Milestone 2: Airbnb Zero Review Classification

VANDERBILT UNIVERSITY

Executive Summary

Project Goal: Target variable is predict the ‘property_quality’ for Airbnb listings with zero reviews.

- Classifying zero review listings on ‘property_quality’ will eliminate bias when booking a newly listed Airbnb.

Data Cleaning: Created a single dataset that contains 184,984 rows with 27 columns, which encompasses 15 cities

- Narrowed 75 columns to 22 columns before doing feature engineering.
- Created imputation logic and removed entries where price or accommodates is either 0 and/or missing.

EDA: Generated plots on both a regional and collective 15 cities basis

- Plots showed that there are only minimal regional differences for amenities.

Feature Transformation: Conducted six feature transformations to pass into modelling

- New features include aggregating amenities, identifying key host information, and parsing out property information in a more meaningful manner.

Modelling: Decision Tree Classifier

- Encoded categorical variables, selected relevant features, split the dataset, trained a model, evaluated its performance, predicted property quality on 0 review data new data

Data Cleaning

1. Addressing Missing Values

- **Bathrooms**: Fill in the same numeric value as 'bedrooms'. If the 'bedrooms' column is also N/A, fill N/As with 1/'beds'
- **Bedrooms** Fill in with the rounded up value of # of bathrooms
- **Beds**: Fill in with the same numeric value as 'bedrooms'
- There was no case that all three columns are NULL.
- **Host_since**: Put the most recent date that is available in host_since column
- **Host_location**: Put 'unknown', which is already in the dataset
- **host_is_superhost**: Put 'f' as false
- **host_verifications**: Put 'None' for NAs and entries with empty list ('[]')
- **host_identity_verified**: Put 'f' as false
- **host_has_profile_pic**: Put 'f' as false

2. Delete Unnecessary Columns

- Left with:
- 'id', 'host_id', 'host_since', 'host_location', 'host_is_superhost', 'host_listings_count', 'host_total_listings_count', 'host_verifications', 'host_has_profile_pic', 'host_identity_verified', 'neighborhood', 'latitude', 'longitude', 'room_type', 'accommodates', 'bathrooms', 'bathrooms_text', 'bedrooms', 'beds', 'amenities', 'price', 'number_of_reviews','review_scores_value','calculated_host_listings_count'

