Problem Statement

ACRTA road taxation data engineering

Astro City Road transport authority (**ACRTA**) in US have come up with an idea to use car registration renewal charges to provide indirect incentives to safe drivers. Also, providing subsidies to certain areas as per the extreme climatic conditions in terms of heavy snow or rain.

ACRTA has contacted us to perform a quantitative study and design a prediction model to support the aforementioned applications.

We, being a part of the data engineering team, are working continuously with the business stakeholders as well as data scientists to create features around these scenarios.

Problem statement that we have been provided is to "**Develop inputs for a model that predicts the chances of having a vehicle accident based on driving conditions. This model will help the transport authority to understand risk patterns and act upon them.**"

This output then would be utilized so as to come up for a risk-based taxation on different drivers and locations as per crash-prone weather conditions.

Use cases would be –

1. Imposing "**unsafe driving tax**" on drivers to provide a positive feedback loop which may be revisited every year by looking at the past year trip data based on the driving patterns.
2. Lower the tax in the regions where the climatic conditions lead them to become a crash-prone site.

Data Description

1. **Drive Data (Connected car data)** – Data coming from the car-mounted devices, which provides you with the car statistics every second. This information will include – Speed, acceleration, engine temperature and other car statistics.

Name	Type	Not Null	Primary Key	Description
accel_x	VARCHAR	YES	NO	Not Relevant
accel_y	VARCHAR	YES	NO	Not Relevant
accel_z	VARCHAR	YES	NO	Not Relevant
datetime	INT	YES	YES (Part of key)	timestamp associated with all readings
engine_coolant_temp	INT	NO	NO	Engine Coolant Temperature in Celsius
eng_load	INT	NO	NO	Calculated Engine Load (a percentage, represented from a scale of 0-255)
fuel_level	INT	NO	NO	Fuel Tank Level Input (a percentage, represented from a scale of 0-255)
iat	INT	NO	NO	Intake Air Temperature in Celsius
rpm	INT	NO	NO	Engine RPM
trip_id	VARCHAR	YES	YES (Part of key)	unique identifier for the trip
vehicle_id	VARCHAR	YES	YES (Part of key)	unique identifier for the vehicle
velocity	INT	NO	NO	Vehicle Speed in kilometers per hour (km/h)

2. Trip – Parameters associated with location of car such as lattitude, longitude, altitude and other similar parameters

Name	Type	Not Null	Primary Key	Description
datetime	DATETIME	YES	YES (Part of key)	timestamp associated with all readings
lat	FLOAT	YES	NO	latitude
lon	FLOAT	YES	NO	longitude
trip_id	VARCHAR	YES	YES (Part of key)	unique identifier for the trip
vehicle_id	VARCHAR	YES	YES (Part of key)	unique identifier for the vehicle
velocity	INT	YES	NO	Vehicle Speed in kilometers per hour

3. Weather – Weather condition at different latitude & longitude during different times each day.

Name	Type	Not Null	Primary Key	Description
x	INT	YES	YES (Part of key)	X axis of HRAP Grid data
y	INT	YES	YES (Part of key)	Y axis of HRAP Grid data
date	DATE	YES	YES (Part of key)	Date of weather data
time	VARCHAR	YES	YES (Part of key)	Time of weather data in PST timezone
lat	FLOAT	NO	YES (Part of key)	Latitude
lon	FLOAT	NO	YES (Part of key)	Longitude
temperature_data	FLOAT	NO	NO	Temperature data
temperature_unit	VARCHAR	NO	NO	Temperature unit
precipitation_data	FLOAT	NO	NO	Precipitation data
precipitation_unit	VARCHAR	NO	NO	Precipitation Unit
wind_ew_data	FLOAT	NO	NO	Wind data for direction East-West
wind_ew_unit	VARCHAR	NO	NO	Wind data unit
wind_ns_data	FLOAT	NO	NO	Wind data for direction north south
wind_ns_unit	VARCHAR	NO	NO	Wind data unit

4. Vehicle Specifications – Different vehicle technical specifications which comes from the manufacturer of the car.

Name	Type	Not Null	Primary Key	Description
vehicle_id	INT	YES	YES	Identifier for the vehicle
year	INT	YES	No	Year of manufacturing
make	VARCHAR	YES	No	Brand
Model	VARCHAR	YES	No	Model name for vehicle
drivetrain	INT	YES	No	Drive axels which are directly provided power with

max_torque	INT	YES	No	Maximum torque generated by vehicle
max_horsepower	INT	YES	No	Maximum horsepower of engine
max_horsepower_rpm	INT	YES	No	
max_torque_rpm	INT	YES	No	
engine_displacement	INT	YES	No	
fuel_type	INT	YES	No	Fuel acceptable by the vehicle
fuel_tank_capacity	INT	YES	No	Capacity of vehicle tank
fuel_economy_city	INT	YES	No	Mileage of vehicle in city
fuel_economy_highway	INT	YES	No	Mileage of vehicle in highway
cylinders	INT	YES	No	Number of cylinders
forced_induction	INT	YES	No	
device_generation	INT	YES	No	

Submission/Output Format –

Using the input data as described above, the participant would need to create features related to different hypothesis around the driving behaviours.

