Setup

1. Clone the repository to get started:
    1. https://github.com/jkonieczny2/grainger_data_science.git
2. Start the virtual environment:
    1. `source env/bin/activate`
3. Run the following command in the root directory:
    1. `./run.sh`

The “run.sh” script will perform the following tasks:

1. Download the parking citations data from S3 and save it in data/parking_citations.corrupted.csv
2. Separate the corrupted and uncorrupted data into two files:
    1. data/parking_citations_corrupted.csv
    2. data/parking_citations_uncorrupted.csv
3. Execute the following script to load the data/parking_citations_uncorrupted.csv data and train a Random Forest model to predict the probability that a given citation involves a vehicle belonging to the top 25 makes in the uncorrupted sample:
    1. scripts/SKLearnRandomForest-ForScript.py
        1. Note: this script will serialize the model and any Encoders used to convert categorical data. These outputs will be saved in the following location:
            1. app/models/
4. Start a Flask server running on localhost:5000 to respond to requests for predictions.

Note that the Flask server will run in the terminal; in order to query it you will need to open a new terminal.

Flask Server

A sample prediction can be obtained from the Flask server (code can be found in app/app.py) by running the following command in a Unix terminal:

curl -X POST -H "Content-Type: application/json" localhost:5000 -d @app/test_post.json

Alternatively, you can run the following script:

app/test_server.sh

You should see the following output containing the prediction class and probability for the sample data:

{"prediction":"1","probability":"0.900691078854698"}

Please note that although the server does convert the raw JSON into the appropriate format for consumption by the trained model (e.g. encoding categorical data, converting latitude and longitude to distance from the city center), it does not gracefully handle missing data, improperly formatted data, or categorical variables that are not part of the original training sample.

In a future iteration of development, the server would handle all of these edge cases. However, for

the purposes of this assessment, it is highly recommended to use the app/test_post.json sample data only.

Analysis

A summary of the results of each assessment can be found in the analysis/ directory. Grainger_Assessment_1.pdf contains a summary of the model deliverable, and the Grainger_Assessment_2.pdf file contains a summary of the Pandas/Sqlite comparison assessment.