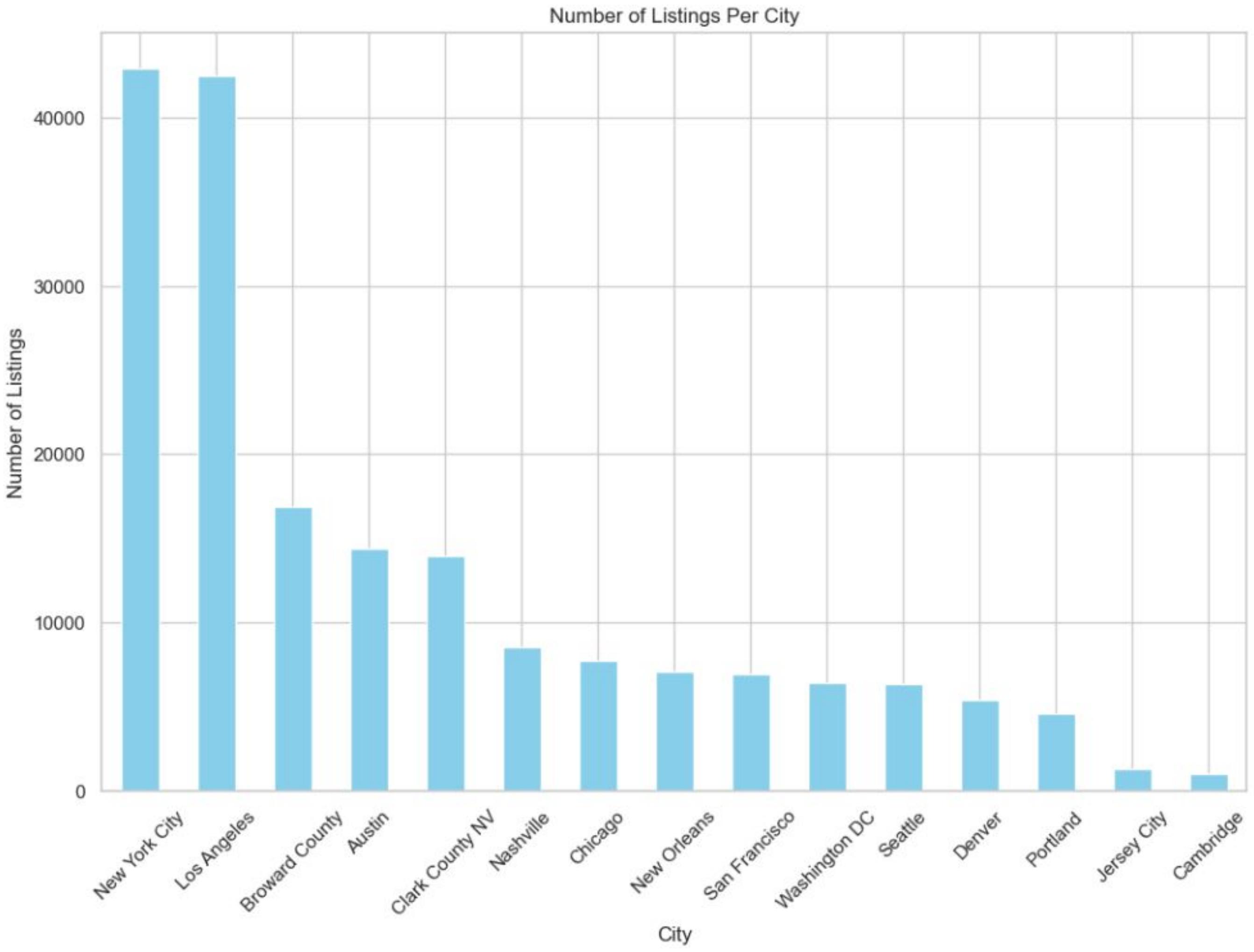
3. Remove Entries Where 'price' and 'accommodates' are