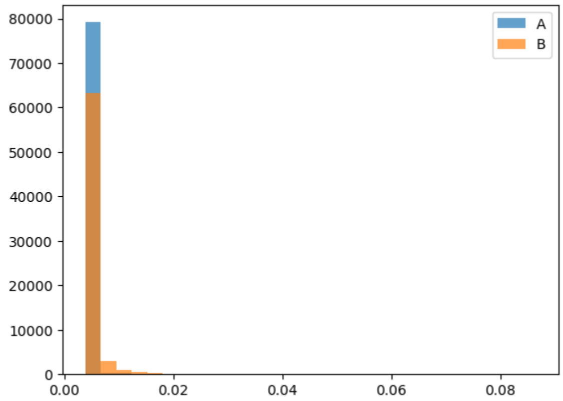
Histogram of RPmag



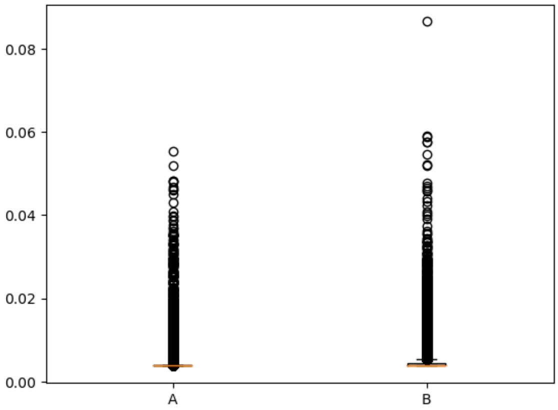
e_RPmag

Compared to previous errors it appears that B labels have a wider range compared to A labels. Because of the severe overlap between A and B distributions, this feature is not indicative of A and B labels.

AB Histogram of e_RPmag



AB Boxplot of e_RPmag

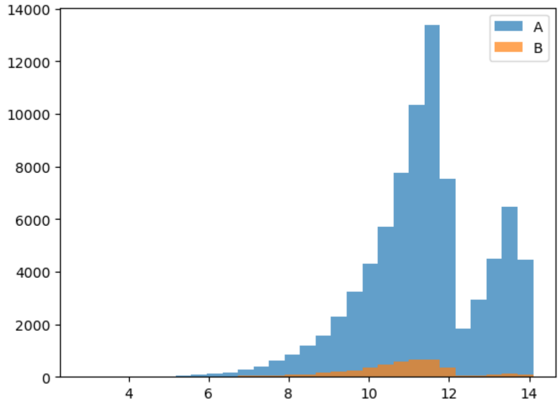


GRVSmag

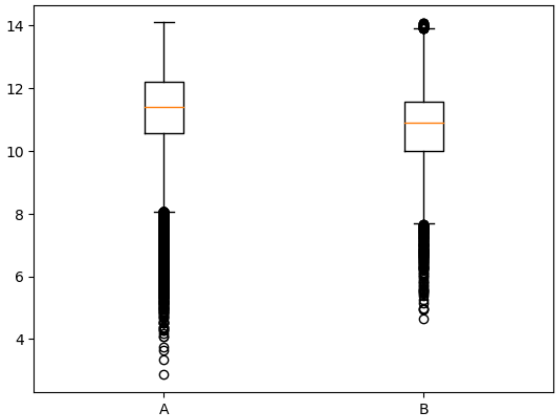
Closer inspection shows that most B values are null. In comparison, A label has no null values.

```
# A
count     80088
unique        1
top       False
freq      80088
Name: GRVSmag, dtype: object
# B
count     68450
unique        2
top        True
freq      64054
Name: GRVSmag, dtype: object
```

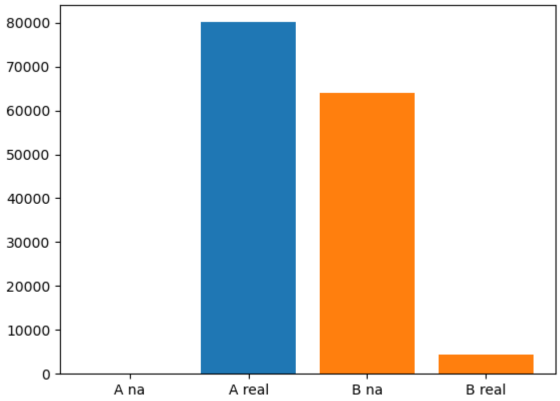
AB Histogram of GRVSmag



AB Boxplot of GRVSmag



Distribution of null values in GRVSmag

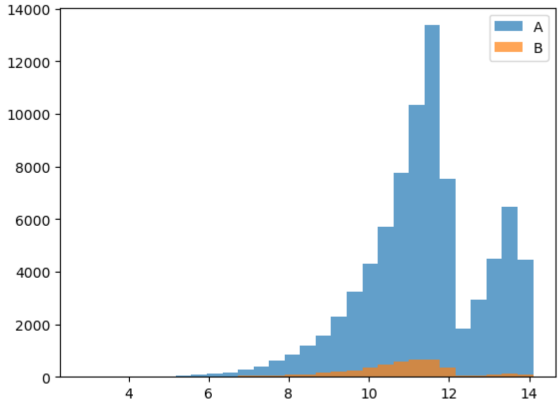


e_GRVSmag

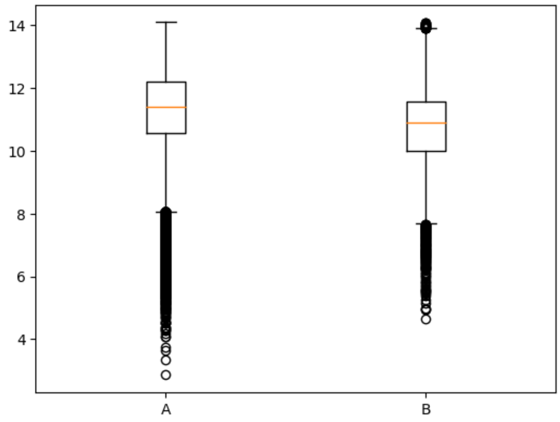
Closer inspection shows that most B values are null. In comparison, A label has no null values.

```
# A
count      80088
unique         1
top        False
freq       80088
Name: e_GRVSmag, dtype: object
# B
count      68450
unique         2
top         True
freq       64054
Name: e_GRVSmag, dtype: object
```

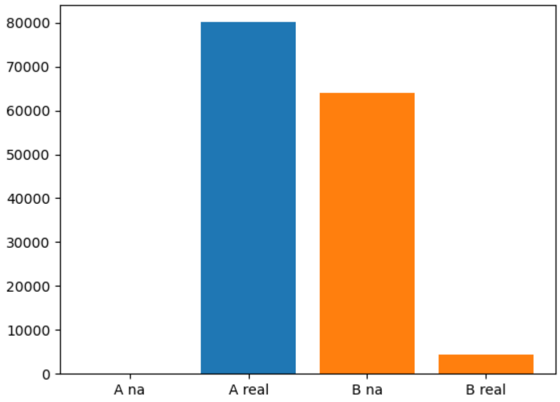
AB Histogram of e_GRVSmag



AB Boxplot of e_GRVSmag



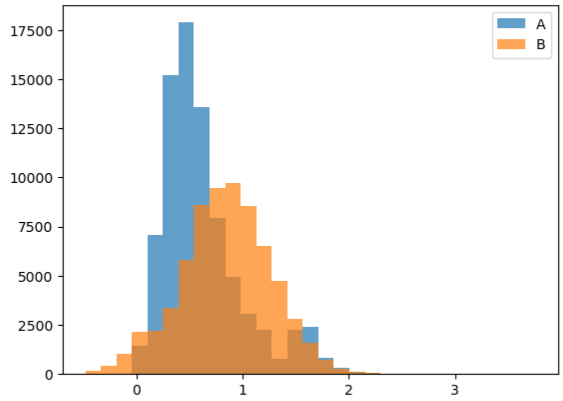
Distribution of null values in e_GRVSmag



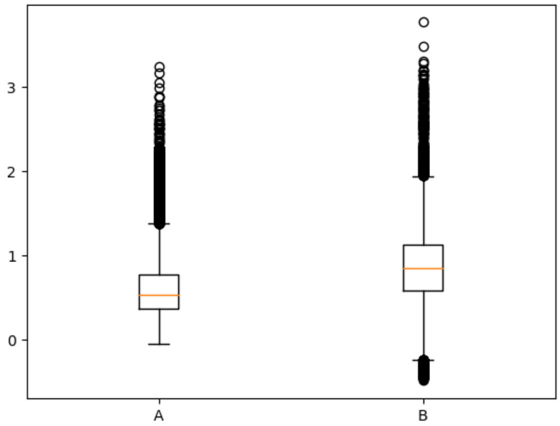
BP-RP

B is normally distributed between 0 and 2. A is right skewed. Difficult to separate as the distributions are mostly overlapping.

