Introduction

For this exercise, we have compiled a small data set of auto insurance policies that we would like to make accessible so that other teams within Ledger can access the data in a consistent and meaningful way.

This might include

1. A software engineer at Ledger implementing a visualization or chart that allows a user to dynamically explore the data set
2. A data scientists needs a quick summary of the data set for a report

Given this information, your task is to expose the information through a REST API that allows the consumer to aggregate, filter, and pull calculated metrics from the data.

Common data exploration would be:

- Give me the <metric_name1>,<metric_name2> by specific <feature_name> where <feature_name> is <feature_value>
- Allow me to filter policy records with certain values for <metric_name>, <feature_name>
- Show me metrics for policies than belong to certain feature clusters.

Definitions:

General:

Feature: is an attribute of an insurance policy

Metric: can be a count or amount

Commonly requested calculated metrics:

Severity: == sum_of_losses(feature) / total_claims_count(feature)

Frequency: == total_claims_count(feature) / policy_count(feature)

Loss ratio: == sum_of_losses(feature) / sum_of_premium(feature)

About the data

Attached is a basic data set of 200,000 auto insurance policies (auto_policies.csv ~14MB).

Notes on columns

- Columns prefixed with "**driver_**" or "**vehicle_**" are features
- Columns prefixed with "**insurance_**" are metrics
- "**year**" and "**month**" correspond to the start year and month of the policy

Requirements

- Please do not spend more than 4 hours on this. If you are unable to complete the case study, send over what you have by pushing your code to a public GitHub repo.

- We should be able to clone and run this project with instructions in a readme. If you use a database, please provide instructions for bootstrapping the data.
- We prefer that you use Python or Javascript but understand there might be more appropriate tools for the job. Please reach out if you'd like to use a different language.
- The case study is open-ended as this is generally how we work at Ledger so feel free to show us what you're capable of.
- Please do not hesitate to reach out if you have any questions.

Evaluation

As a senior engineer, we will be evaluating your ability to think beyond just implementation.

- Ability to communicate (written and verbally) your thought process
- Empathy for your team - is the code documented, readable, extensible, testable, etc.
- We don't expect the code to be production-ready but be prepared to discuss tradeoffs and decisions or what you would have done given more time.

-