

**Practical AI**

**ST1508**

**CA1 Project Report**

**Submitted By:**

Liang Zheng Kai Lucas - P2222770

Khoo Yan Lun Darius - P2222259

Ng Sheng Kang Ernest - P2222668

**Submitted To:**

Mr Wilson Qiu

AY 2023/2024 Sem 2

**Table of Contents**

| Objectives | 3 |
| --- | --- |
| Business Understanding | 3 |
| Data Understanding | 4 |
| Data Loading and Partial Cleansing MSSQL | 5 |
| SQL Complex Query 1 | 5 |
| SQL Complex Query 2 | 6 |
| SQL Complex Query 3 | 7 |
| Python ETL and Data Cleansing | 8 |
| Data Visualisation | 10-20 |
| References |  |

**Objectives**

The objective of CA1 is to help you gain a better understanding of the data science project workflow, including creating SQL databases for data storage, setting up data pipelines between SQL and Python, using Python for data analysis and building interactive dashboards for business users. Gobest Cab is a ride-hailing service company that hired us as their data science team to develop an easy to use, and intuitive machine learning software application that may help them to visualize and analyze cab data. On top of that, the company also wants us to build an interactive dashboard for data visualization, which would provide the company insights on drivers’ behaviors.

**Business Understanding**

Gobest Cab is a ride-hailing service company. Companies in Singapore similar to Gobest Cab include Grab, Uber, Gojek etc. These are ride-hailing or ridesharing companies that hire independent contractors as drivers.

The success of the company significantly hinges, if not entirely, upon the reputation of its contractors, namely the drivers. Customers tend to favor more reputable companies for safer and consistently pleasant ride experiences. Conversely, platforms with a history of risky or unsafe incidents involving drivers are likely to deter customers. Such occurrences can detrimentally impact a company's reputation, leading to negative consequences for its business.

Take for example, on 6 April 2022, 46 year old Micheal Chua was sentenced to nine weeks' jail and was banned from driving for three and a half years after he pleaded guilty to two counts of driving in a manner dangerous to the public.On two separate occasions, Chua injured 3 people, one of which ended up having a permanent injury. Because of this the reputation of one of the more reputable ridesharing companies, Grab, took a hit.



Michael Chua pleaded guilty to two counts of driving in a manner dangerous to the public on 6 April 2022

The phrase: “seeing the glass half empty,” describes this situation well. No matter how good of a reputation you build, one rotten apple is enough to sully it.

In conclusion, the success of a ride-hailing company heavily relies on its reputation to earn money. From a business viewpoint, this makes safety the biggest priority for Gobest Cab. With this, we proceed to get an understanding of our data, which will allow us to identify patterns and signs associated with the behaviors of dangerous drivers.

**Data Understanding**

Lets redefine each feature in the sensor dataset as it might be confusing for people

1. **Accuracy** - As we can see from the picture accuracy is a sort of area inferred from multiple satellites where if the area is smaller meaning a smaller value the sensor is more confident/accurate
2. **Bearing** - Bearing basically is the direction of the car relative to the north.

So if bearing 0 = north 90 = east, 180 = south, 270 = west

1. **Acceleration x, y, z** means how fast the car accelerates in each plane

**x** = side

**y** = front

**z** = up and down. (in case you are confused z can mean going uphill/downhill)

Just like shown in the picture

1. **Gyro x,y,z** - this might be slightly more confusing

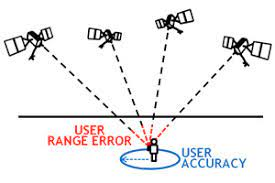
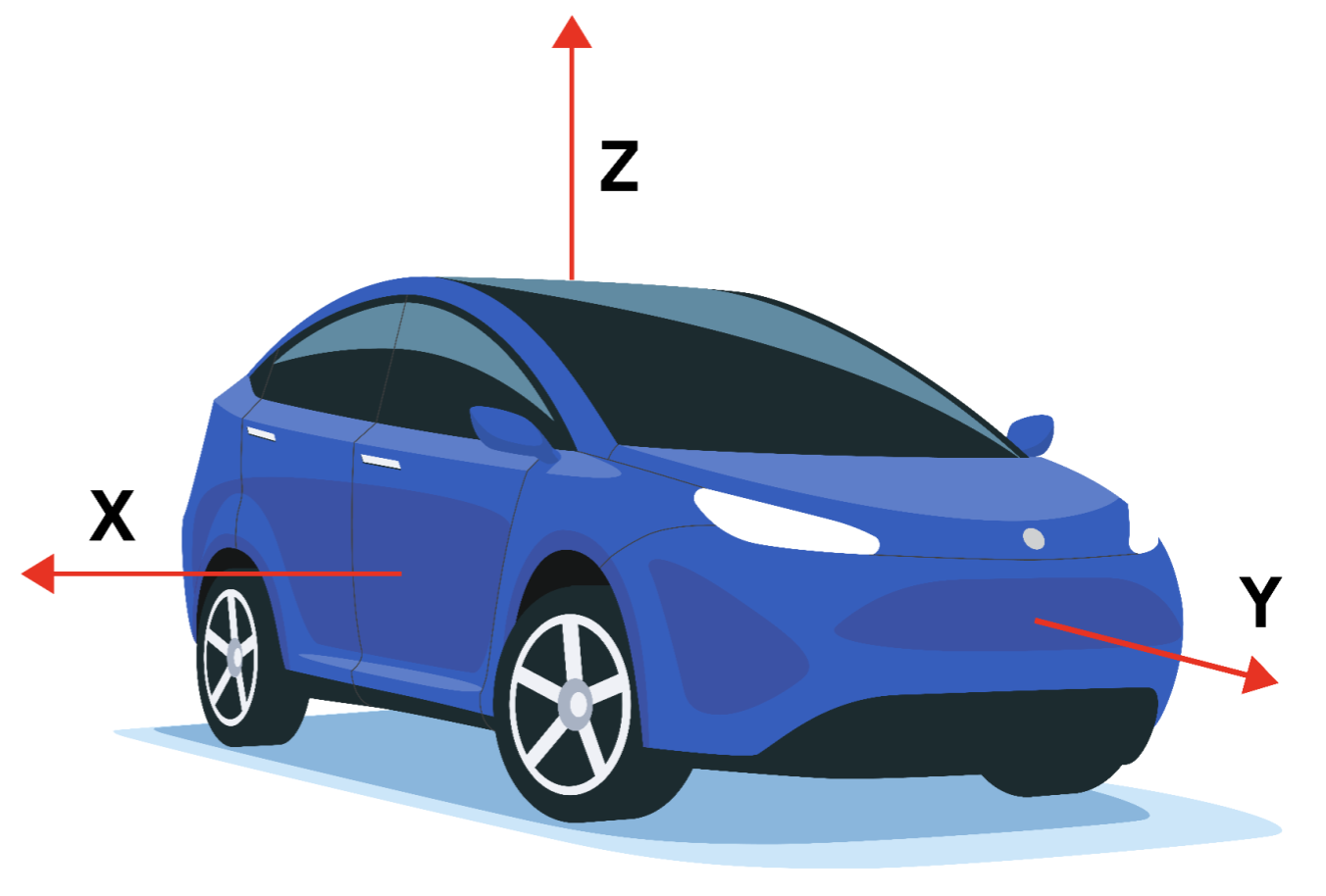
**gyro x** measures how much the car rolls/rock from side to side,

**gyro y** measures sudden acceleration/break

**gyro z** measures how much the car turn from the vertical axis

(so if you can’t tell a high value or fluctuating values in any of these 3 sensors should be avoided for every good car ride)

1. **Speed** - speed of the car measured by GPS
2. **Second** - second that the measurements were recorded

****

**Data Loading and Partial Cleansing MSSQL**

To perform queries we have to load the data into my SQL database, so first we used the normal method of loading each file into a table through import\_csv. After all the data are imported we then create 3 holding tables namely, TempDriver, TempSafety, TempSensor where we will keep the respective data, with the datatype of the datasets being all wrong (mainly all strings), we will need to configure our Temp database to access VARCHAR, once we insert the data we noticed that there are some empty values, but weirdly they are represented as ‘’ empty strings making them hard to work with. So I decided to change that, I set each of the data in each column to their correct datatype if value = ‘’ I set them to null for easier handling. When using cast to set the dtype it doesn’t necessarily change the dtype of the table only formatting the values to fit that dtype so we still need to alter the table to the correct dtype for each column Once that is done we can finally start doing some queries