AB Histogram of BP-RP



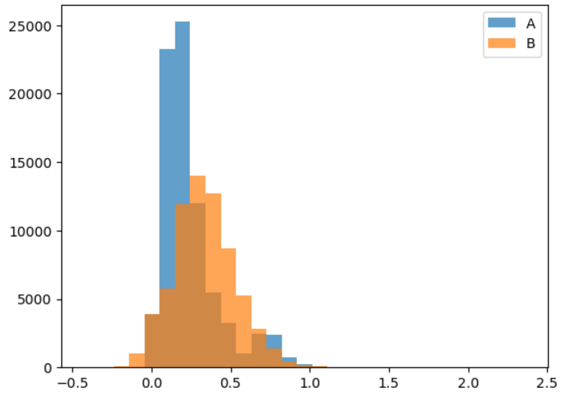
AB Boxplot of BP-RP



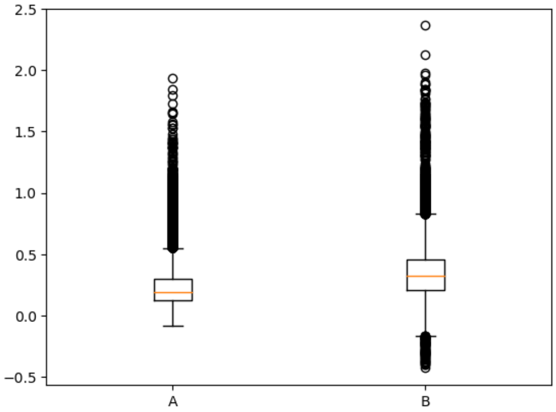
BP-G

B values are more evenly distributed compared to A values. Most A values occur around 0 to 0.25. Difficult to separate as the distributions are mostly overlapping.

AB Histogram of BP-G



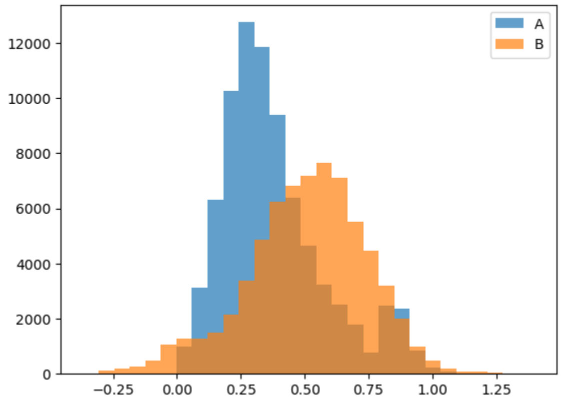
AB Boxplot of BP-G



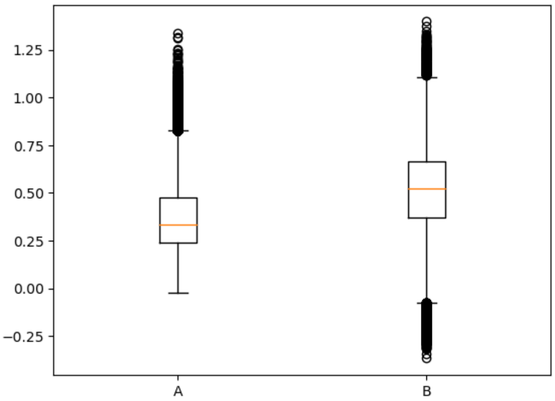
G-RP

B values are distributed further to the right, more evenly distributed, and are left skewed compared to the right skewed A values. The distribution of the two labels shows they are mostly overlapping and likely the whole feature is not indicative of any separability.

AB Histogram of G-RP



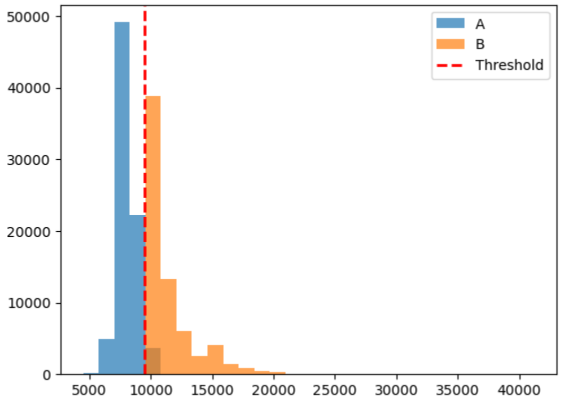
AB Boxplot of G-RP



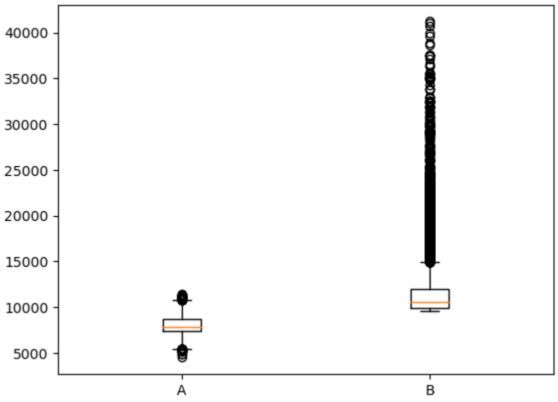
Teff

B values have a far larger distribution range compared to A values. A values occur between 5000 and 10000 while B values occur between 10000 and 40000. Appears to be quite a good differentiator between A and B values.

AB Histogram of Teff



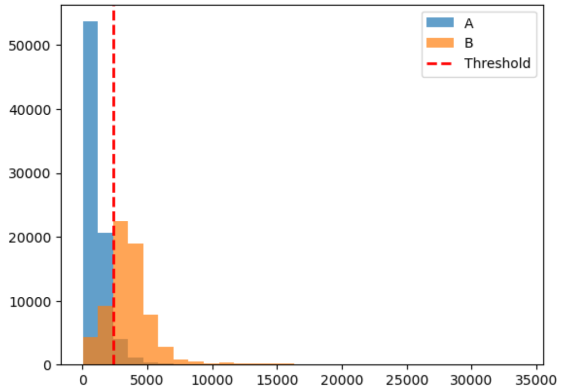
AB Boxplot of Teff



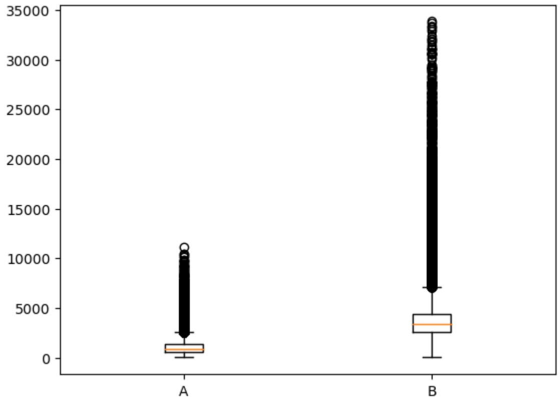
Dist

B values are more evenly distributed. B values occur in a wider range. A (0 to 5000) B (0 to 20000). From this we can determine that Dist is somewhat separable according to A and B labels.