Zero or NA

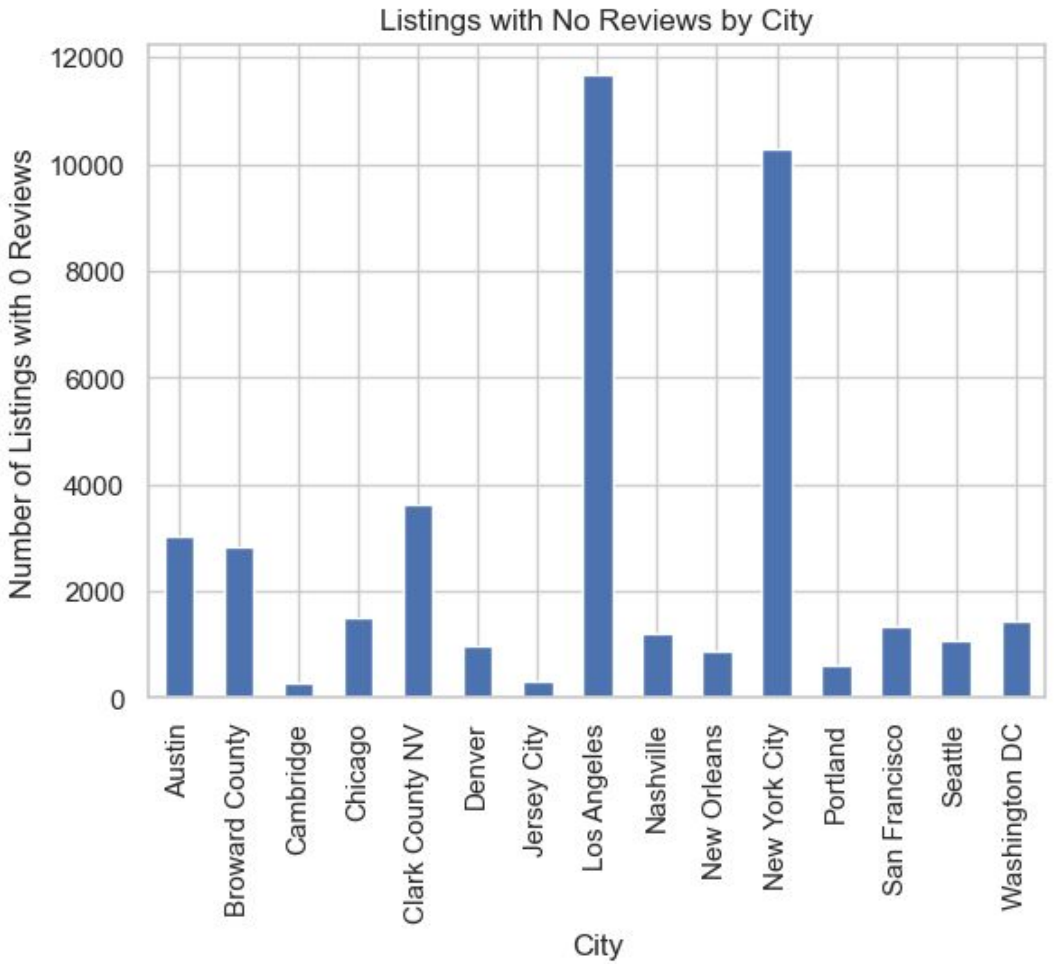
4. Add New Column 'city'

