Loan Approval Prediction Report

Paula Lozano Gonzalo

February 20, 2025

1 Introduction

This report outlines the process of developing and evaluating classification models for predicting loan approval using the Kaggle Playground Series - Season 4 - Episode 10 Loan Approval Prediction dataset. The goal was to train multiple classification models, perform hyperparameter tuning using cross-validation, and submit predictions to the Kaggle competition.

2 Data Understanding

The dataset contains the following features:
(I had to do some searches to understand what each thing was because I am not familiar with all the terms and there was no explanation of the features in kaggle like in the last assignment. That is why I am adding this into the repport)

- **id**: Unique identifier for each record.
- **person_age**: Age of the applicant.
- **person_income**: Annual income of the applicant.
- **person_home_ownership**: Type of home ownership (e.g., 'own', 'rent', 'mortgage').
- **person_emp_length**: Employment length in years.
- **loan_intent**: Purpose of the loan (e.g., 'education', 'medical', 'personal').
- **loan_grade**: Loan risk grade (e.g., 'A', 'B', 'C').
- **loan_amnt**: Loan amount requested.
- **loan_int_rate**: Loan interest rate (APR).
- **loan_percent_income**: Percentage of income allocated to loan repayment.
- **cb_person_default_on_file**: Binary indicator of default history (yes/no).
- **cb_person_cred_hist_length**: Length of credit history in years.
- **loan_status**: Loan status (0 or 1).

3 Data Exploration and Visualization

When I plotted the histograms and the scatter plots I realized that since a lot of the data is polarized you can barely see the points. Because of this I used the histograms to visualize the distribution of features. (I will leave some examples bellow)