AB Histogram of Dist



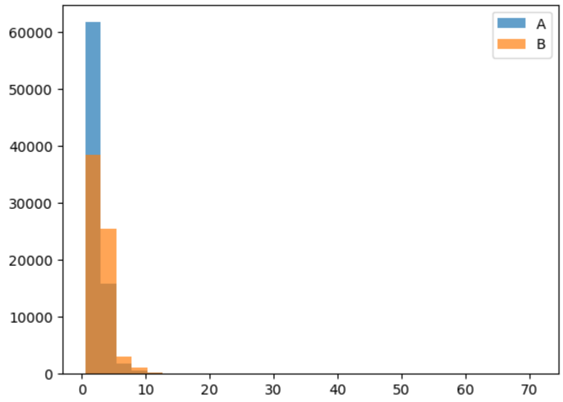
AB Boxplot of Dist



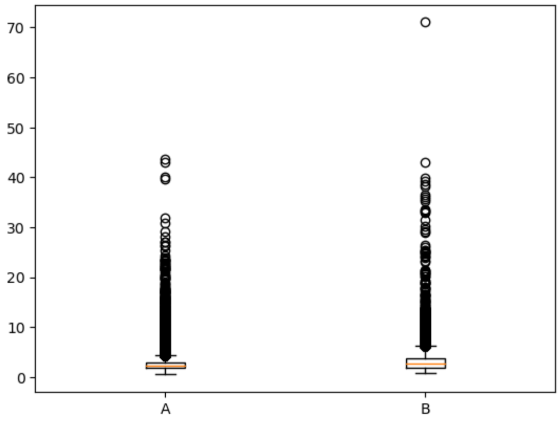
Rad

B values have a larger distribution A (0 to 10) B (0 to 15). Both features overlap 0 to 10 and it seems that there is too much overlap between the two features to be indicative of A and B labels.

AB Histogram of Rad



AB Boxplot of Rad

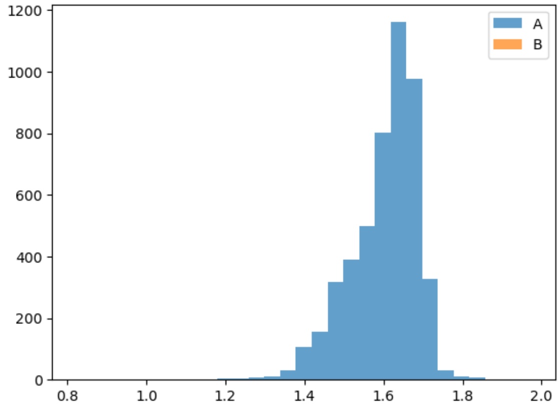


pscol

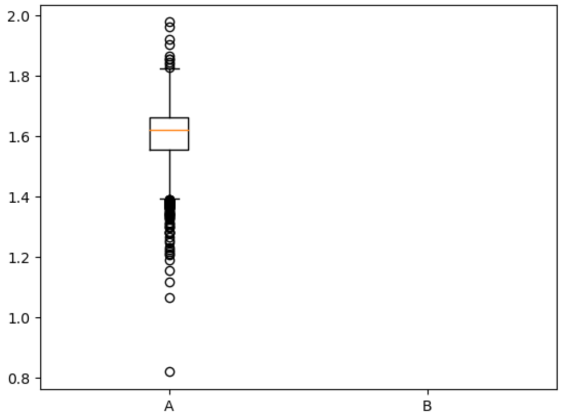
Closer inspection shows that all B values are null. Although both values mostly contain null values, it can be surmised that any entry that contains a non-null value will always be an A label.

```
# A
count     80088
unique        2
top        True
freq      75235
Name: pscol, dtype: object
# B
count     68450
unique        1
top        True
freq      68450
Name: pscol, dtype: object
```

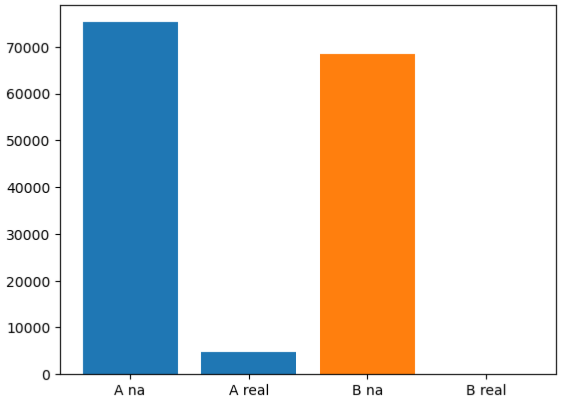
AB Histogram of pscol



AB Boxplot of pscol



Distribution of null values in pscol

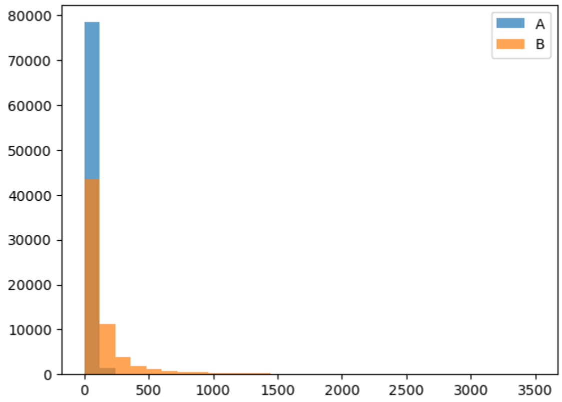


Lum-Flame

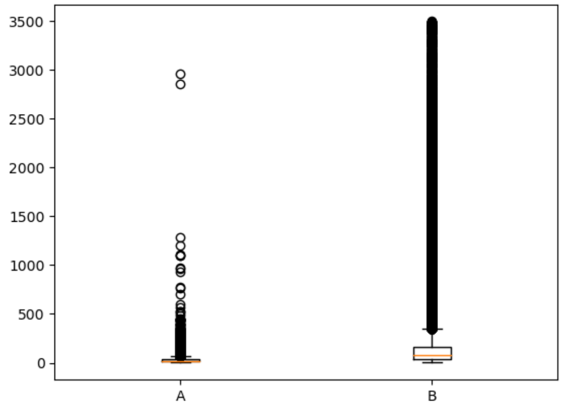
B has a wider range of values. About 3000 B values are null. Although the overlap of distributions may not provide too much insight, it can be determined that an value containing a null value is of B type.

```
# A
count     80088
unique        1
top       False
freq      80088
Name: Lum-Flame, dtype: object
# B
count     68450
unique        2
top       False
freq      65455
Name: Lum-Flame, dtype: object
```

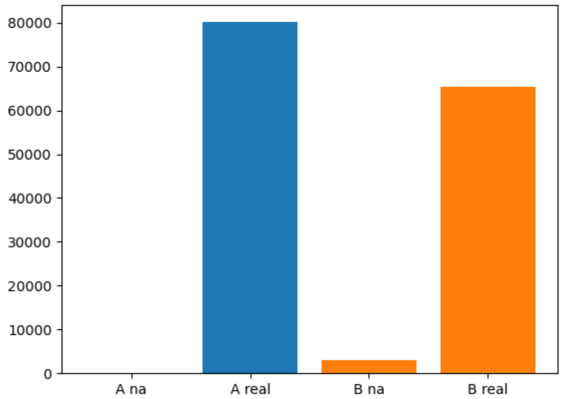
AB Histogram of Lum-Flame



AB Boxplot of Lum-Flame



Distribution of null values in Lum-Flame

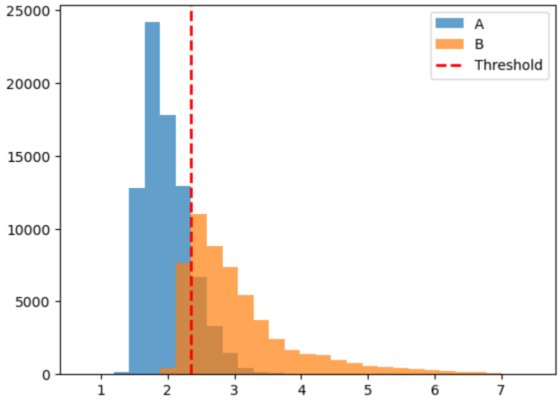


Mass-Flame

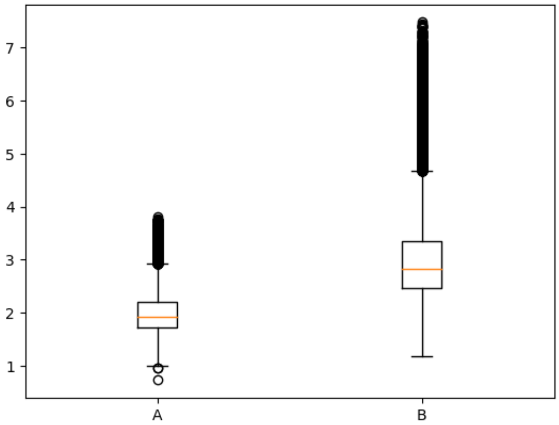
A values are complete compared to13000 missing B values. B has a wider distribution and are heavily right skewed while A values are distributed further to the left. The whole feature appears somewhat separable.

```
# A
count     80088
unique        1
top       False
freq      80088
Name: Mass-Flame, dtype: object
# B
count     68450
unique        2
top       False
freq      55899
Name: Mass-Flame, dtype: object
```