**Complex Query 1**

| Query |  |
| --- | --- |
| Result |  |
| Insight | Complex Query 1, ranks all the car brands by percentage of Dangerous trips. Together with the car brand’s average driver rating, average years of experience and average car age. With Lexus having the highest dangerous percentage, we would expect a linear correlation between the variables, rating, experience, vehicle age and dangerous rides. However there does not seem to have any correlation. From our query, we can conclude that the car brand does heavily influence the occurrence of dangerous rides. Thus we should explore other factors which contribute more to the safety of the ride, such as looking at the sensor data and driving habits of drivers. |

**Complex Query 2**

| Query |  |
| --- | --- |
| Result |  |
| Insight | Complex Query 2 bands each driver into 1 of 4 groups, experience < 10, <15, < 20, and 20+ and calculates their safety percentages with a common perception that if someone has more experience they should be relatively safer however the data shows otherwise, with each age group having the same amount of dangerous rides why is that. I think we are forgetting that even if someone has little experience in being a driver for Gobest Cab, they have plenty of experience driving a car. So we can infer that amount of experience someone has does not directly correlate with how safe they are as a driver, instead we should look at other factors |

**Complex Query 3**

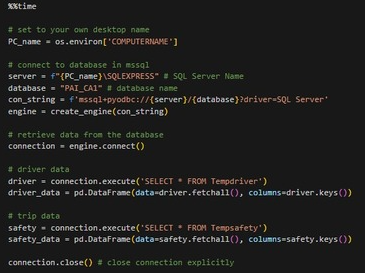
| Query |  |
| --- | --- |
| Result |  |
| Insight | Complex Query 3, ranks all 500 drivers in Gobest Cab by the danger percentage of their rides (Output above shows only 35/500 drivers). Danger percentage is calculated by taking the number of dangerous rides (of a driver) and dividing it by the total number of rides they have given. We see that over 11 drivers have danger percentages of over 40% with one driver even being in the 50th percentile. By knowing the danger percentages of every driver, the company can consult stakeholders in the business on the maximum percentage they would tolerate, and implement measures such as revoking dangerous driver’s partnership with Gobest Cab. |

**Python ETL and Data Cleansing**

**Extract, Load, Transform Pipeline**

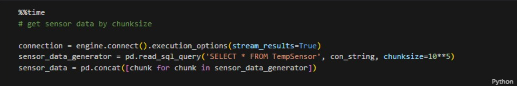
The partially cleaned sensor, driver and safety data is stored inside a Microsoft SQL Server (MSSQL). This data will now be extracted into a Python Jupyter Notebook where the final data cleansing and transformation will be carried out before being exported to be used for visualizations.

To extract data from MSSQL, we utilized the SQLAlchemy library, and loaded the data into Pandas Dataframes using the Pandas library.



Using SQLAlchemy to connect to MSSQL and load the safety and driver data

We loaded the safety and driver data first as there wasn’t a lot of data as compared to the sensor data.



For the sensor data, we decided to load it in chunks of 10,000 making the process more efficient instead of loading it all at once.

After that we dropped duplicated rows in each pandas dataframe (sensor, safety and drivers). We then proceeded to merge the data frames to produce a singular data frame that is to be sent for one last round of final cleansing.

**Python Data Cleansing**

Once we load our semi-cleaned dataset into Python, we need to perform the remaining data preprocessing so that the data can be used in future steps.

After that is done we can finally start the main part, dealing with missing values, **for the missing values in seconds, we immediately dropped** them as we decided that a row with a missing ‘second’ has lost its identity and can’t be possibly replaced or interpolated in any way.

After that has been done we need to decide what to do with the rest of the null values in each column, namely being (accuracy, bearing, acceleration(x,y,z), gyro(x,y,z), and speed for these values dropping them is definitely not the correct thing to do it might be the easiest method but we will lose a lot of data through that method, we also cannot replace all missing values with the average for that bookingID as it might not make the most sense when the car starts off with a speed of 200 eg. So what do we do. **Interpolation,** I decided to interpolate the missing data with an in-between value of the second before it and the second after it. This way we can keep the context of the value while replacing the null values

So i first sorted the data by bookingID, in each bookingID, it will sort by seconds this is so I can see the data in the order that they are recorded in. Now that that is done I used the .interpolate to interpolate each value, there are still some null values left as the first and last records can’t use 2 data points to interpolate so I just set those to be 0 as I assumed the ride either just started or just ended

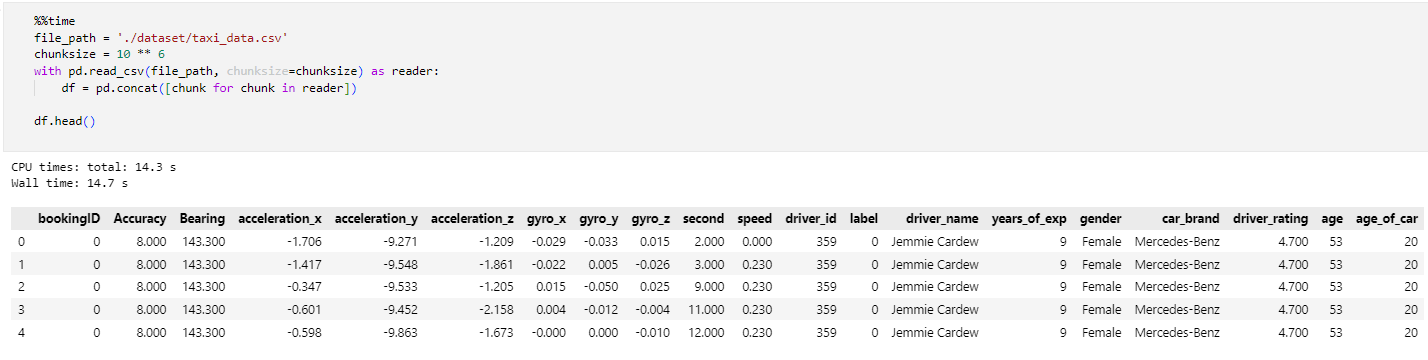
Lastly, I changed the date of birth to an age column by calculating the age based on the date that I ran that code. Along with changing the car model to age of car through the same method. Finally i switched the bookingID to integer value and that's the end of preprocessing, after saving the data to a csv file we can move on to the next step.

**Data Visualisation**

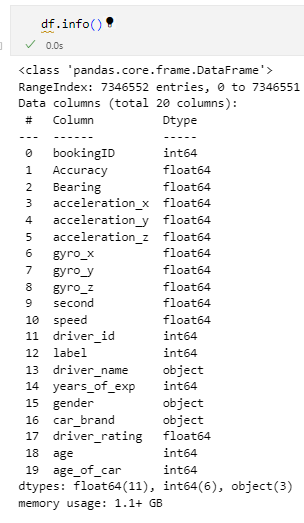
**Python Exploratory Data Analysis (EDA)**

After cleaning, we will be performing Exploratory Data Analysis on our data to gain a greater understanding of our data in relation to the safety of the rides.

First, we want to chunk load our dataset using **Pandas**. We will be using both **Dask** and **Pandas** to load our dataset. As Dask is suited for large amounts of data due to it leveraging on machines by using their multi-core CPUs and its lazy computation where it does not compute the result unless explicitly requested.

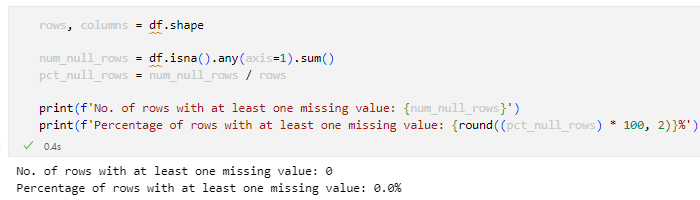


After loading our data, we perform a mini data analysis to find our the shape, and info of our dataset.