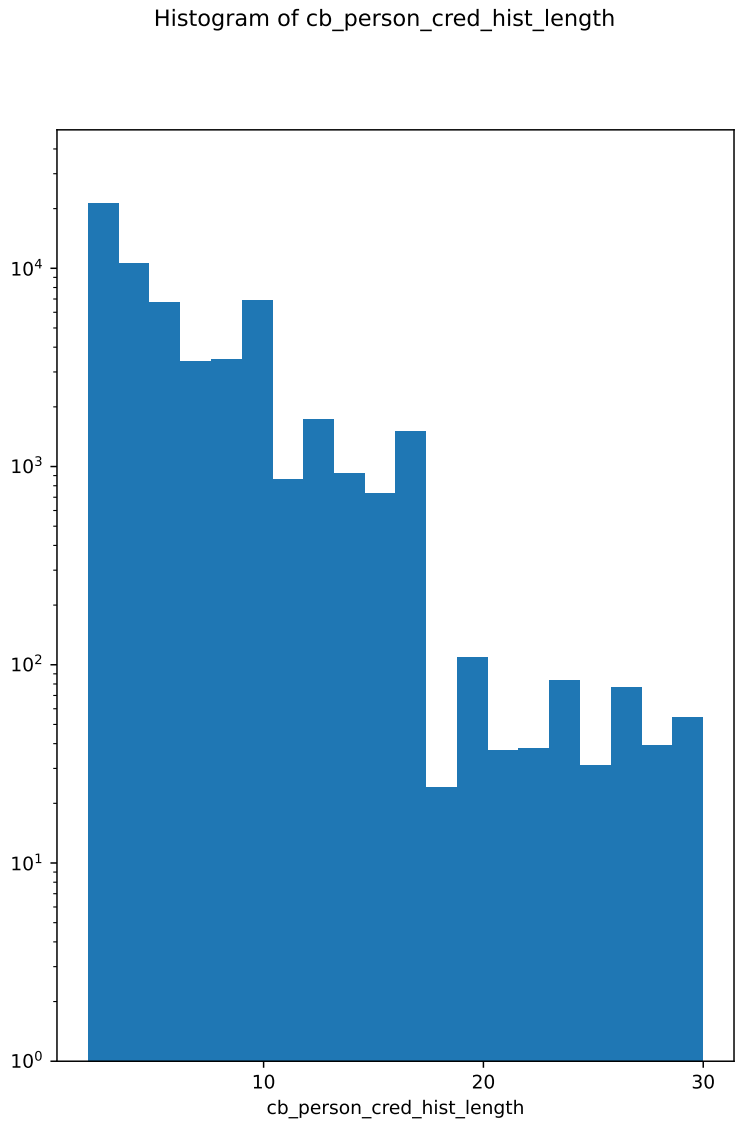

Figure 1: Credit History Lenght

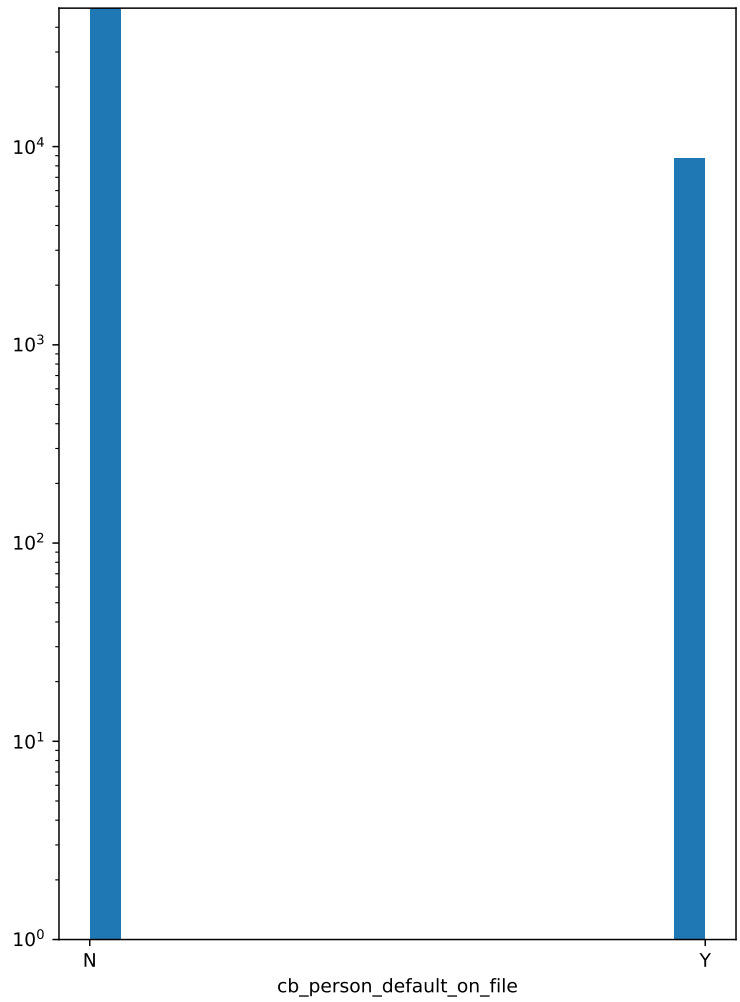

Figure 2: Does the person have default history?

4 Model Development and Evaluation

Three models were trained and evaluated:

1. **SGDClassifier**
2. **RandomForestClassifier**
3. **GradientBoostingClassifier**

4.1 SGDClassifier

- **Best Score (Cross-Validation)**: 0.9144

- **Confusion Matrix**:

```
t/p      F     T
   F 48509.0 1786.0
   T 6518.0 1832.0
```

- **Precision**: 0.506
- **Recall**: 0.219
- **F1 Score**: 0.306
- **Analysis**: The model performed poorly, with only 50.6% of positive predictions being correct and 21.9% of actual positives identified.

The best hyperparameters found for the SGD classifier were:

```
{
  'features__categorical__categorical-features-only__do_numerical': False,
  'features__categorical__categorical-features-only__do_predictors': True,
  'features__categorical__encode-category-bits__categories': 'auto',
  'features__categorical__encode-category-bits__handle_unknown': 'ignore',
  'features__categorical__missing-data__strategy': 'most_frequent',
  'features__numerical__missing-data__strategy': 'median',
  'features__numerical__numerical-features-only__do_numerical': True,
  'features__numerical__numerical-features-only__do_predictors': True,
  'features__numerical__polynomial-features__degree': 2,
  'model__alpha': 0.0001,
  'model__average': False,
  'model__class_weight': None,
  'model__early_stopping': False,
  'model__epsilon': 0.1,
  'model__eta0': 0.0,
  'model__fit_intercept': True,
  'model__l1_ratio': 0.15,
  'model__learning_rate': 'optimal',
  'model__loss': 'log_loss',
  'model__max_iter': 1000,
  'model__n_iter_no_change': 5,
  'model__n_jobs': -1,
  'model__penalty': None,
  'model__power_t': 0.5,
  'model__random_state': None,
  'model__shuffle': True,
  'model__tol': 0.001,
  'model__validation_fraction': 0.1,
  'model__verbose': 0,
  'model__warm_start': False
}
```