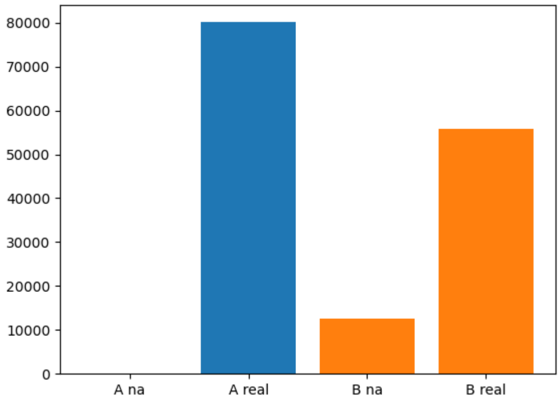
AB Histogram of Mass-Flame



AB Boxplot of Mass-Flame



Distribution of B null values in Mass-Flame

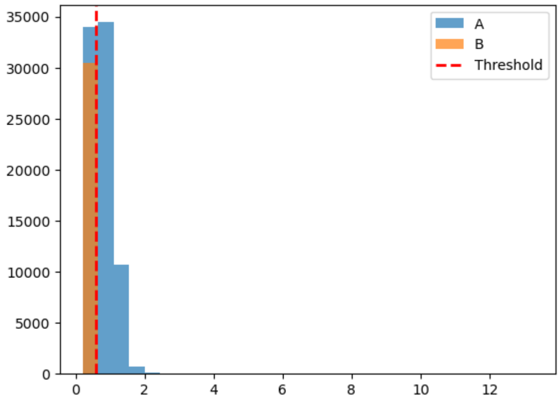


Age-Flame

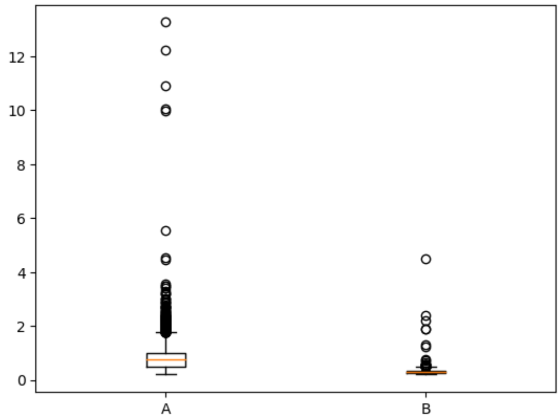
Most B values are missing. There is a wider distribution of A values than there are B values. Too much overlap to be indicative however, soft margin could be drawn to separate main distributions.

```
#A
count      80088
unique         1
top        False
freq       80088
Name: Age-Flame, dtype: object
# B
count      68450
unique         2
top         True
freq       38016
Name: Age-Flame, dtype: object
```

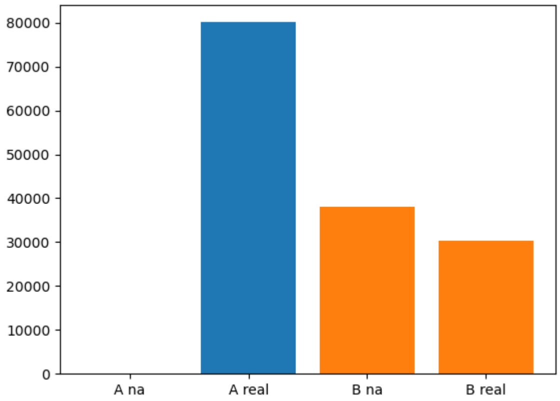
AB Histogram of Age-Flame



AB Boxplot of Age-Flame



Distribution of B null values in Age-Flame

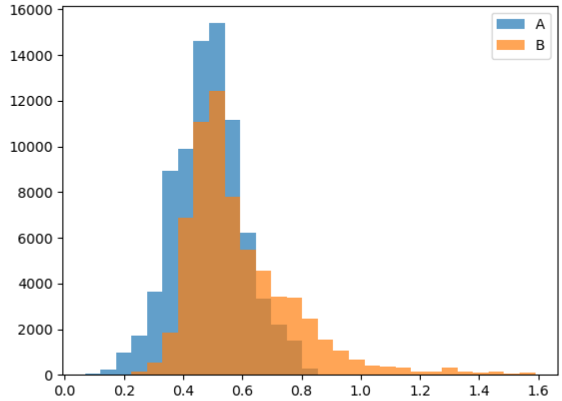


z-Flame

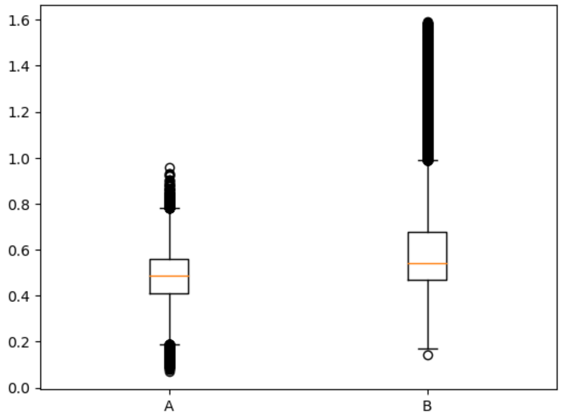
Both A and B values appear normally distributed. Only a few missing B values. Too much overlap to be indicative.

```
# A
count      80088
unique         1
top        False
freq       80088
Name: z-Flame, dtype: object
# B
count      68450
unique         2
top        False
freq       65455
Name: z-Flame, dtype: object
```