Solution file must be a single zip containing the 3 csvs for 3 kinds of features with the file names as given below:

- Engine Features: engine_features.csv
- Drive Features: drive_features.csv
- Weather Features: weather_features.csv

The csv files must have the same file names and columns in order so that checks may be done. **Failure to do so would result in direct rejection as these would be tested through automated scripts.**

To understand the type of features to create, Please go through the requirements for the features carefully:

1. **Engine Features** (file name – engine_features.csv)-

Grain - every vehicle aggregated at week start date(Monday) for the complete week in YYYY-MM-DD format.

Sorted - by Vehicle ID and week_start_Date in ascending manner

Hints:

- Convert timezone to PST before any calculations
- All vehicles from drive data should be in the final output even if you do not have specifications (Fill with 0 if specs are not given)
    - **Active horsepower -** Engine load / 255 * Max Torque * RPM / 5252
    - **Horsepower utilization –** Active horsepower / Max Horsepower
    - **Torque Utilization** - calculated as Engine load/ 255
    - **RPM Utilization** – RPM / Maximum horsepower rpm

Fields and order required in engine_features.csv –

Column name	Description	Sort Order
vehicle_id	Identifier for a specific vehicle	1
week_start_date	Start date of the week for the vehicle journey in YYYY-MM-DD format.	2
ft_torque_util_60pct_s	total time taken in seconds for torque to be utilised more than 60% - (>=60 & < 70)	
ft_torque_util_70pct_s	total time taken in seconds for torque to be utilised more than 70% - (>=70 & < 80)	
ft_torque_util_80pct_s	total time taken in seconds for torque to be utilised more than 80% - (>=80 & < 90)	
ft_torque_util_90pct_s	total time taken in seconds for torque to be utilised more than 90% - (>=90 & < 100)	
ft_horsepower_util_50pct_s	total time taken in seconds for horsepower to be utilised more than 50% - (>=50 & < 60)	
ft_horsepower_util_60pct_s	total time taken in seconds for horsepower to be utilised more than 60% (>=60 & < 70)	
ft_horsepower_util_70pct_s	total time taken in seconds for horsepower to be utilised more than 70% (>=70 & < 80)	
ft_horsepower_util_80pct_s	total time taken in seconds for horsepower to be utilised more than 80% (>=80 & < 90)	
ft_rpm_util_50pct_s	total time taken in seconds for rpm to be utilised more than 50% (>=50 & < 60)	
ft_rpm_util_60pct_s	total time taken in seconds for rpm to be utilised more than 60% (>=60 & < 70)	

2. **Drive features**(file name – drive_features.csv) -

Grain – Every trip's aggregated features at a trip id level.

Sorted - by trip_id in ascending manner

Hints:

- Acceleration m/s is calculated as a change in velocity over time
- If a vehicle keeps on accelerating continuously over a period of time, please treat them as a single acceleration or deacceleration period.

Fields and order required in drive_features.csv -

Column name	Description	Sort Order
trip_id	Identifier of a trip generated through device	1
ft_cnt_vehicle_deaccel_val	Find out the number of times there was deceleration in a trip; if there is a continuous deceleration that is treated as one single deceleration.	
ft_sum_hard_brakes_10_flg_val	Total number of times when hard brakes were applied and min deceleration was less than or equal to -10 m/s (Example: -13 m/s, -15 m/s). If there are multiple hard breaks applied in a given deceleration window then only the minimum deceleration is counted for that period. Example: If there is acceleration of 1, -2, -11, -1, -10, 8, -3, -12 in the order of time for a given trip then count would be 2 i.e. for -11 and -12 and not for -10 as there was continuous deacceleration.	
ft_sum_hard_brakes_3_flg_val	Total number of times when hard brakes were applied i.e. min deceleration to be greater than equal to -3 and less than -10m/s; if there is continuous deceleration then the minimum deceleration is checked for that deceleration window.	
ft_sum_time_deaccel_val	Total time in the trip when the vehicle was decelerating in sec	
ft_cnt_vehicle_accel_val	find out the number of times there was acceleration in a trip ; if there is a continuous acceleration that is treated as one single acceleration.	
ft_sum_hard_accel_10_flg_val	Total number of times when hard acceleration was applied and max acceleration to be greater than equal to 10 ; if there is continuous acceleration then the maximum acceleration is checked for that acceleration window. Example: if there is acceleration of -1, 2, 11, 1, 10, -8, 3, 12 in the order of time for a given trip then count would be 2 i.e. for 11 and 12 and not for 10 as there was continuous acceleration.	
ft_sum_hard_accel_3_flg_val	Total number of times when hard acceleration was applied i.e. max acceleration to be greater than equal to 3 and less than 10 ; if there is continuous acceleration then the maximum acceleration is checked for that acceleration window	
ft_sum_time_accel_val	Total time in the trip when the vehicle was accelerating in sec.	