- For the purpose when concatenating all 15 cities all together.

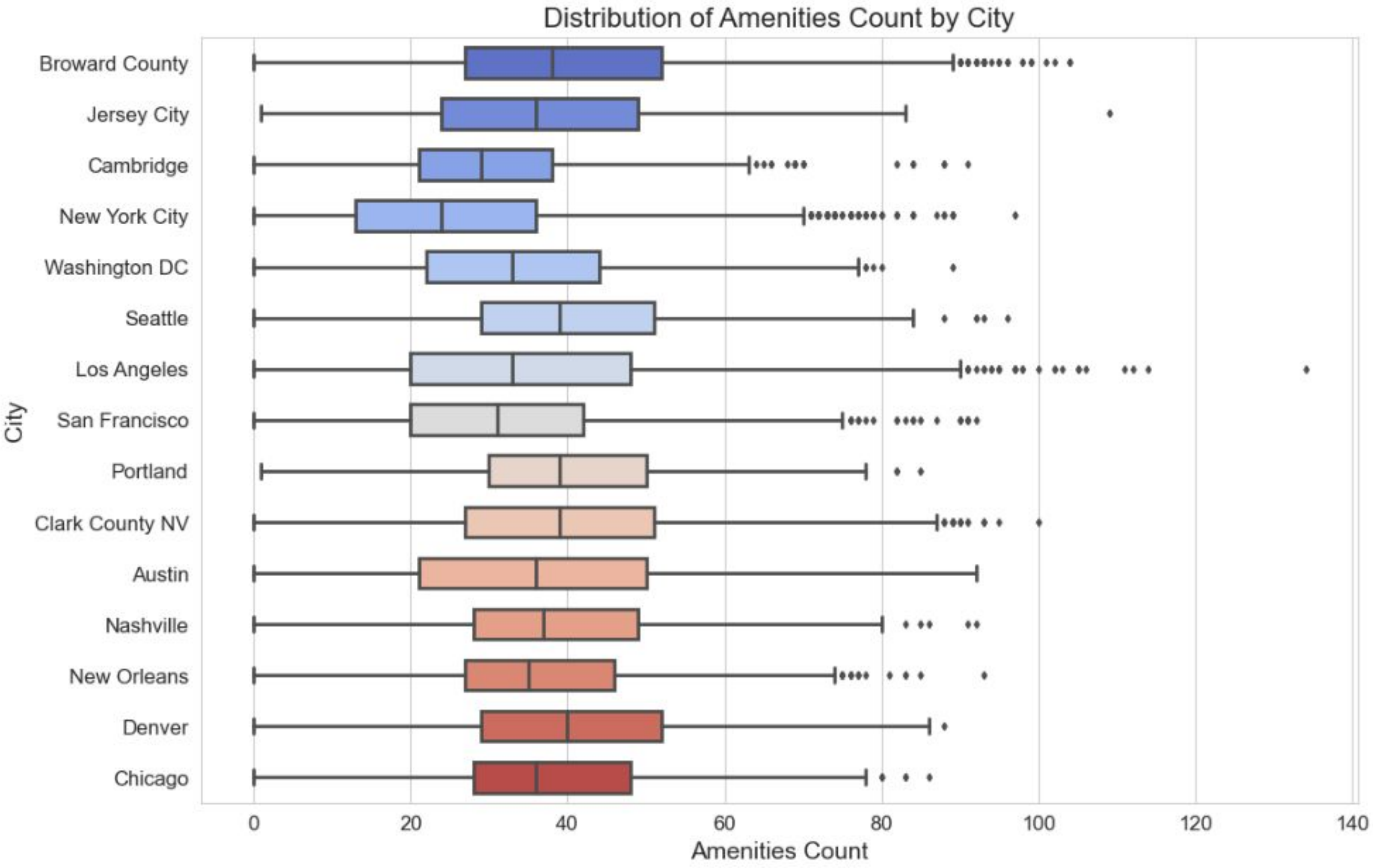
EDA

- Conducted EDA for each U.S. region and on the collective dataset encompassing all regions
  - Created plots for time-series, categorical, numerical, and geojson data
  - After exploring seven different tables for each city, we determined to focus on the listings_detailed tables for further EDA understanding.
- Important Highlights:
  - Identified that many listings with no reviews will likely have review_scores_value missing
  - Determined that having host_response_time is not substantially related to zero reviews
  - The median number of amenities per listing is similar for all cities
  - Airbnb experienced rapid business expansion in the mid-2010s through host sign-ups
  - Despite regional differences, many listings contain the five essential amenities across cities
  - Time-series price data shows seasonality or disruptions from COVID





VANDERBILT UNIVERSITY



VANDERBILT UNIVERSITY

Feature Transformations

1. **Count Amenities**
   a. Counted total number of amenities each listing has
2. **Count Essential Amenities**
   a. essential_ammenities = [Fridge, AC, Kitchen, WiFi, Essentials]
   b. Make new column called essential_amenities and count how many are included in the listing (max num: 5, min: 0)
   c. Error handling to include amenities that have similar name
3. **Bathroom**
   a. Parse out to include the decimal digit for bathrooms, and deleted text part to make it numeric column
4. **City-Neighborhood**
   a. Make a new column that has city and neighborhood together to resolve potential naming duplicates in neighborhoods
   b. For example, Hollywood, CA, and Hollywood in Broward County Florida
5. **Full-time Host**
   a. If host_listings_count is >= 10, the hosts are more experienced/dedicated hosts who treat AirBnB as a full-time job
   b. 'T' for true if >=10, and 'F' for false if <10
6. **Host Verification**
   a. Due to Spark data read-in issue, change the host_verification column to be string, not list object.

```
essential_amenities_normalized = {
    'fridge': ['fridge', 'refrigerator', 'mini fridge',
               'mini-fridge', 'energy-efficient refrigerator'],

    'essentials': ['essentials', 'essential',
                   'bathroom essentials', 'bedroom essentials'],
    'ac': ['air conditioning', 'air conditioner',
           'ac', 'a/c', 'central air', 'hvac', 'ac unit'],
    'kitchen': ['kitchen', 'full kitchen'],
    'wifi': ['wifi', 'wi-fi', '5g wifi', '5g wi-fi',
             'fast wifi', 'fast wi-fi', 'wireless internet',
             'high-speed internet']
}
```

```
def categorize_verifications(verification_str):
    # Mapping of verification strings to codes
    verifications_to_code = {
        "['phone']": 'p',
        "['email']": 'e',
        "['email', 'phone']": 'ep',
        "['email', 'work_email']": 'ew',
        "['phone', 'work_email']": 'pw',
        "['email', 'phone', 'work_email']": 'epw',
        "['email', 'phone', 'photographer', 'work_email']": 'eppw',
        "['email', 'phone', 'photographer']": 'epp',
        'None': 'none'  # Handling the string 'None'
    }
    # Return the corresponding code or a default value if not found
    return verifications_to_code.get(verification_str, 'none')  # U
```