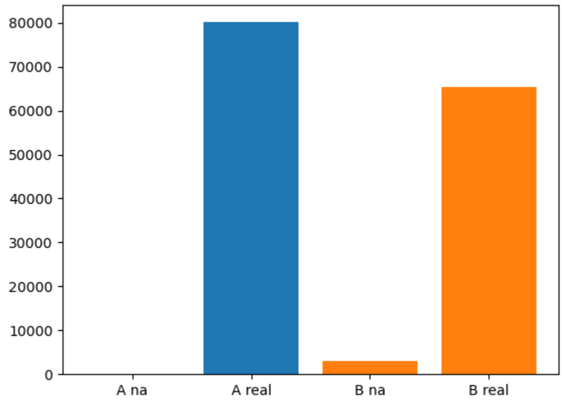
AB Histogram of z-Flame



AB Boxplot of z-Flame



Distribution of B null values in z-Flame

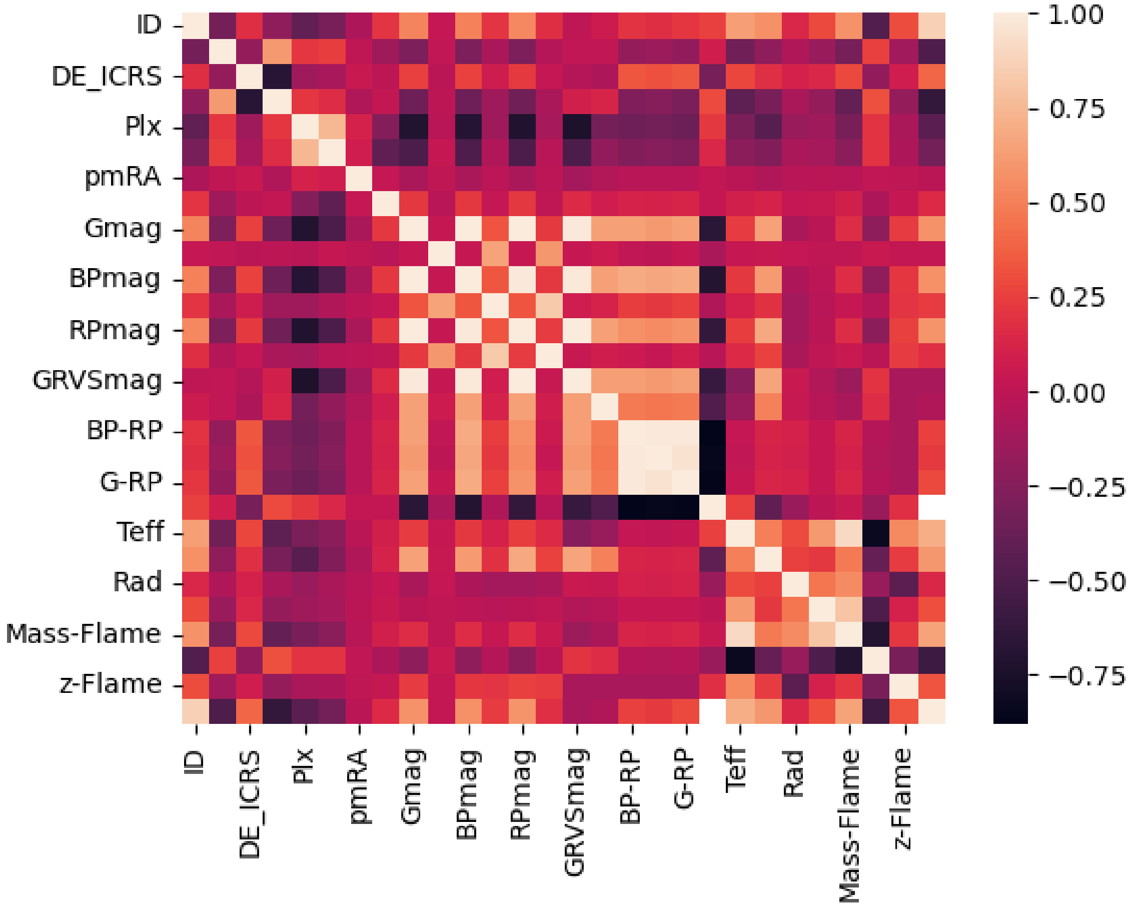


Pearson Correlation

Most indicative features appear to be ID, Teff, Mass-Flame, Source, and Dist.

```
ID             0.863172
Source        -0.625606
Gmag           0.579398
BPmag          0.566933
RPmag          0.587500
Teff           0.703806
Dist           0.599478
Mass-Flame     0.650493
Age-Flame     -0.577920
SpType-ELS     1.000000
Name: SpType-ELS, dtype: float64
```

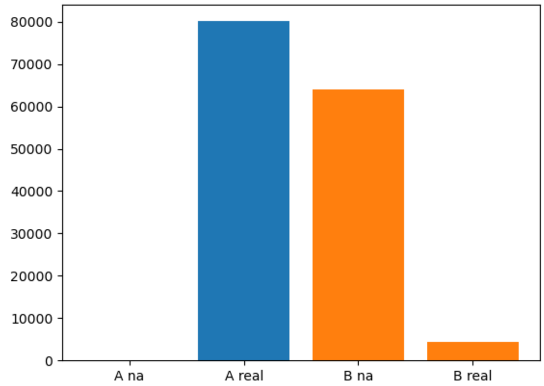
Pearson Correlation Matrix of all Features



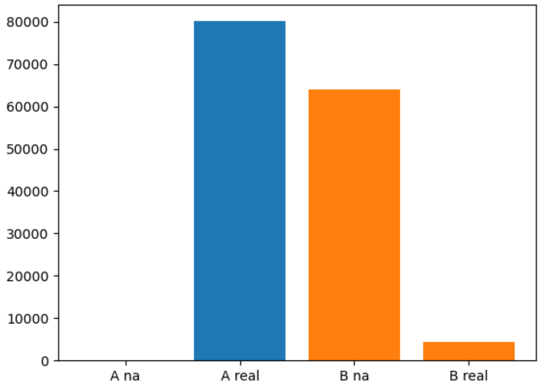
Data Preprocessing

Identified that for the features that contained null values, they are almost always indicative of a B label.

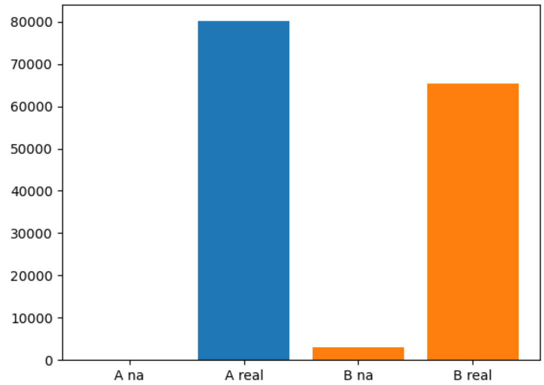
Null distribution of GRVSmag



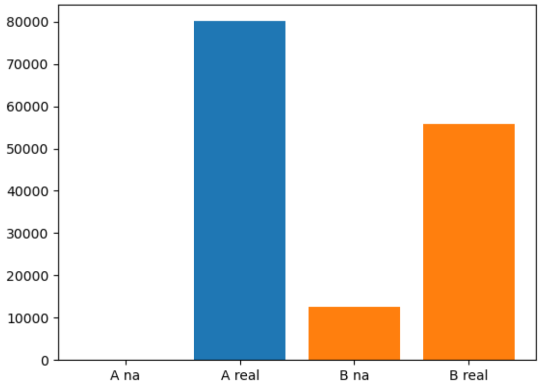
Null distribution of e_GRVSmag



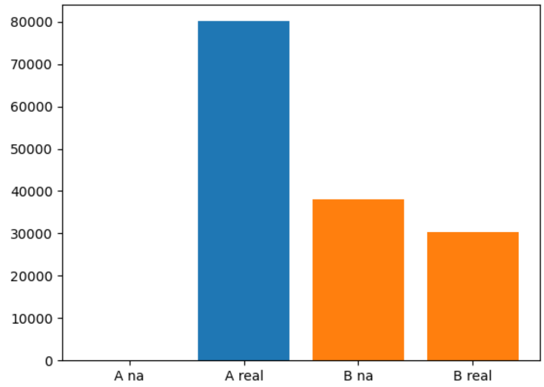
Null distribution of Lum-Flame



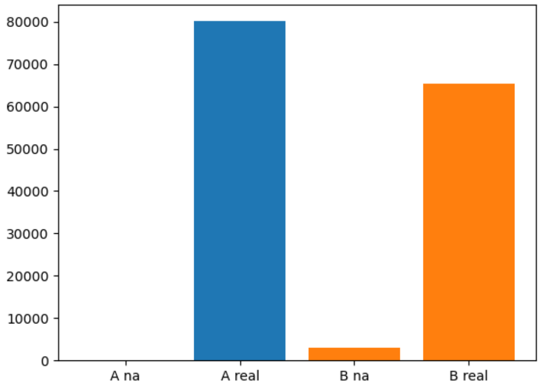
Null distribution of Mass-Flame



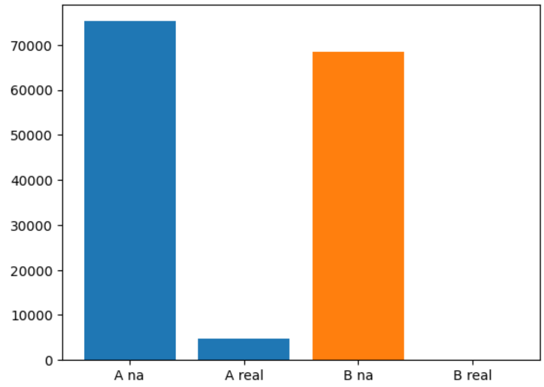
Null distribution of Age-Flame



Null distribution of z-Flame



Null distribution of pscol



To better represent this, new binary features were created where if a label was initially null, it was given a value of 1 and 0 for all other values.