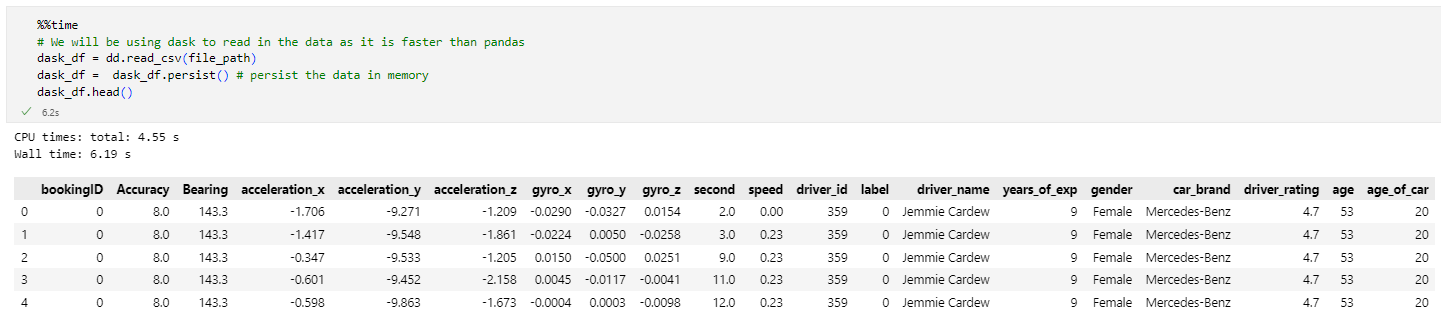
It is observed that we can reduce the dtypes of our columns to reduce the memory space require for our dataset.

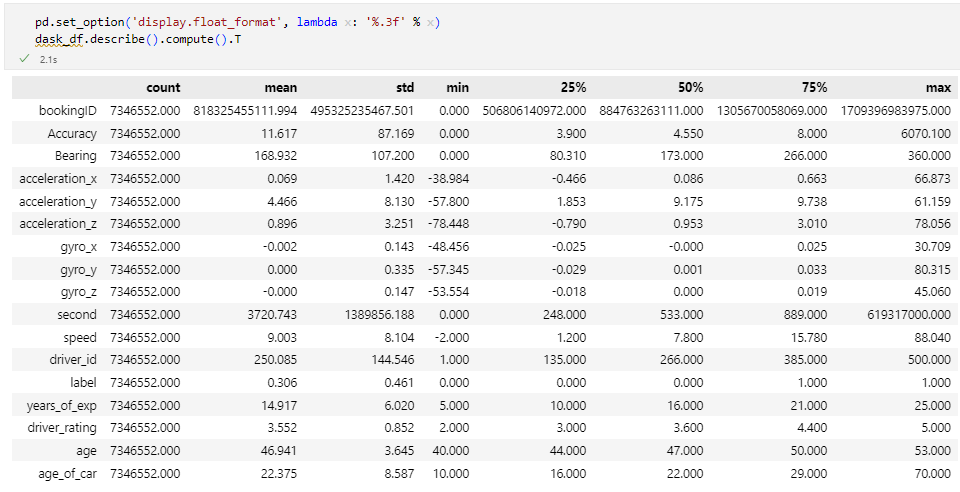
To prepare our dataset for EDA, we need to double check any NULL or missing values that have slipped through during data cleaning.



There are no missing values in our dataset, and we can proceed with EDA!

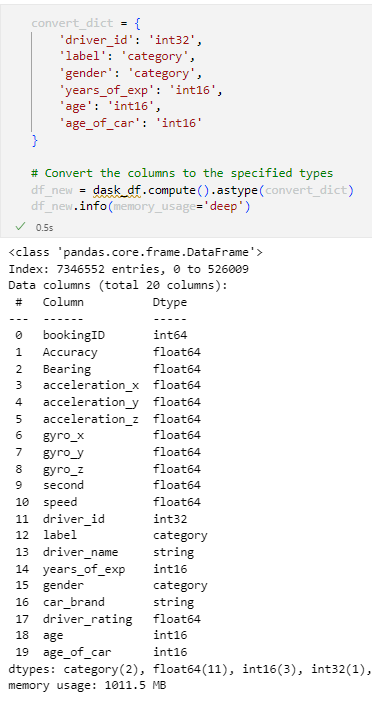
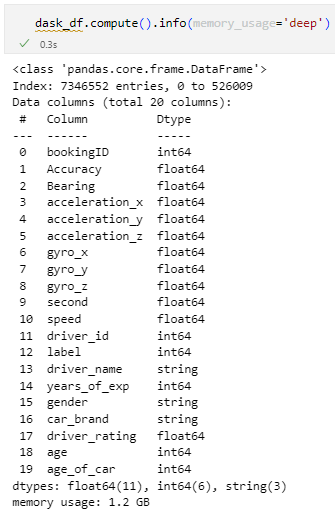
We convert our Pandas Dataframe into a Dask Dataframe for more efficient computing of our huge dataset





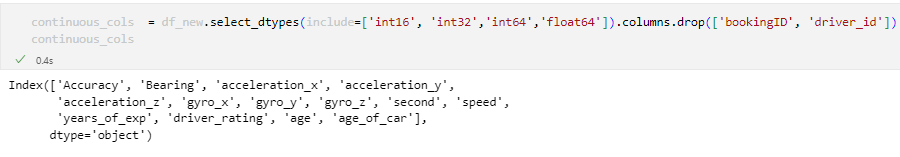
Summary statistics for our dataset

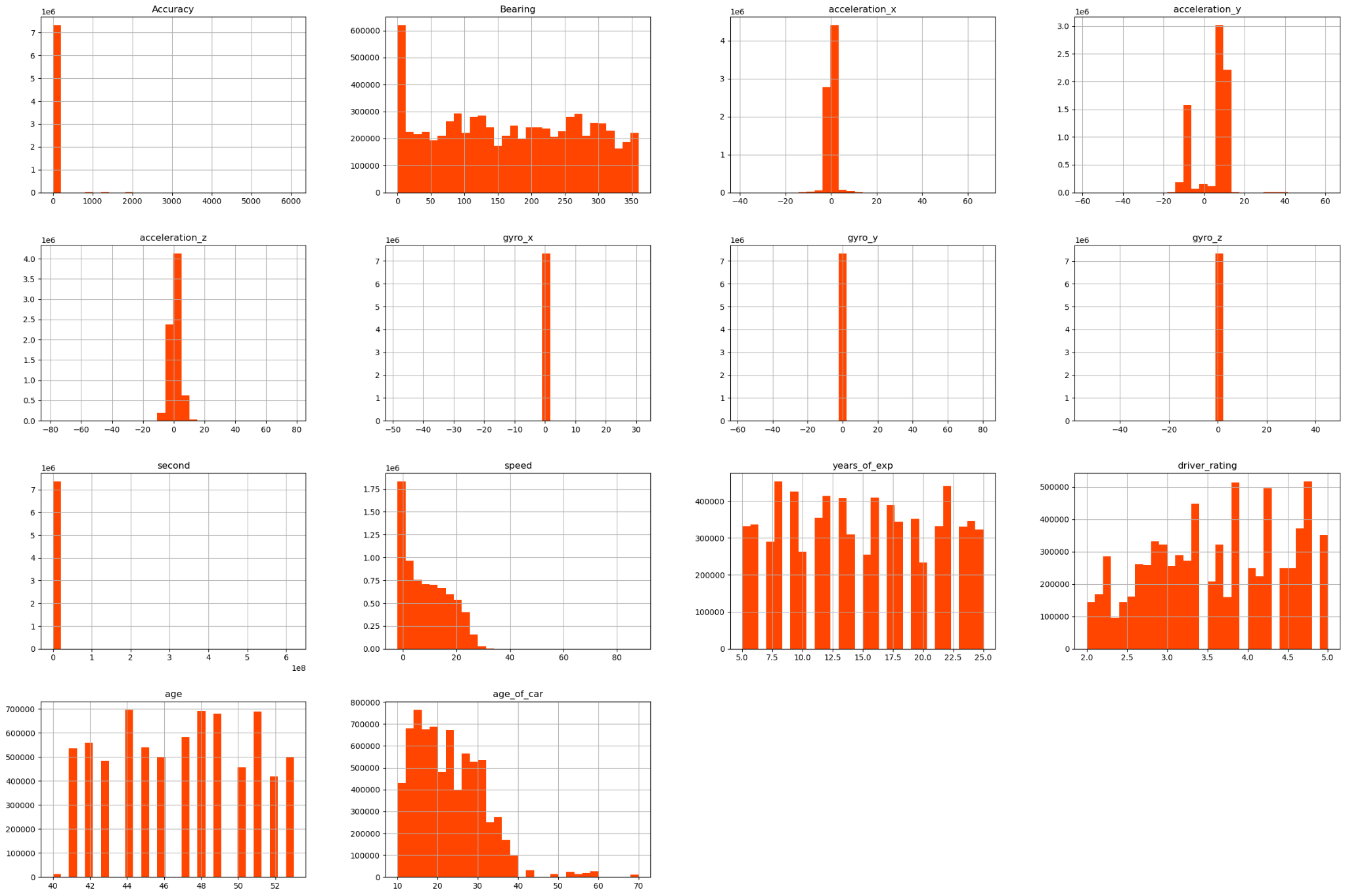
We will be reducing the memory load of our dask dataframe for better performance.