Features

Original Dataset:
185,989 Rows, 75 Columns

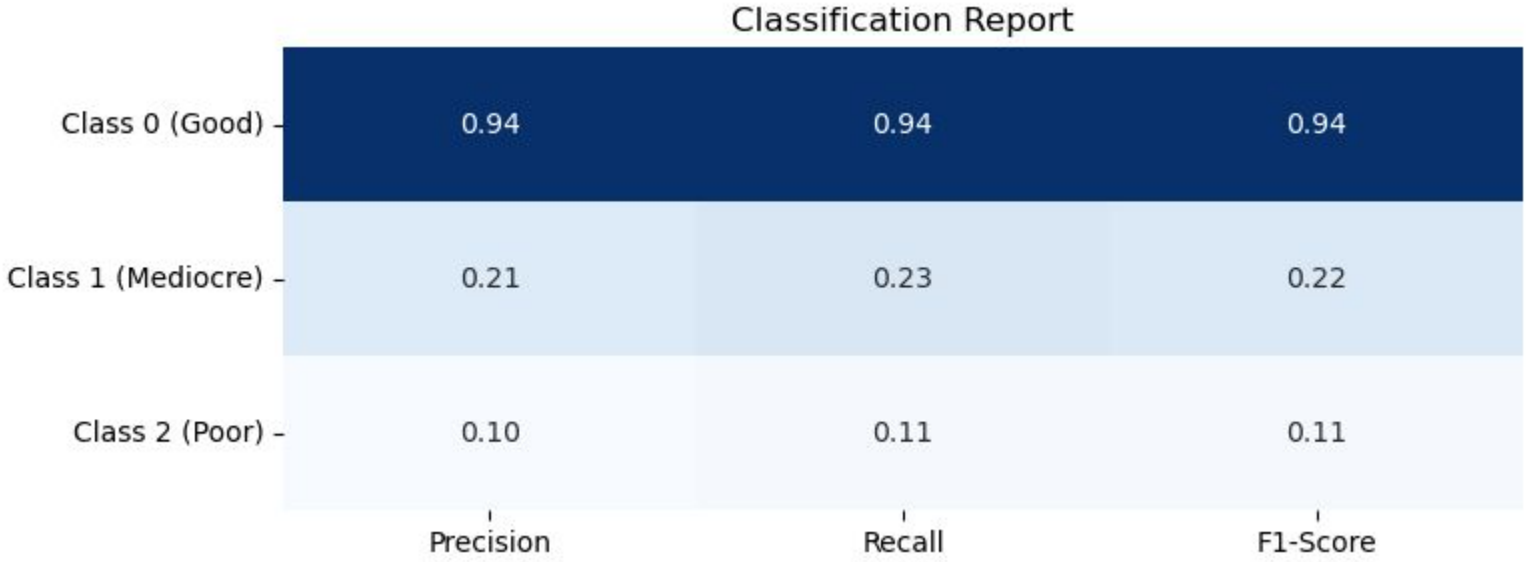
After Cleaning and Feature Transformation:
184,984 Rows, 27 Columns