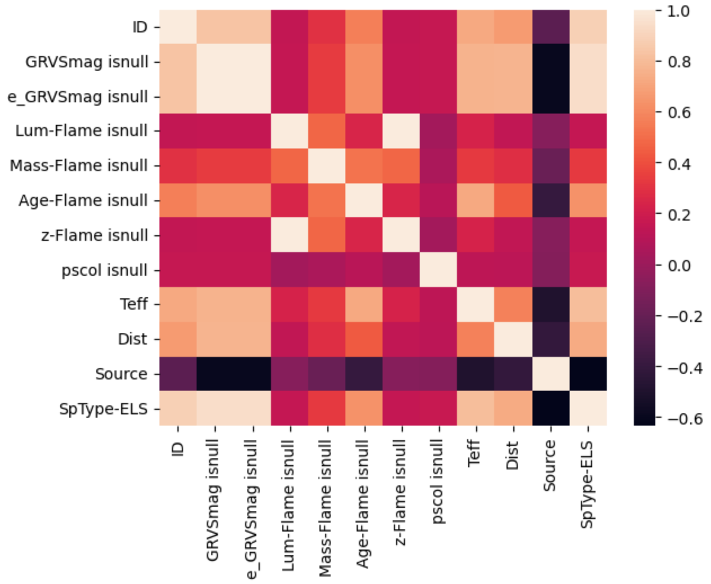
```
# These null features aren't indicative of A and B labels however, their missing values appear to only be B values
null_features = ["GRVSmag", "e_GRVSmag", "Lum-Flame", "Mass-Flame", "Age-Flame", "z-Flame", "pscol"]
label = lambda attr: 1 if pd.isnull(attr) else 0

# Hence, we create new features to show which features were originally null or not
for feature in null_features:
    data[f"{feature} isnull"] = data[feature].isnull().astype(int)
```

Realized that some features having a value was dependant on other features. For example, the status of Age-Flame having a value is dependent on both Lum-Flame, and Mass-Flame having a value. Hence it is redundant to use all flame values. We remove these dependencies because we don't want multiple features telling the same information that effectively reduce the weight of other features that tell different information. It can also be noted that both Lum-Flame and z-Flame share the same null entries. The remaining features used for model training were "GRVSmag isnull", and "Age-Flame isnull".

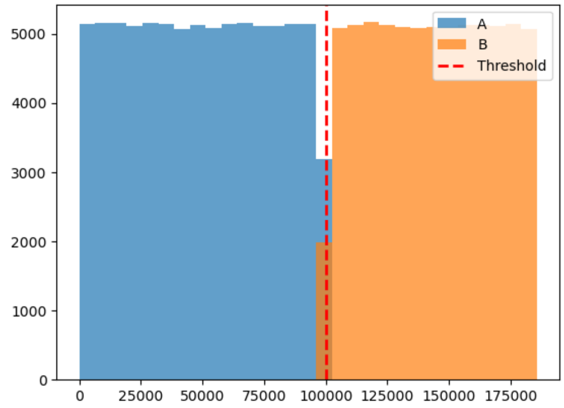
```
Entries where Lum-Flame is null and z-Flame is not null: 0
Entries where Lum-Flame is not null and z-Flame is null: 0
Entries where Lum-Flame is null and Mass-Flame is not null: 0
Entries where Lum-Flame is not null and Mass-Flame is null: 9556
Entries where Mass-Flame is null and Age-Flame is not null: 0
Entries where Mass-Flame is not null and Age-Flame is null: 25465
```

Matrix of Null and High Correlation Features

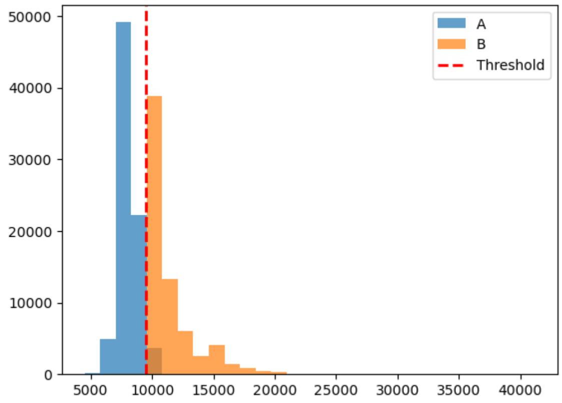


The other features that were selected for model training were those that from initial data exploration that appeared the most separable and had a higher correlation with the target SpType-ELS.

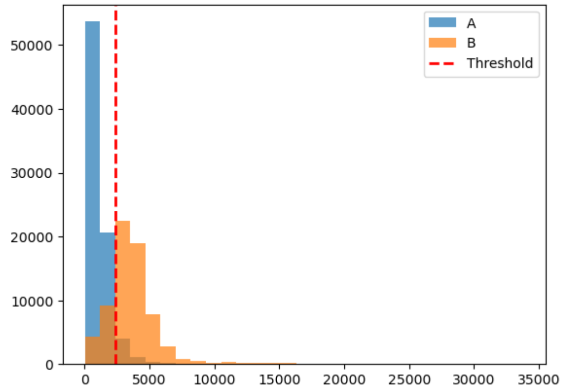
Histogram of ID



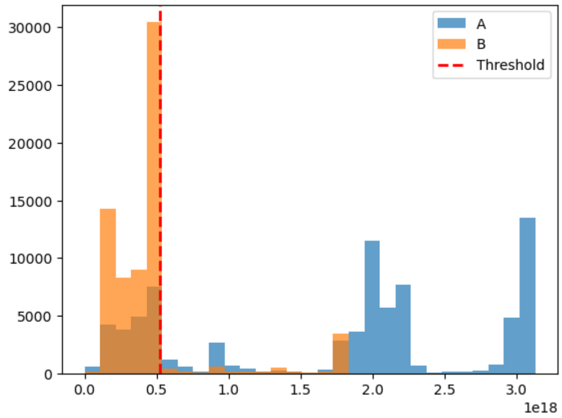
Histogram of Teff



Histogram of Dist

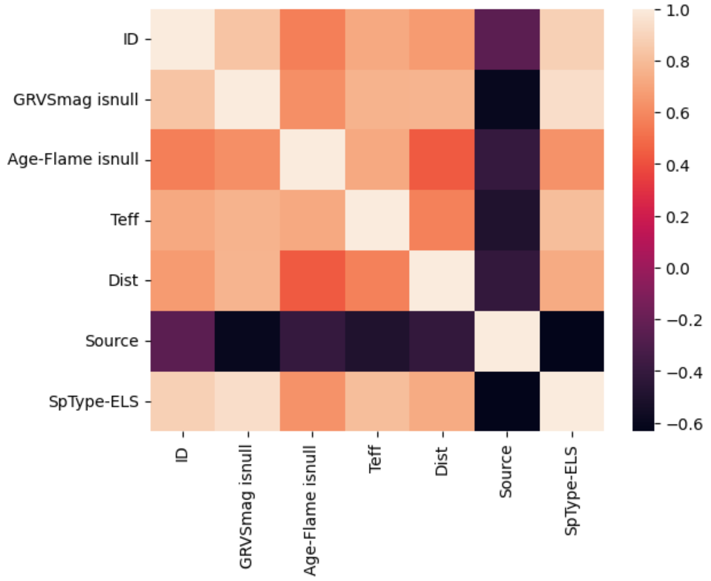


Histogram of Source



All features that weren’t already distributed between 0 and 1 were normalized using a sigmoid normalization. One of the models that was tested was K nearest neighbours where the Euclidian distance between points is used for classification. It is imperative that a single feature does not have more impact on the model than any other feature. Another reason sigmoid normalization was used is that since it is intended to receive new data points in the unlabelled test set, the model is likely to encounter values that appear out of range of the original training set.

Final Features for Model Training



Approach to the Problem

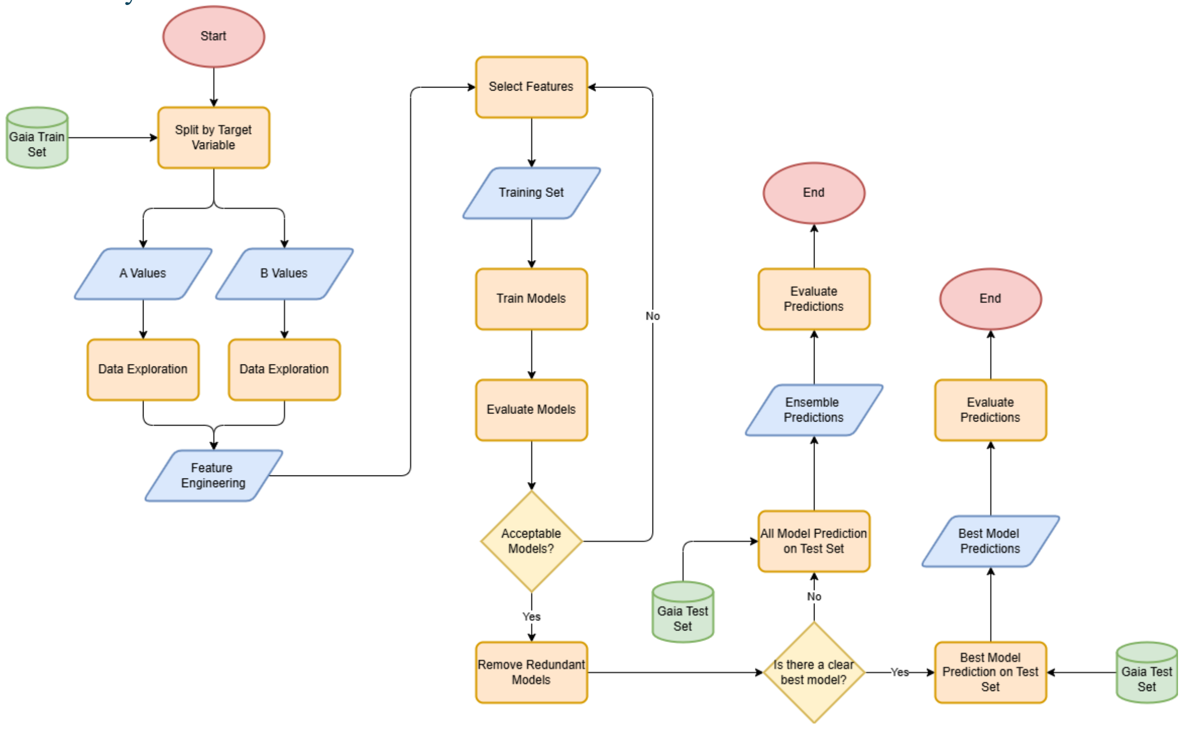
Data Exploration: Started with initial data exploration to see how A and B labels were distributed for each feature. No features were initially dismissed.
A and B values were initially split to view the differences between the labels to identify if there were any determining factors that influenced each. From this it was found that in almost all cases, entries that contained null values were the most likely to be labelled as B types.