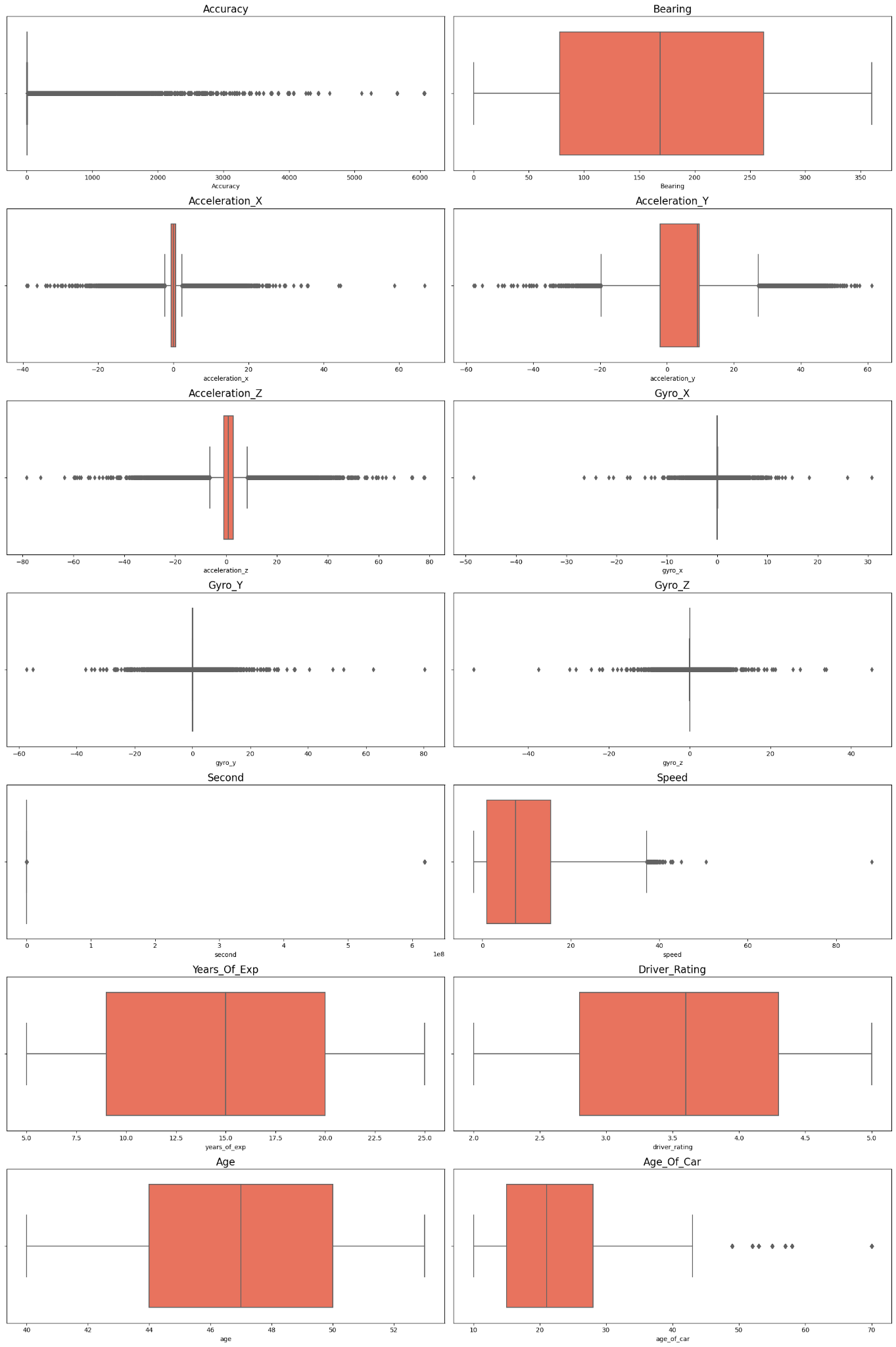
After reducing and changing the dtypes of our columns, we successfully reduced our memory load by 200MB.

We will be checking our outliers

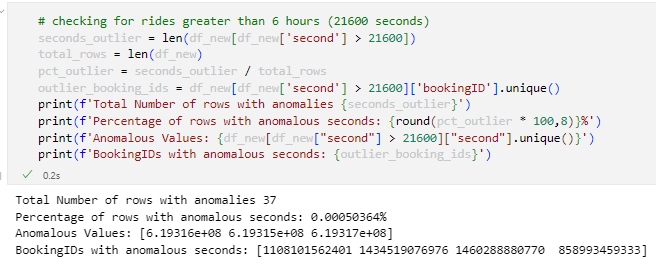




From the histograms, we can see that our data especially the sensor data is heavily skewed, where the vast majority of data points lie around 1 std. We can see it more clearly with a boxplot.



With accuracy being positively skewed and seconds having a clear anomaly in the data. We need to dig deeper into this anomaly.



37 of our rows have seconds greater than 21600 (6 hours), this seem impossible as taxi rides do not last for 173,032 Hrs long. That is almost **18 years** of constant driving. This anomaly may be attributed to errors within the sensor of the taxi as there are multiple of the similar value anomalies in different bookingIDs.

We will now be doing 2 Statistical tests to see the relations between safety and our columns.

**Point Biserial Correlation** is used to determine the correlation between our label and numerical columns:



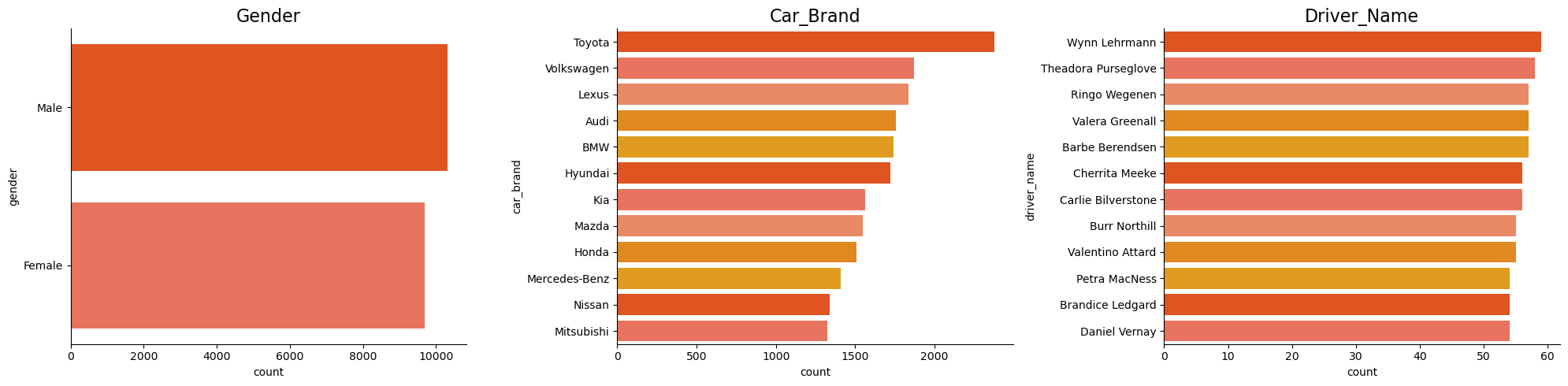
All our continuous variables individually have weak to no linear relationships with the safety of the ride.