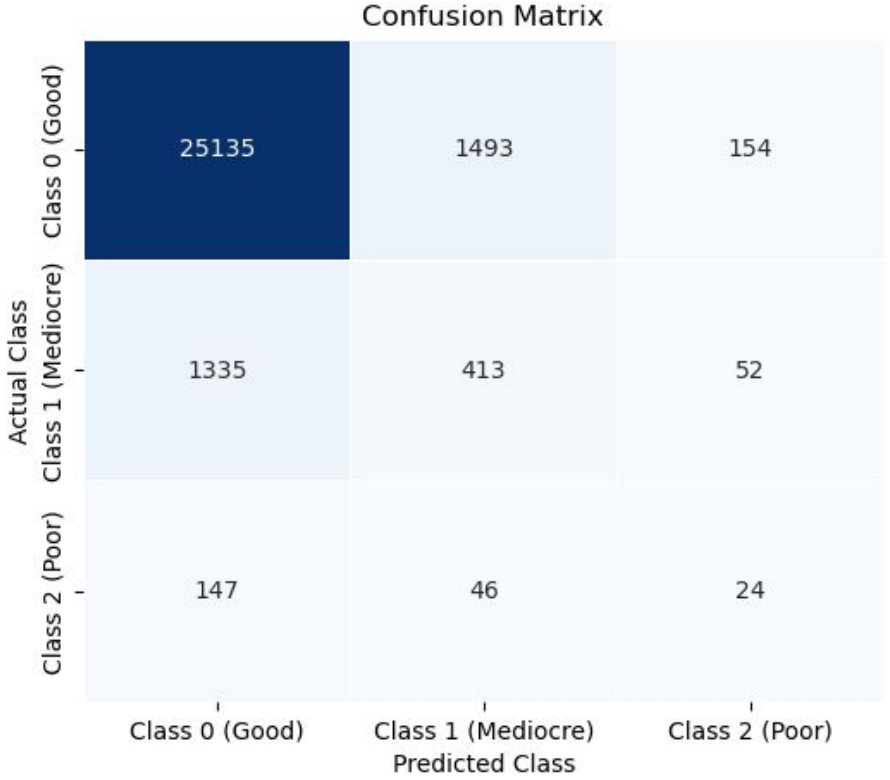
amenities_count	essential_amenities	num_bath	neighborhood_city	full_time_host	host_verifications_clean
10	3	1.0	Fort Lauderdale Broward County	f	p
29	3	2.0	Pompano Beach Broward County	f	ep
14	3	3.0	Southwest Ranches Broward County	f	ep
22	4	2.0	Pompano Beach Broward County	f	pw
17	3	2.0	Hollywood Broward County	f	ep
...	...	...	...	...	...
40	5	1.0	Lincoln Park Chicago	f	ep
44	5	2.0	Logan Square Chicago	f	ep
40	5	1.5	Uptown Chicago	f	epw
43	5	1.0	West Town Chicago	f	ep
36	5	1.0	West Town Chicago	f	ep

VANDERBILT UNIVERSITY

Modelling

1. **Baseline Model**
    a. Decision Tree
2. **Steps**
    a. Preprocess Data
        i. Check missing values
        ii. Encode categorical variables
        iii. Feature selection
    b. Splitting Data
    c. Train the Decision Tree Classifier
    d. Model Evaluation
        i. Precision, Recall, F1-score
        ii. Confusion Matrix


Classification Report
Class 0 (Good)
Class 1 (Mediocre)
Class 2 (Poor)
0.94
0.94
0.94
0.21
0.23
0.22
0.10
0.11
0.11
Precision
Recall
F1-Score


Confusion Matrix
Actual Class
Class 0 (Good)
Class 1 (Mediocre)
Class 2 (Poor)
25135
1493
154
1335
413
52
147
46
24
Class 0 (Good)
Class 1 (Mediocre)
Class 2 (Poor)
Predicted Class

Conclusion & Open Issues

1. **Initial Model Output**
   a. Precision - Good: 0.94, Mediocre: 0.21, Poor: 0.10
2. **Open Issue 1: Modeling**
   a. Hyperparameter tuning
   b. More diverse model such as Random Forest, Ensemble Model
   c. Parallel processing pyspark
      - How are we planning to split the data efficiently and process it parallely
   d. Convert string columns to be numeric, and put it into vector for Spark modeling

3. **Open Issue 2: Spark**
   a. Spark data loading issue when special characters in the data matches to the file's delimiter