Feature Engineering: New features were created to represent whether a certain entry was initially null or not.
Based on a Pearson correlation matrix and initial data exploration, the chosen features for model training were determined by their degree of correlation and perceived separability between features.
Feature selection ensured that the chosen features were not redundant by removing features that told the same information as another feature.

Model Training: Training was performed using a k-fold cross-validation as it is more robust that the standard training and testing split and is less likely to encounter bias from the training set.
The results of each fold for all models were evaluated using an overall accuracy score and ROC curve.
If the model did not perform well enough, parameters and feature selection was reevaluated, and the model was trained again.

Ensembling: Ensembling was used in the case that it was unclear which model to select for the final predictions.
Since there was no clear model that performed better than the others, the bagging ensembling method was used.
The final model predictions were evaluated against the true labels via a competition hosted on Kaggle.

Gaia Analysis Framework



Classification & Justification

Decision Tree

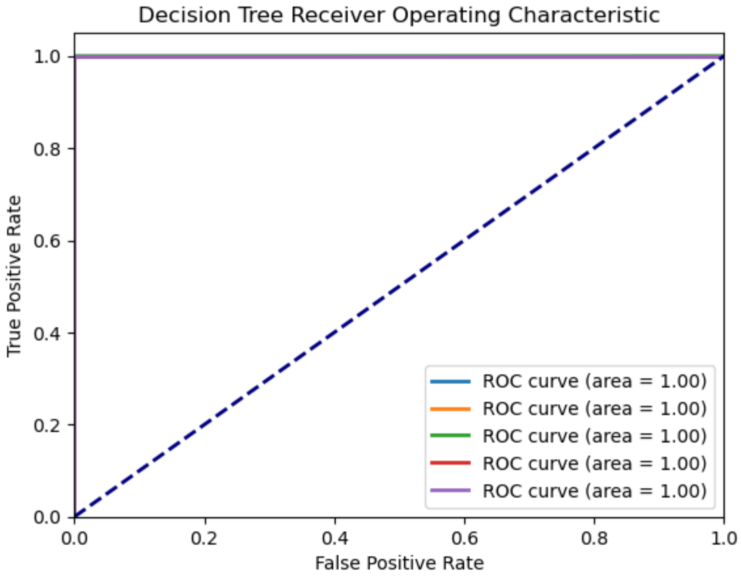
Justification: Based on the features that were chosen for model training, Entries should be linearly separable e.g. "GRVSmag isnull" is a binary feature. Other features that were chosen were done so based on their perceived separability and correlation to the target.

Pros: Requires few resources. Simple model is unlikely to overfit to the training data.

Cons: Not particularly robust as it is only a single decision tree. Prone to bias based on given training split.

Parameters: Max depth was chosen to continue until only pure leaf nodes remained. Reason being that for some features like "Teff" and "Dist" the exact amount of perfectly separable clusters is unknown. Hence, no max depth was selected for full coverage.
For the same reason, the minimum sample split was set to two such that each node would continue splitting until only pure leaf nodes remained.

Results: Using the decision tree model, it was able to predict the target B=1 label with a complete true positive rate and no false positives.



K-Nearest Neighbours

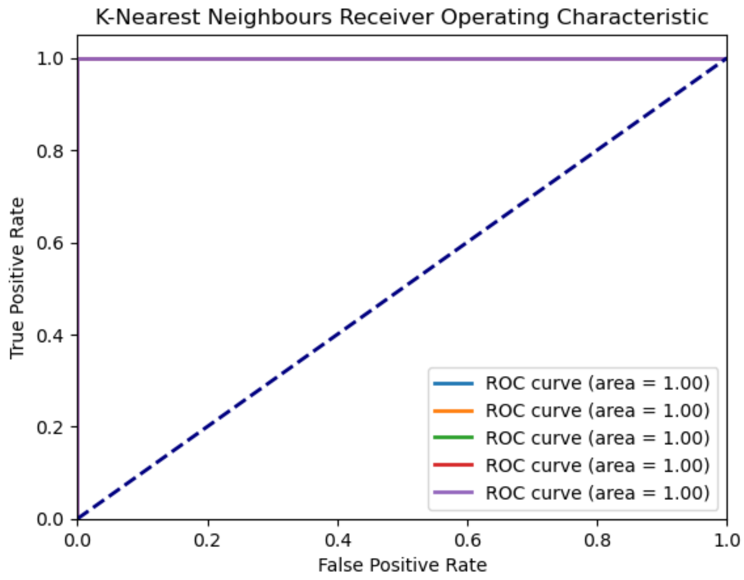
Justification: Four of the six features used are numerical with distributions that overlap to some extent. Although an effort has been made to choose features that are easily separable, these numerical features distinguish between target labels with more granularity compared to simple binary features.

Pros: Unlike other classification models, k-nearest neighbours does not require a training phase. This makes the model computationally efficient.

Cons: Equally sensitive to all features. It should be ensured that the chosen features are indicative of the target.

Parameters: N neighbours were chosen to be an odd number to prevent ties in classification. A value of five neighbours was chosen to maintain model simplicity.
The metric used was Euclidean distance to find neighbours across multiple dimensions as it is assumed that the straight-line distance in any direction is indicative of the target label.

Results: Given the target B=1 the kNN model achieved complete true positive rate and false positive rate.



Multilayer Perceptron

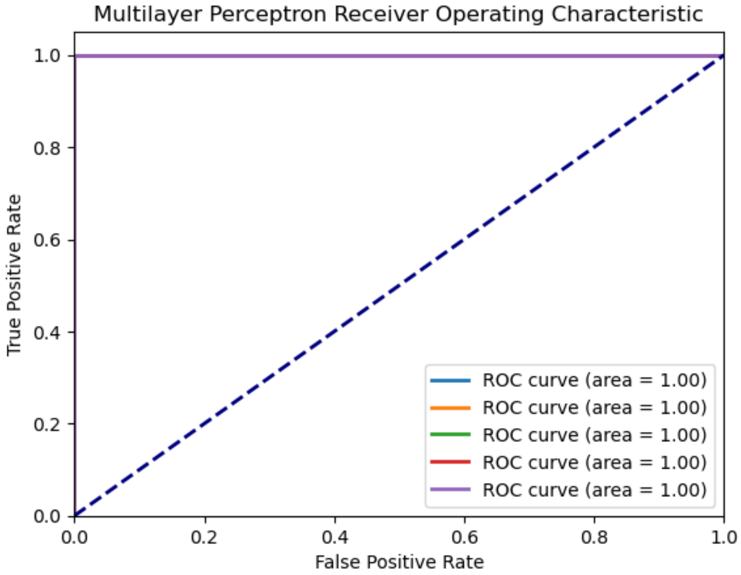
Justification: The multilayer perceptron model can classify complex shapes in datasets. Additionally, they can learn relationships and extract relevant features in the data that may not be easily identifiable by humans.