4.2 RandomForestClassifier

- **Best Score (Cross-Validation)**: 0.9499
- **Confusion Matrix**:

```
t/p      F     T
   F 49754.0 541.0
```

```
T 2355.0 5995.0
```

- **Precision**: 0.917
- **Recall**: 0.718
- **F1 Score**: 0.805
- **Kaggle Score**: 0.93810
- **Analysis**: The model performed significantly better, with 91.7% of positive predictions being correct and 71.8% of actual positives identified.

The best hyperparameters found for the RandomForest classifier were:

```
{
  'features__categorical__categorical-features-only__do_numerical': False,
  'features__categorical__categorical-features-only__do_predictors': True,
  'features__categorical__encode-category-bits__categories': 'auto',
  'features__categorical__encode-category-bits__handle_unknown': 'ignore',
  'features__categorical__missing-data__strategy': 'most_frequent',
  'features__numerical__missing-data__strategy': 'median',
  'features__numerical__numerical-features-only__do_numerical': True,
  'features__numerical__numerical-features-only__do_predictors': True,
  'features__numerical__polynomial-features__degree': 2,
  'model__bootstrap': True,
  'model__ccp_alpha': 0.0,
  'model__class_weight': None,
  'model__criterion': 'gini',
  'model__max_depth': None,
  'model__max_features': None,
  'model__max_leaf_nodes': None,
  'model__max_samples': None,
  'model__min_impurity_decrease': 0.0,
  'model__min_samples_leaf': 1,
  'model__min_samples_split': 2,
  'model__min_weight_fraction_leaf': 0.0,
  'model__monotonic_cst': None,
  'model__n_estimators': 500,
  'model__n_jobs': -1,
  'model__oob_score': False,
  'model__random_state': None,
  'model__verbose': 0,
  'model__warm_start': False
}
```