**Chi-squared Test** is used to test for the correlation between labels and nominal variables (eg. car\_brand)



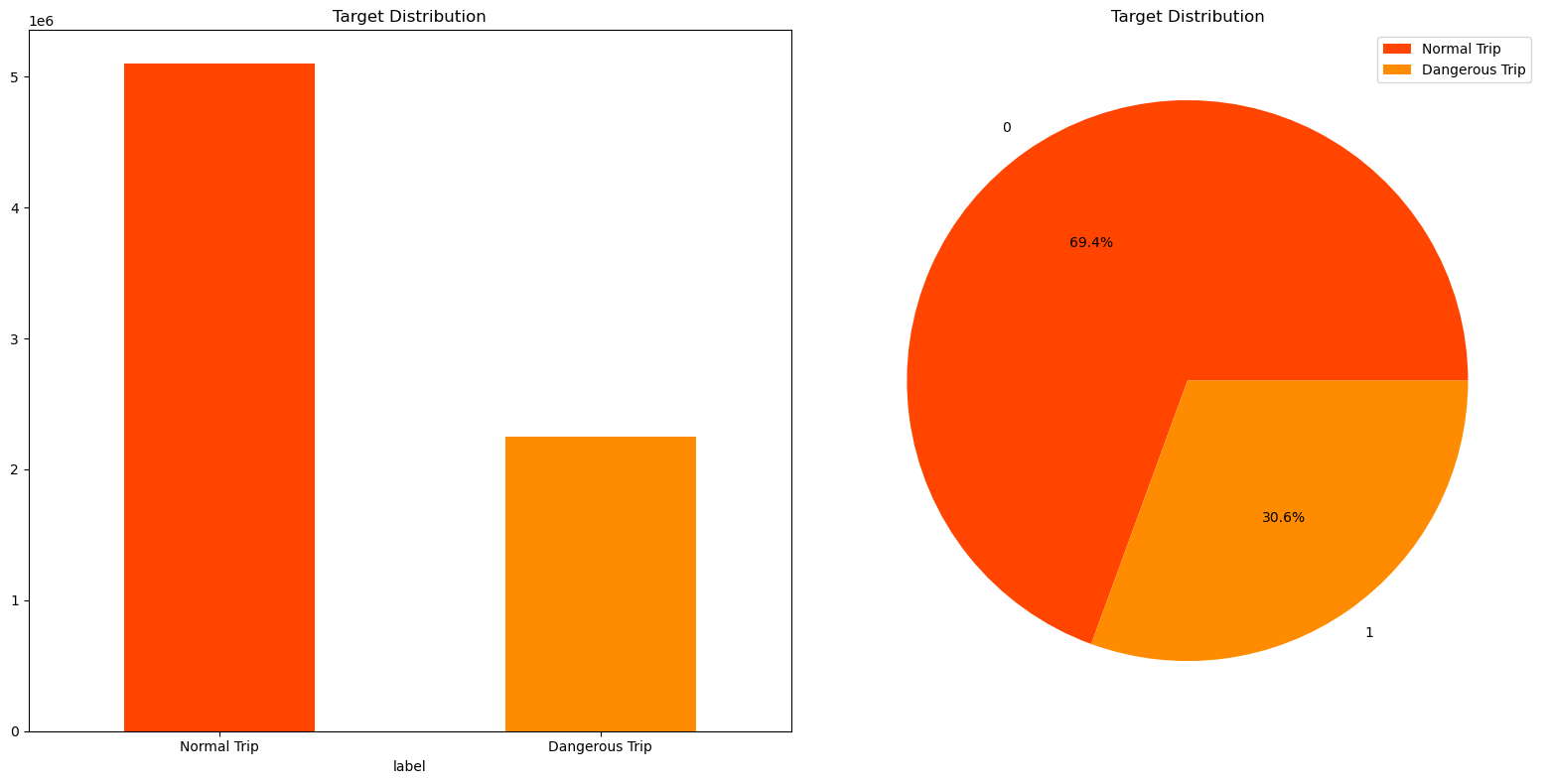
We can confirm that there is no correlation between car brands and the safety of a ride.

**Univariate Analysis**

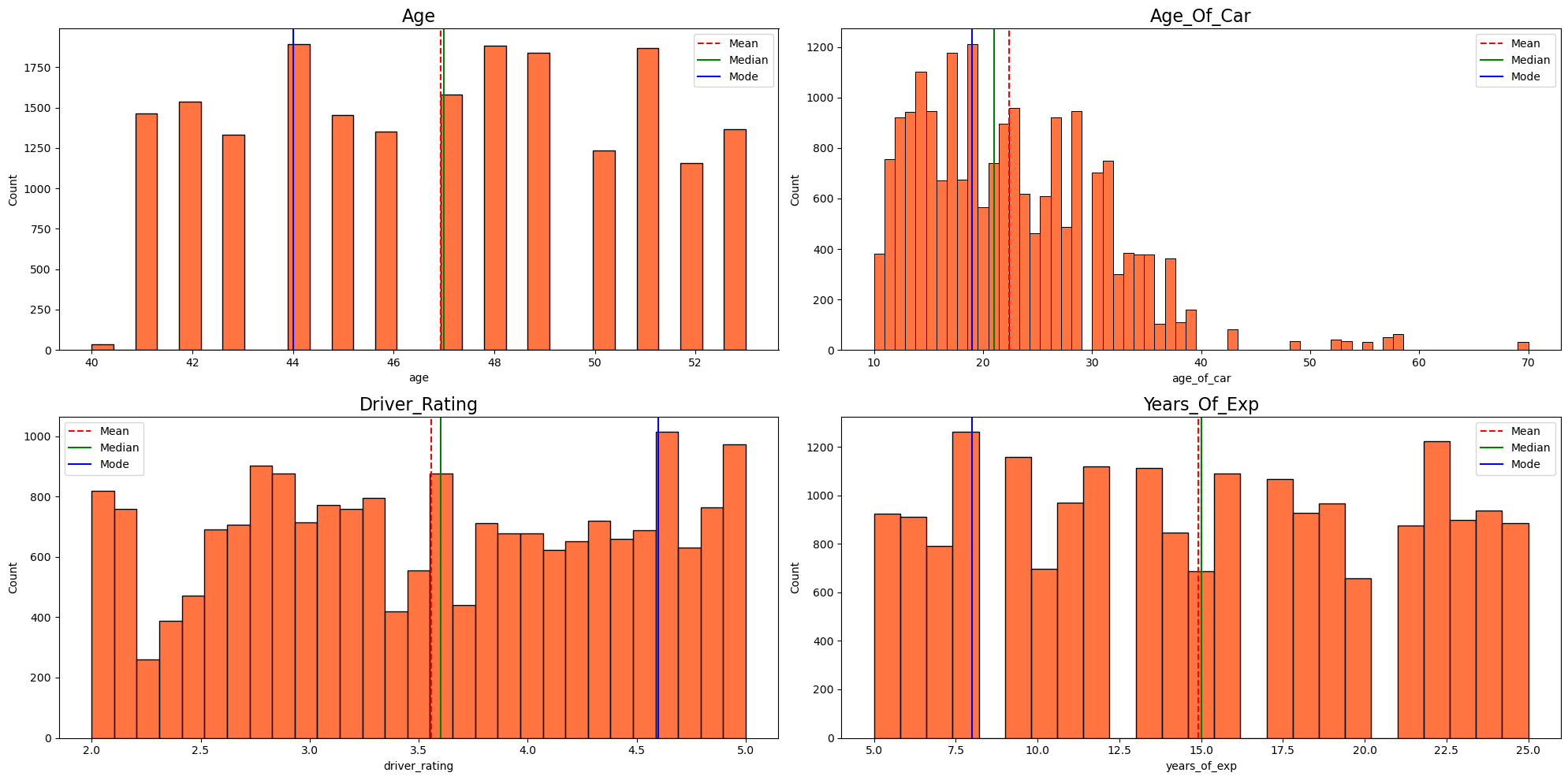


* There are slightly more male drivers than female drivers
* Drivers use Toyotas the most as their vehicle of choice
* Driver Wynn Lehrmann has the most amount of unique rides, making him the best driver in the company