3. **Weather features** (file name – weather_features.csv)

Here are the weather conditions for your reference and generating weather feature accordingly

Weather Condition	Low end T'(F)	High end inclusive T'(F)
SNOW	-	27
FREEZING RAIN	27	32
RAIN	32	-

Weather Perception Nature	Low end precip. (kg/m^2)	High end precip. (kg/m^2) (Inclusive)
LIGHT	0	2.5
MODERATE	2.5	7.6
HEAVY	7.6	-

Grain – Every vehicle detail should be aggregated at a week start date.

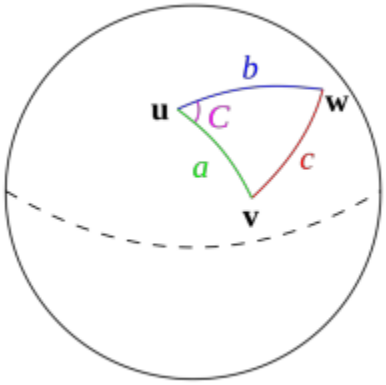
Sorted - by vehicle_id and week_start_date in ascending manner

Hint: convert time zone to PST before any calculations

Assumptions & Hints–

- Weather data is already in PST and may not need any timezone conversion. You may consider the weather data to be constant for complete hour basis. For example- if the temperature is given to be 284.51 for 2017-02-14 19:00:00, it would be the same for time 2017-02-14 19:15:45 as well.
- Haversine formula **must** be utilized to measure the distance between any 2 consecutive points in between the trips.
- Matches in between datasets must be on geohash precision point 5.

Studies have found that - The **haversine formula** determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.



Fields and order required in weather_features.csv -

Column name	Description	Sort Order
Vehicle_id	Identifier for a specific vehicle	1
Week_start_date	Start date of every week starting on a Monday in YYYY-MM-DD format.	2
total_light_rain_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_light_freezing_rain_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_light_snow_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_moderate_rain_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_moderate_freezing_rain_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_moderate_snow_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	
total_heavy_rain_driving_km	Distance travelled in kilometres under specific weather condition during the complete week	

Evaluation Criteria & Rules-

- Least deviation from the actual result would decide the accuracy of the output.
- For engine and drive features, we are going to use the absolute percent deviation in between pre-calculated output versus the output provided by you. For weather features, it is going to be calculated as per mean percentage deviation from the actual output.
- The final score is calculated using the deviation scores after applying weights as follows

Score = 100[W1(1- engine_deviation) + W2(1- drive_deviation) + W3(1- weather_deviation)]

- Quality of code would be judged on the following parameters – functionality, reusability, modularity, documentation, testing and validation.
- Should be scalable to be executed on 5 GB data as well.
- Please note that scoring is going to be done using an automated script and **difference in between the field names or order** from the above-defined feature **may result in zero scoring/error message** due to the failure of the scoring script.
- Participants may do multiple submissions. They would have to select on the platform which one to be treated as the final submission. If not selected, the submission with the highest score would be considered as final.
- Final winners would be announced only after the **submitted code reviews** and the **analysis of the rest of the document submissions** made by the participants.
- Only 5 submissions per day are allowed

Submission Format

Solution checker has 2 upload links - one for solution file (described above) and 1 for code files. Final submission must be done along with code files, however, other submissions can be just made with the solution file.

Code File

Final submission must include the following relevant code files and documents. You can download a sample code file format here.

- **Exploratory data analysis for the raw data** – This should contain all the profiling outputs and plots generated as part of exploratory analysis. Every data source should have an individual html file with the name as that of the data source.
- **Data Quality issues report – Word/ PPT** – High-level issues that are observed while working on the data. A single page for every data source.

- **Codebase (python/Java/Spark)** – Code which contains a central main file for the execution of the complete pipeline and generates the final CSV outputs.
- Final Model input file containing the features –
  - As mentioned above in the Output Format segment.
- **Insights if any – Word/Plot/PPT –**
  - Do you see any information which could be useful to come to final output for the ACRTA.