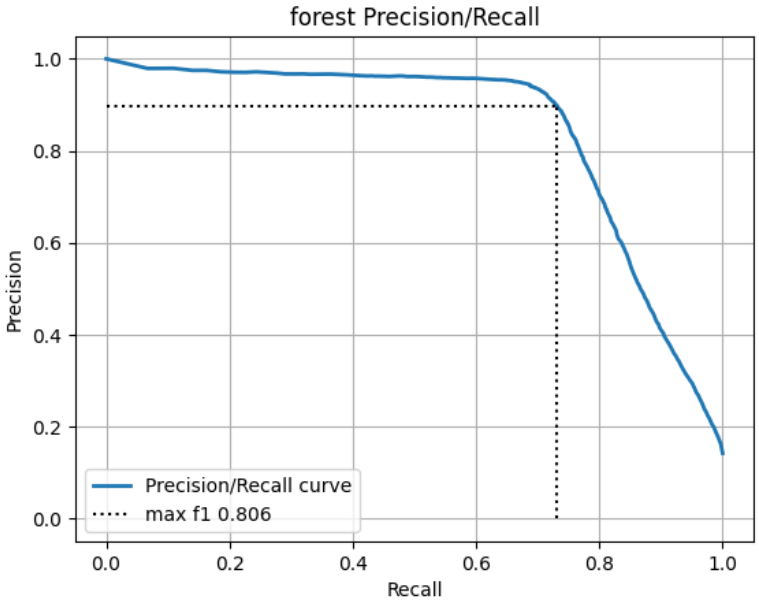

Figure 3: PR

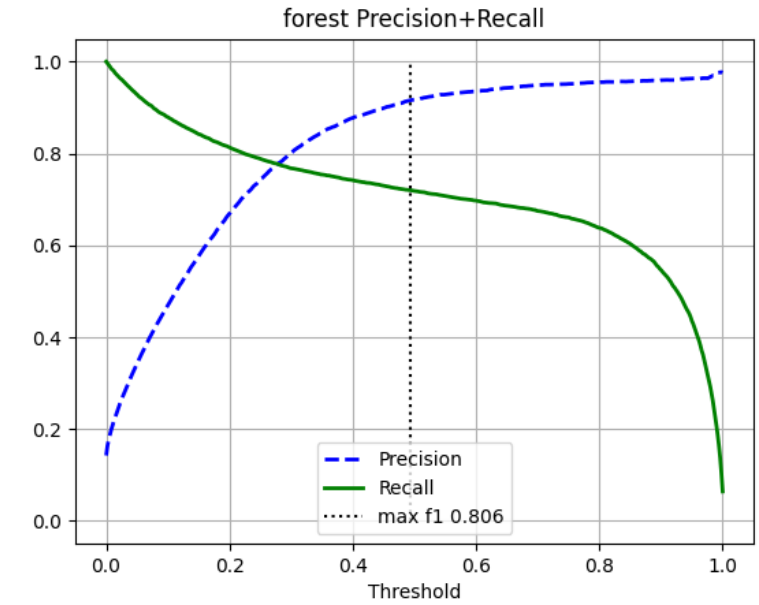

Figure 4: Precision/Recall

I have to say that I don't fully understand these graphs because I am not really familiar with them but I still wanted to plot them and look at them to see how good/bad they were.

4.3 GradientBoostingClassifier

- **Best Score (Cross-Validation)**: 0.9490

- **Confusion Matrix**:

```
t/p      F     T
   F 49775.0 520.0
   T 2359.0 5991.0
```

- **Precision**: 0.920
- **Recall**: 0.717
- **F1 Score**: 0.806
- **Kaggle Score**: 0.94164
- **Analysis**: The model performed slightly better than RandomForestClassifier, with 92.0% of positive predictions being correct and 71.7% of actual positives identified.

The best hyperparameters found for the GradientBoosting classifier were:

```
{
  'features__categorical__categorical-features-only__do_numerical': False,
  'features__categorical__categorical-features-only__do_predictors': True,
  'features__categorical__encode-category-bits__categories': 'auto',
  'features__categorical__encode-category-bits__handle_unknown': 'ignore',
  'features__categorical__missing-data__strategy': 'most_frequent',
  'features__numerical__missing-data__strategy': 'median',
  'features__numerical__numerical-features-only__do_numerical': True,
  'features__numerical__numerical-features-only__do_predictors': True,
  'features__numerical__polynomial-features__degree': 2,
  'model__ccp_alpha': 0.0,
  'model__criterion': 'squared_error',
  'model__init': None,
  'model__learning_rate': 0.1,
  'model__loss': 'exponential',
  'model__max_depth': None,
  'model__max_features': None,
  'model__max_leaf_nodes': None,
  'model__min_impurity_decrease': 0.0,
  'model__min_samples_leaf': 1,
  'model__min_samples_split': 2,
  'model__min_weight_fraction_leaf': 0.0,
  'model__n_estimators': 200,
  'model__n_iter_no_change': None,
  'model__random_state': None,
  'model__subsample': 0.9,
  'model__tol': 0.0001,
  'model__validation_fraction': 0.1,
  'model__verbose': 0,
  'model__warm_start': False
}
```