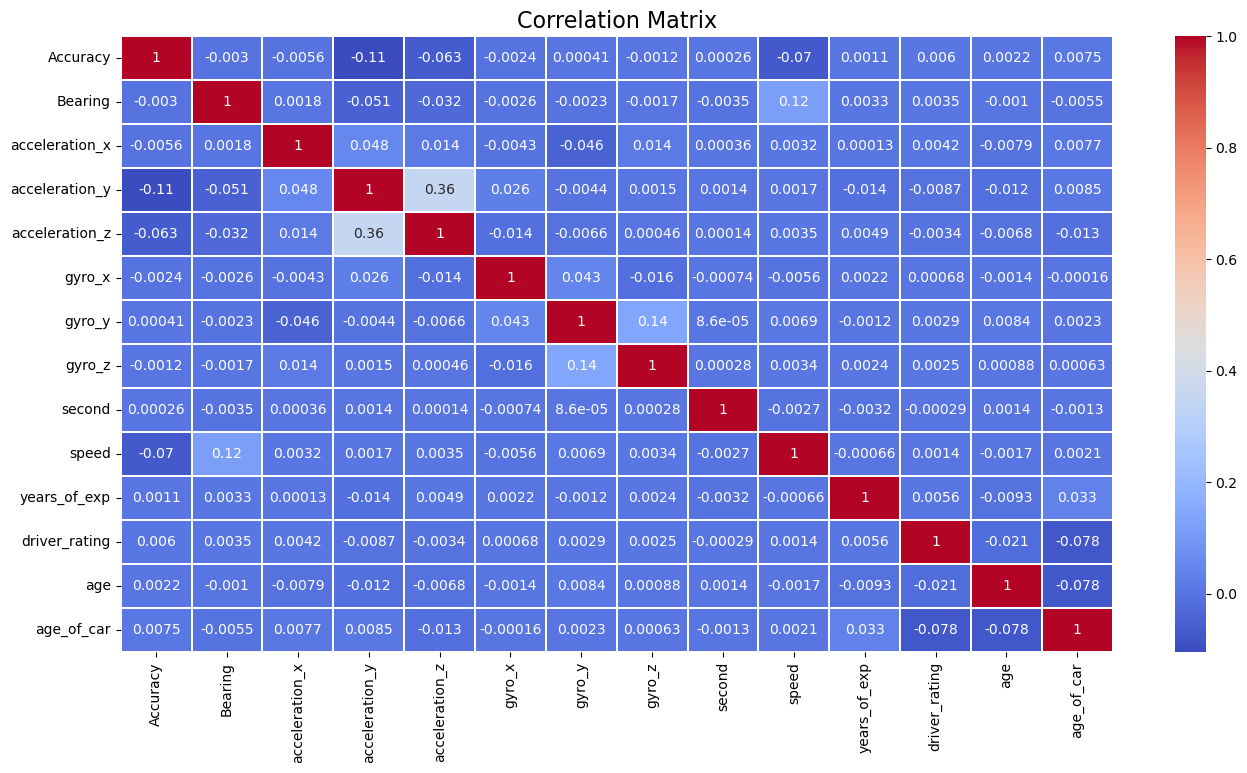
**Target Label**



* 30.6% of total trips taken are labeled as Dangerous, showing that drivers are generally safe drivers.



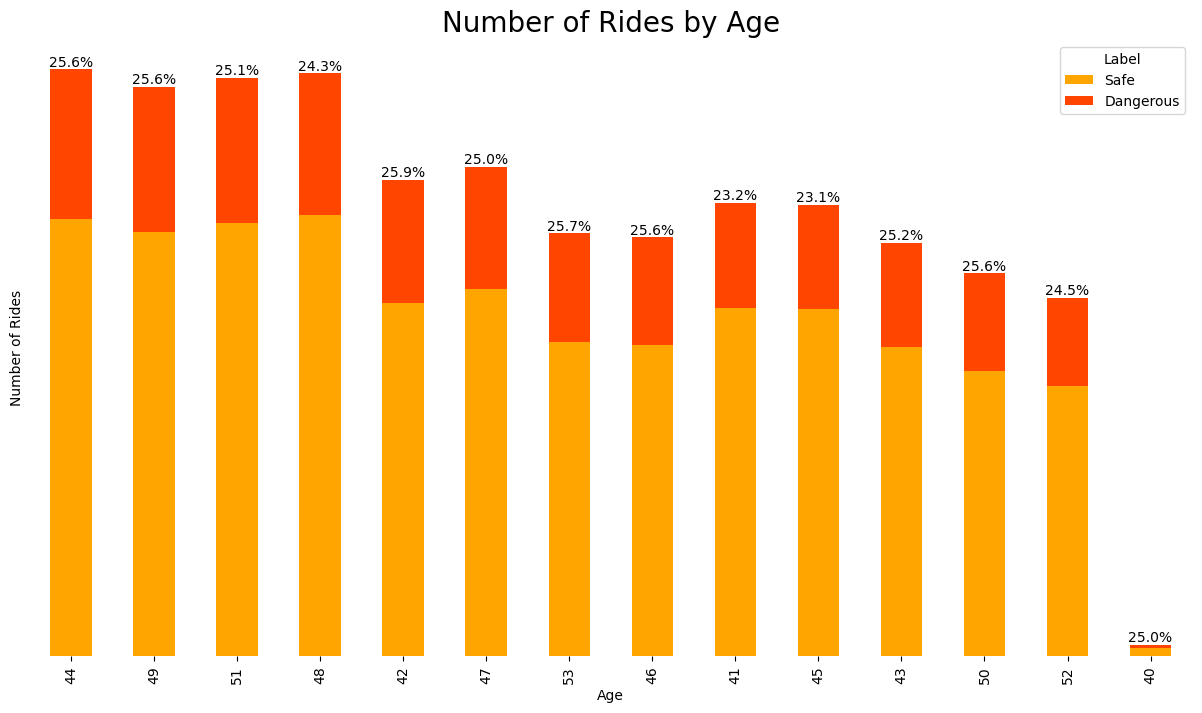
* There is an even distribution of driver’s age, rating and years of experience. With the mean and median being in the middle.
* Most cars used in the company are quite outdated, with the majority of them being 20+ years old. With a few outliers in the 50s and 60s.



* There is a high correlation between accerlation\_z and acceleration\_y, Bearing and speed and gyro\_y and gyro\_z.