Pros: Able to classify both linearly and non-linearly separable data.
Handles high dimensional data with good scalability.

Cons: Computationally expensive.

Parameters: The logistic/sigmoid activation function was chosen because all features lie between a range of 0 and 1. Potentially, a softmax activation function could have been used however, it doesn't appear as though the sklearn library supports softmax activation.
Hidden layers were reduced from the default 100 to 10 as the number of features used were quite small and a larger hidden layer size may cause overfitting in the model.
Learning rate was kept at 0.001 due to the limited range of each feature and the activation function used.

Results: The final MLP model produced a perfect true positive and false negative rate across a five-fold cross-validation.



Support Vector Machine

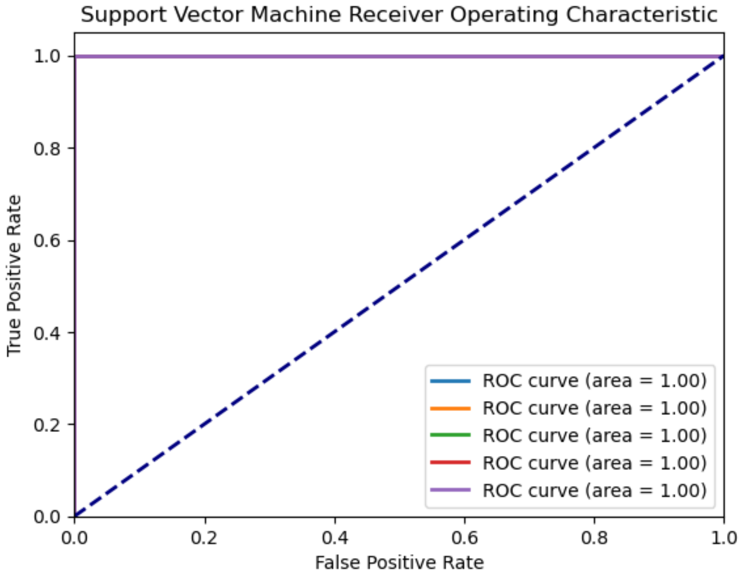
Justification: Features were chosen according to perceived linear separability. Hence, using a Support Vector Machine to draw a hyperplane should be reasonably simple.

Pros: Effective in high dimensional space.
Decision boundaries are simple and easily interpretable.

Cons: Quadratic training time is computationally expensive for large datasets.
Some kernels train in perpetuity depending on the shape of the data. Thus, it is imperative to have an understanding on the shape of the data before selecting a kernel.

Parameters: A polynomial kernel was chosen to generate the hyperplane. It was chosen over a linear kernel as it is more resilient to separability that isn't strictly a linear function while also being able to produce linear separable hyperplanes.

Results: Across all folds the cross-validation shows that the model was able to predict all classes with a 100% true positive rate and 0% false positive rate given the target B=1.



Random Forest

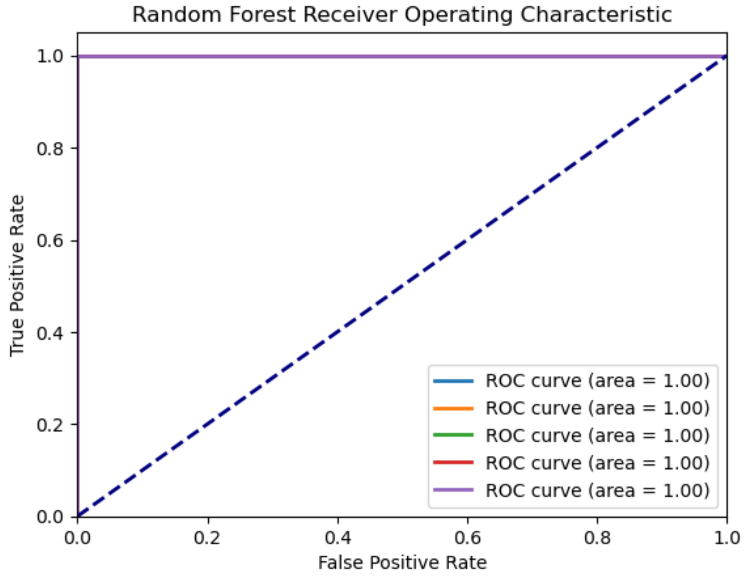
Justification: In the same way as decision trees, random forest is appropriated because most of the chosen features are easily separable. Random forests are the same as decision trees except with multiple iterations.

Pros: Computationally inexpensive.
More robust than decision trees.
Less prone to overfitting due to the simplicity of the model.

Cons: Prone to bias based on given training split.

Parameters: Max depth was chosen to continue until there are only pure leaf nodes remaining because for some features like “Teff” and “Dist” although it is generally surmisable that the feature is separable by a single line, the exact number of clusters is unknown. Hence, the model will continuously split until no impure clusters remain.
Minimum samples per split was kept at two for the same reason as it is unknown how many clusters exist in certain features.
N estimators were reduced to 30 from the default 100 value as the features chosen weren’t particularly complex. The number of estimators is still reasonably high such that the robustness of the random forest model is still present compared to a single decision tree.

Results: Given the target B=1 the model was able to predict all B labels with a complete accuracy maintaining a 100% true positive rate and 0% false positive rate across a K-fold cross-validation where k=5.



Ensembling

Justification: The results of each previous model produced highly accurate results such that it is difficult to identify which model is the most appropriate for the final classification task. As such, it was decided that an bagging ensembling approach would produce the best results.

Pros: More resilient to bias among models.

Cons: Expensive both computationally and in time as it requires prediction using all models to produce a result.

Parameters: All models excluding decision trees were chosen for ensembling as it is redundant with the random forest classifier.

Results: Testing of the results for this task were performed through the Kaggle platform. Results of the competition evaluation produced a result of a 1.00000 public score and a 1.00000 private score.

Reflection

Why stop at only null features, were there any other trends present?

The identification of null values being indicative of was the most obvious trend. Although there may be other trends that were present in the given dataset, they were not identified in this study. Increased development time and collaboration with other researchers could potentially help in identifying any other significant trends.

Why create null features and then remove them?

The features that were removed from the training set were redundant to others. For example, “GRVSmag” would share null features with “e_GRVSmag” as values that were originally null would not have a measurable error. Although not entirely the same, null features for “Lum-Flame”, “z-Flame”, and “Mass-Flame” were removed because the information they provided was also present in “Age-Flame isnull”. Removing these redundancies is necessary for model training where duplicate features could effectively skew the classification results.

Why were so few features selected?

According to the analysis framework, only a few features were initially selected for model training. Depending on the results of the trained model, feature selection and model parameters would be reevaluated. However, all models performed with very high accuracy across a k-fold cross-validation. Additional development time could be used to evaluate the effect of certain features on the models.

Were the selected models the best ones for the task?

The selected models were chosen based on the researcher’s background knowledge and their perceived appropriateness to the given task. Although there may be models that approach the classification task more appropriately, it would be incorrect to apply a model without first having an understanding on how the model achieved the predicted results in the same way that it is inappropriate to perform model training without identifying trends in the initial data exploration.

Conclusion

In this report, a classification prediction model was built for the client to identify the spectral type of celestial bodies in the Gaia system. Initial data exploration was performed on the given labelled dataset to identify the distributions of A and B labels and any trends associated with them. It was found that of the raw features in the dataset “Teff”, “Dist”, “Source”, and “ID” were the most indicative of spectral type due to their perceived separability.

Additionally, it was found that entries with null features in “GRVSmag”, “e_GRVSmag”, “pscol”, “Lum-Flame”, “z-Flame”, “Mass-Flame”, and “Age-Flame” were indicative of the B spectral type. Hence, new features were created to indicate whether an entry was initially null via binary representation. The usage of these features created highly separable data for further model training.

For the sake of model selection and training, the target variable “SpType-ELS” was recoded from “A” and “B” to 0 and 1 respectively. The features that were chosen were composed of sigmoid normalized features that showed to have high separability between each label type and only the significant null features that told unique information.

Five models were tested using a k-fold cross-validation for robustness and each fold was plotted along an ROC curve for decision tree, k-nearest neighbour, multilayer perceptron, support vector machine, and random forest classifiers. Both the ROC and accuracy of each model showed their predictions to be highly effective.

As there was no clearly defined best model, each of the models excluding decision trees were aggregated into an average prediction on the unlabelled testing set for the final ensemble model.