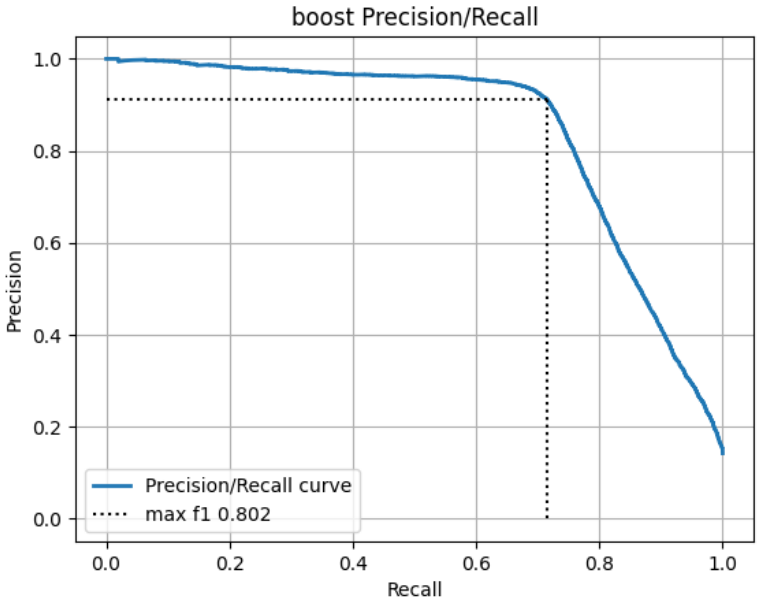

Figure 5: PR

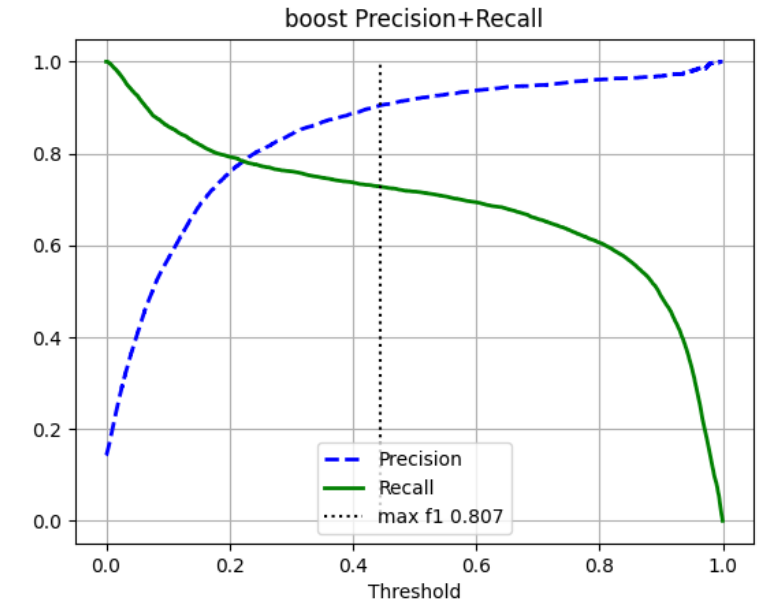

Figure 6: Precision/Recall

5 Model Analysis

- **SGDClassifier**: Performed poorly, probably because to the complexity of the data.

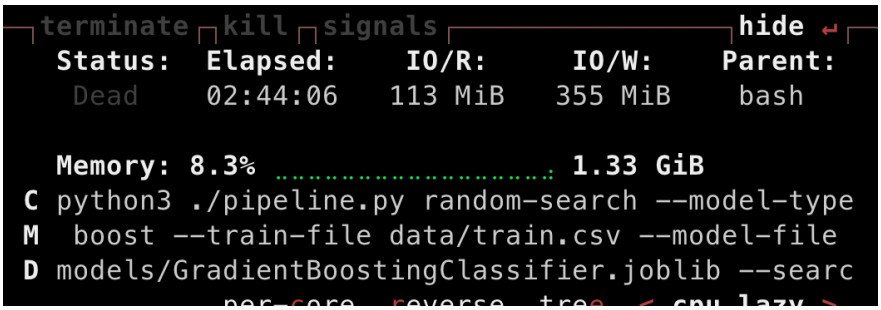

Figure 7: Fun fact. Time boost took to run.

- **RandomForestClassifier**: Performed well, achieving a high Kaggle score and demonstrating good precision and recall.
- **GradientBoostingClassifier**: Slightly outperformed RandomForestClassifier, indicating that boosting techniques may be more effective for this dataset.

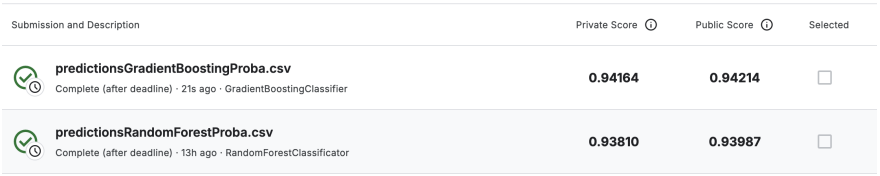

Figure 8: Kaggle scores.