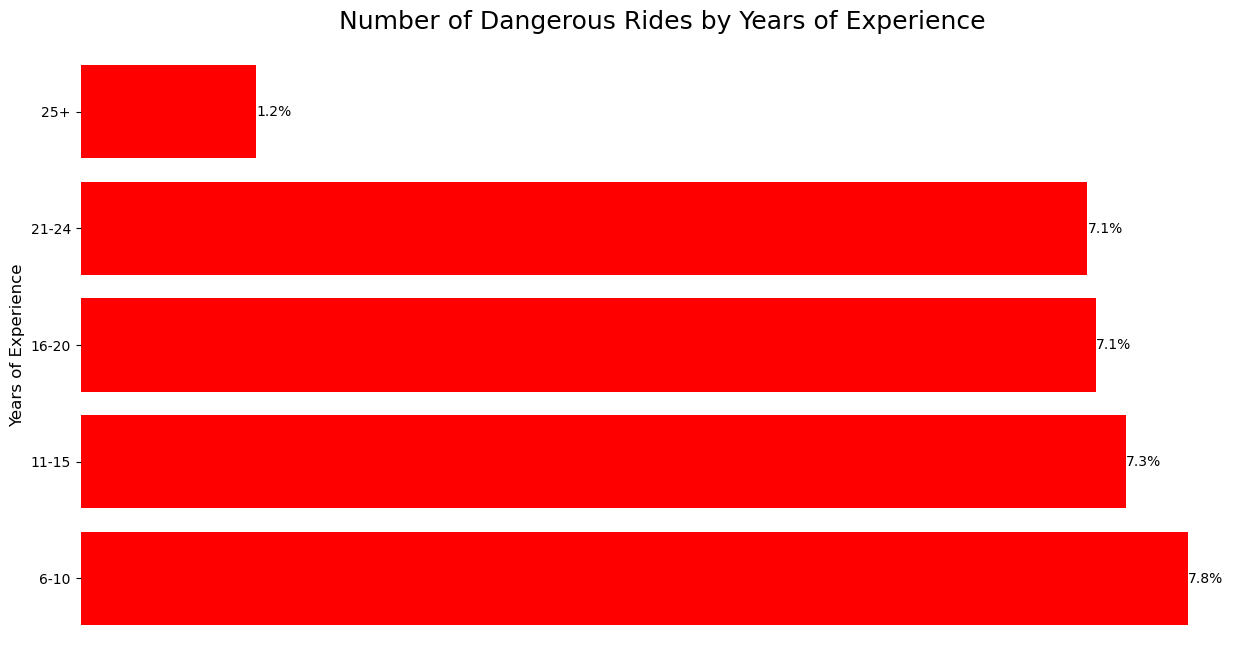
To have a deeper understanding on our data, we will be plotting some graphs to show the statistics on Dangerous rides within Gobest.

**Amount of Dangerous Rides by Age**

****

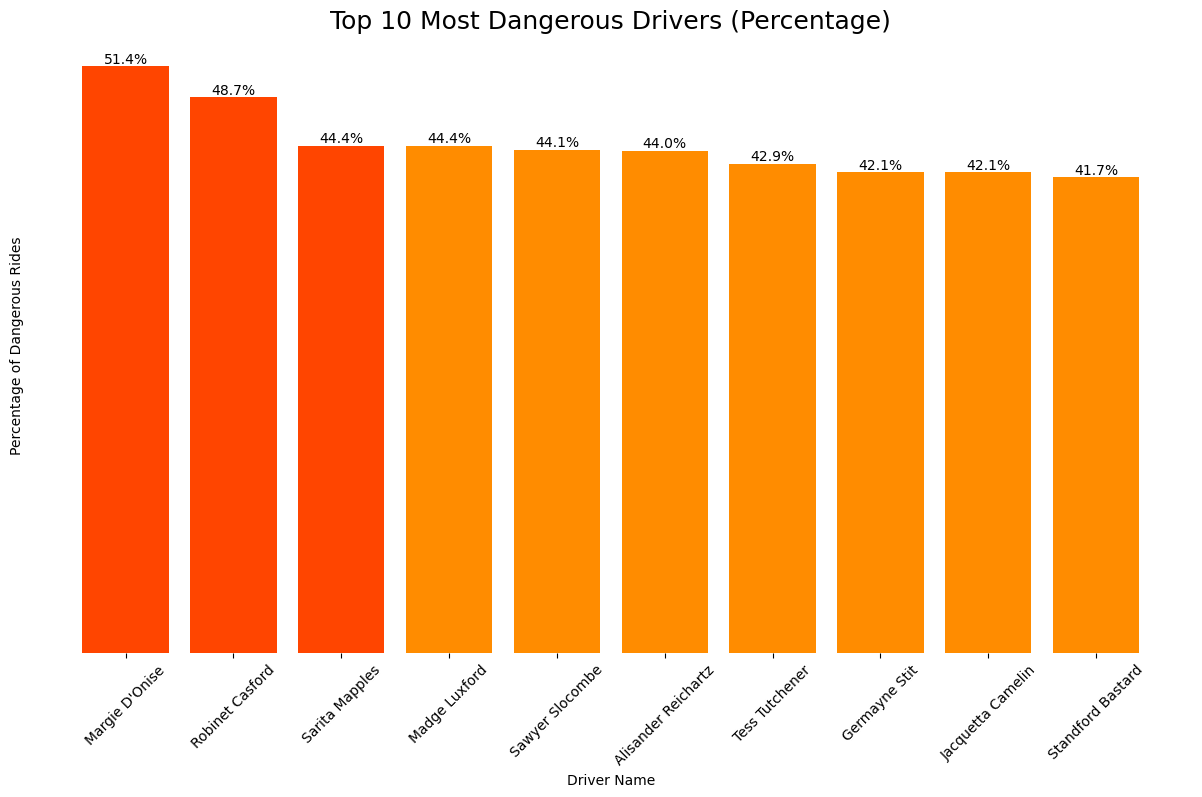
* There is no clear correlation between the age of the driver and the amount of dangerous rides
* Generally as most drivers are around the 40-50+ years old range, there is no 1 specific age where there is an increased dangerous ratio

**Dangerous Rides by Years of Experience**



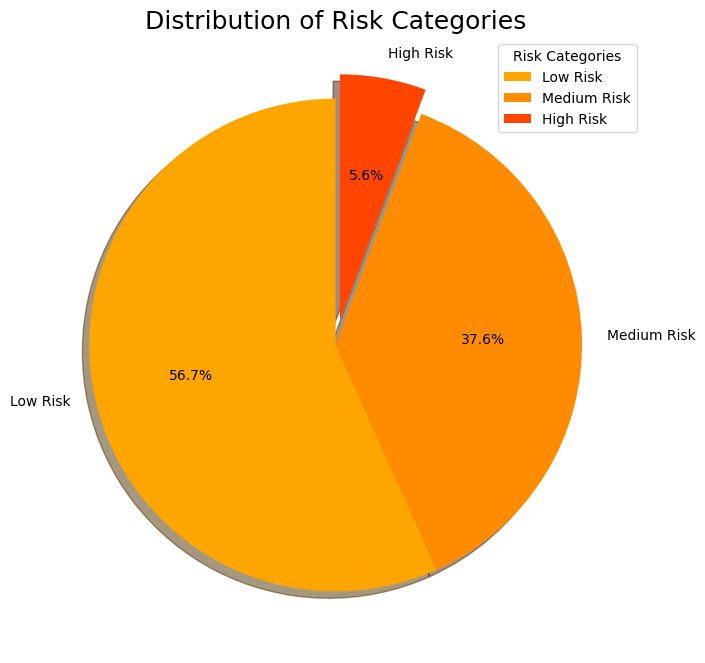
* Inexperienced drivers (6-10 years of experience) tend to have a higher amount of dangerous rides. This can be attributed to there being a higher amount of inexperienced drivers in the company.

**Most Dangerous Drivers in the company**



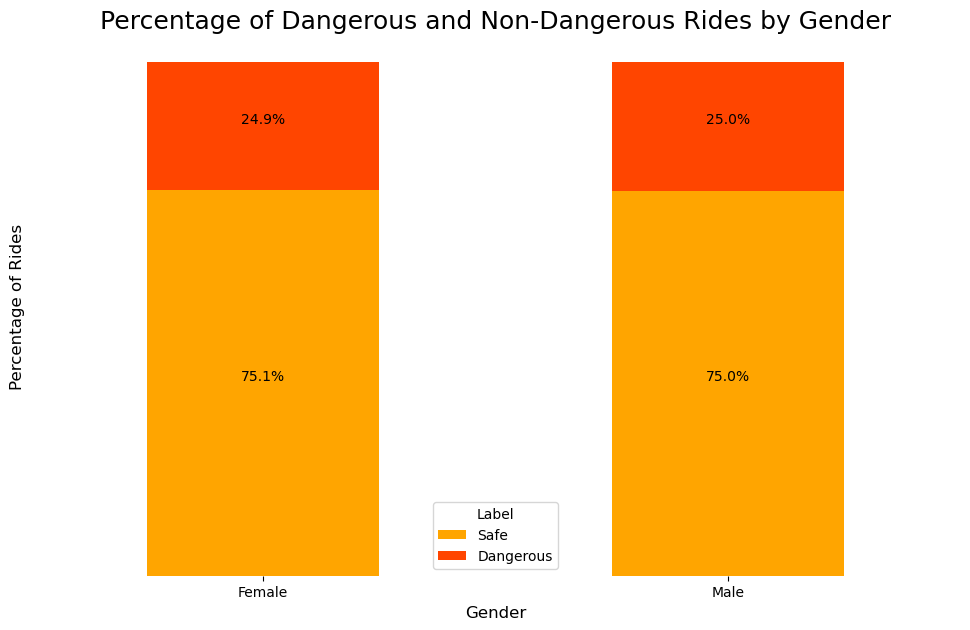
* Drivers Margie D’Onise, Robinet Casford and Sarita Mapples have the highest percentages of dangerous rides in the company

**Percentage of High Risks dangerous rides**



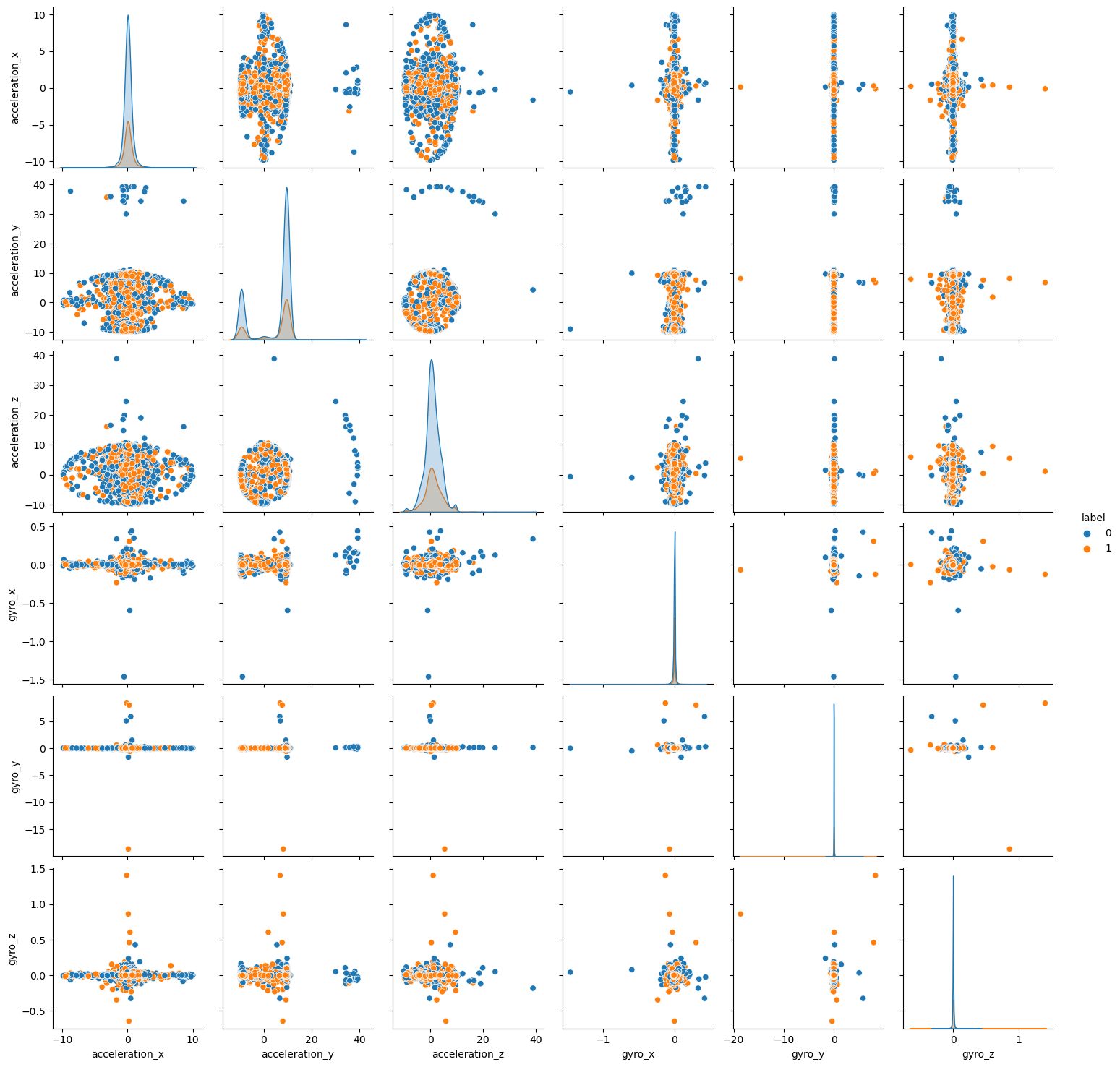
* We categorise High risk dangerous rides by having average speeds and acceleration being higher than 10m/s and 0.5m/s2. Medium risk have either of the conditions fulfilled.
* 5.6% of all dangerous rides are high risk

**Dangerous Rides by Gender**

****

* There is no difference between either gender when it comes to dangerous driving The percentages in respect to the total rides are almost the same.

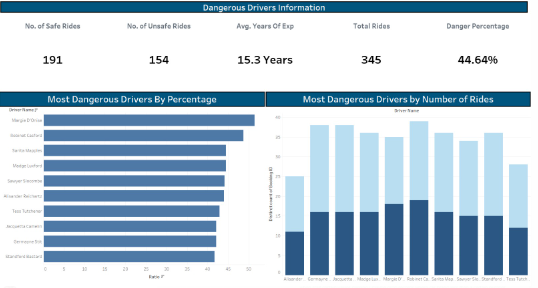
**Relation between Safety and average sensor data**

****

* Outliers for correlations between acceleration x,y,z and gyro x,y,z are usually labelled as dangerous rides. This can suggest that intense fluctuations in either acceleration or gyro during the ride can be a good indicator of a dangerous ride,

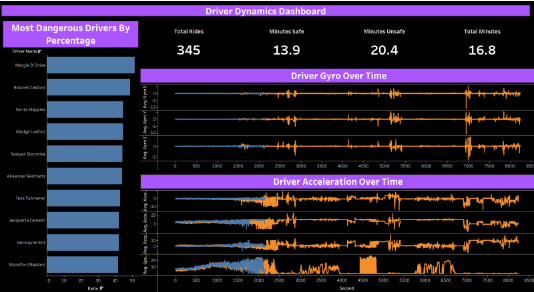
**Tableau Dashboard**

**Dashboard 1 - Identifying the Top 10 Dangerous Driver**

****

From the dashboard, we can view the details on the most dangerous drivers in more detail. The top 10 drivers have an average danger rate of 44.64% which is very high.. Aside from this, we see that the average years of experience is 15.3 years, which is relatively long. This means that more experienced drivers don't necessarily translate to safer rides.

**Dashboard 2 - Dangerous Drivers Dynamics**

****

Dashboard 2 is also related to the top 10 dangerous drivers. However, this time it displays the average acceleration and gyroscopic reading and trip time for safe and unsafe trips. We see that on average, longer trips tend to be flagged as more dangerous. Aside from that, the acceleration and gyroscopic readings of dangerous trips tend to be more unstable and fluctuate more often compared to safe trips. If we click on each individual driver, we see that the following observations are true. These observations are important as it may help Gobest cab identify other potentially dangerous drivers, and consequently handle them.

**Summary**

All in all, we were able to achieve our objective of data loading, cleansing, ETL and creating an interactive dashboard for business users to understand. In the next part of this project, we will adjust certain practices (such as implementing DASK instead of Pandas) related to what we have done in this part in order to carry out it more efficiently.

**References**

1. Wong Shiying. (2022) Grab driver gets 9 weeks' jail for dangerous driving in 2 incidents where 3 people were injured, The Straits Times. Available at: <https://www.straitstimes.com/singapore/courts-crime/grab-driver-gets-9-weeks-jail-for-dangerous-driving-in-2-incidents-where-3-people-were-injured>
2. *Sensor types*. (n.d.). Android Open Source Project. <https://source.android.com/docs/core/interaction/sensors/sensor-types>