**Dynamic Pricing Strategy using Python**

## Dynamic Pricing is an application of data science that involves adjusting the prices of a product or service based on various factors in real time. It is used by companies to optimize revenue by setting flexible prices that respond to market demand, demographics, customer behaviour and competitor prices. So, if you want to learn how to build a data-driven Dynamic Pricing Strategy, this article is for you. In this article, I will take you through building a Dynamic Pricing Strategy using Python.

## **What is Dynamic Pricing?**

Dynamic Pricing is an application of Data Science that involves adjusting product or service prices based on various factors in real time. It is employed by businesses to optimize their revenue and profitability by setting flexible prices that respond to market demand, customer behaviour, and competitor pricing.

Using data-driven insights and algorithms, businesses can dynamically modify prices to achieve the most favourable outcomes.

For example, consider a ride-sharing company operating in a metropolitan area. The company wants to optimize its pricing strategy to maximize revenue and improve customer satisfaction. The traditional pricing model used by the business is based on fixed rates per kilometre, which does not account for fluctuations in supply and demand.

By implementing a dynamic pricing strategy, the company can leverage data science techniques to analyze various factors such as historical trip data, real-time demand, traffic patterns, and events happening in the area.

Using Machine Learning algorithms, the company can analyze data and adjust its prices in real-time. When demand is high, such as during rush hours or major events, the algorithm can increase the cost of the rides to incentivize more drivers to be available and balance the supply and demand. Conversely, during periods of low demand, the algorithm can lower the prices to attract more customers.

**Dynamic Pricing Strategy: Overview**

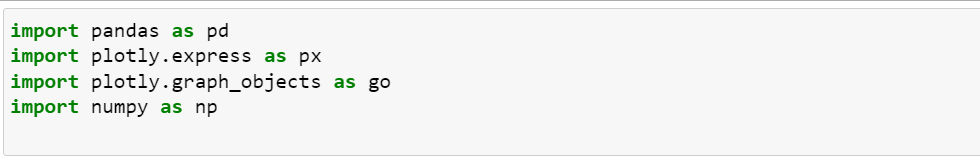
So, in a dynamic pricing strategy, the aim is to maximize revenue and profitability by pricing items at the right level that balances supply and demand dynamics. It allows businesses to adjust prices dynamically based on factors like time of day, day of the week, customer segments, inventory levels, seasonal fluctuations, competitor pricing, and market conditions.

To implement a data-driven dynamic pricing strategy, businesses typically require data that can provide insights into customer behaviour, market trends, and other influencing factors. So to create a dynamic pricing strategy, we need to have a dataset based on:

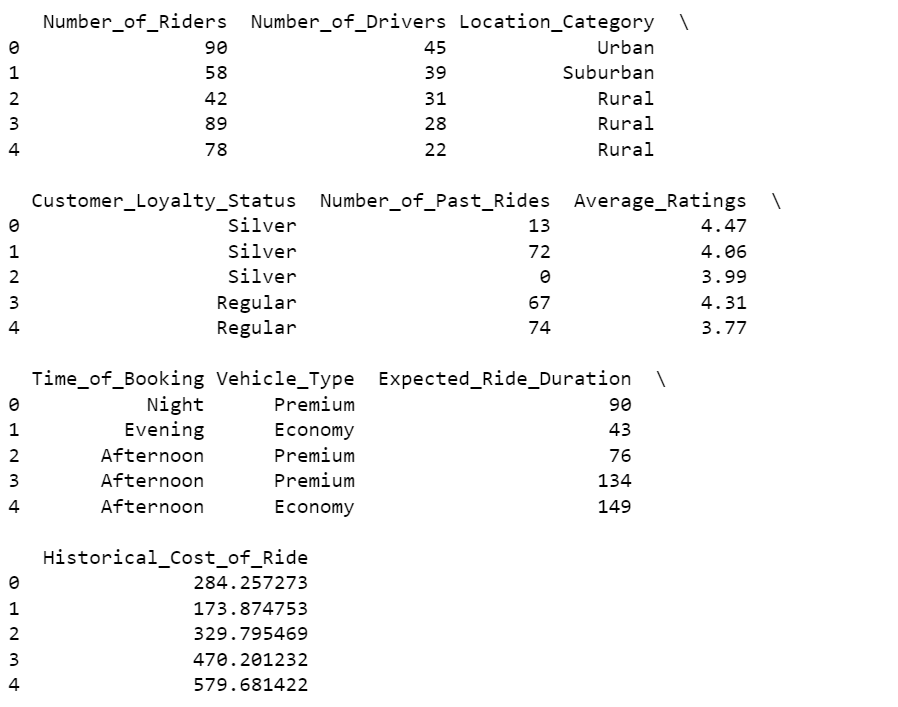
1. historical sales data
2. customer purchase patterns
3. market demand forecasts
4. cost data
5. customer segmentation data,
6. and real-time market data.

A dataset containing historical ride data has been taken. The dataset includes features such as the number of riders, number of drivers, location category, customer loyalty status, number of past rides, average ratings, time of booking, vehicle type, expected ride duration, and historical cost of the rides.

Let’s start the task of building a dynamic pricing strategy by importing the necessary Python libraries and the dataset**.**

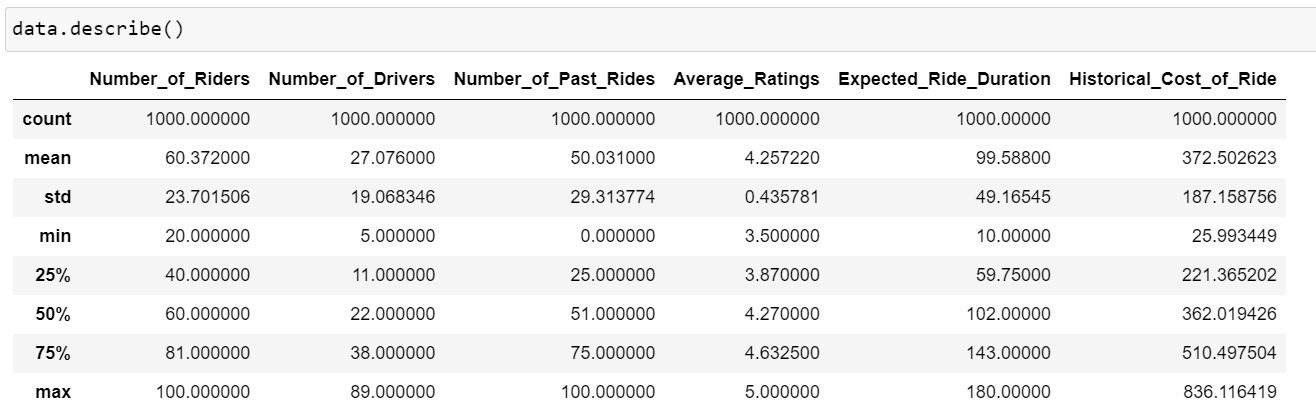
****

****

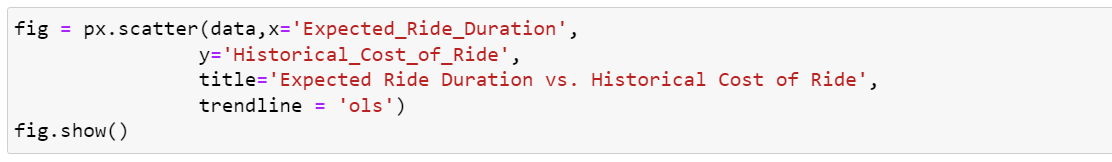
****

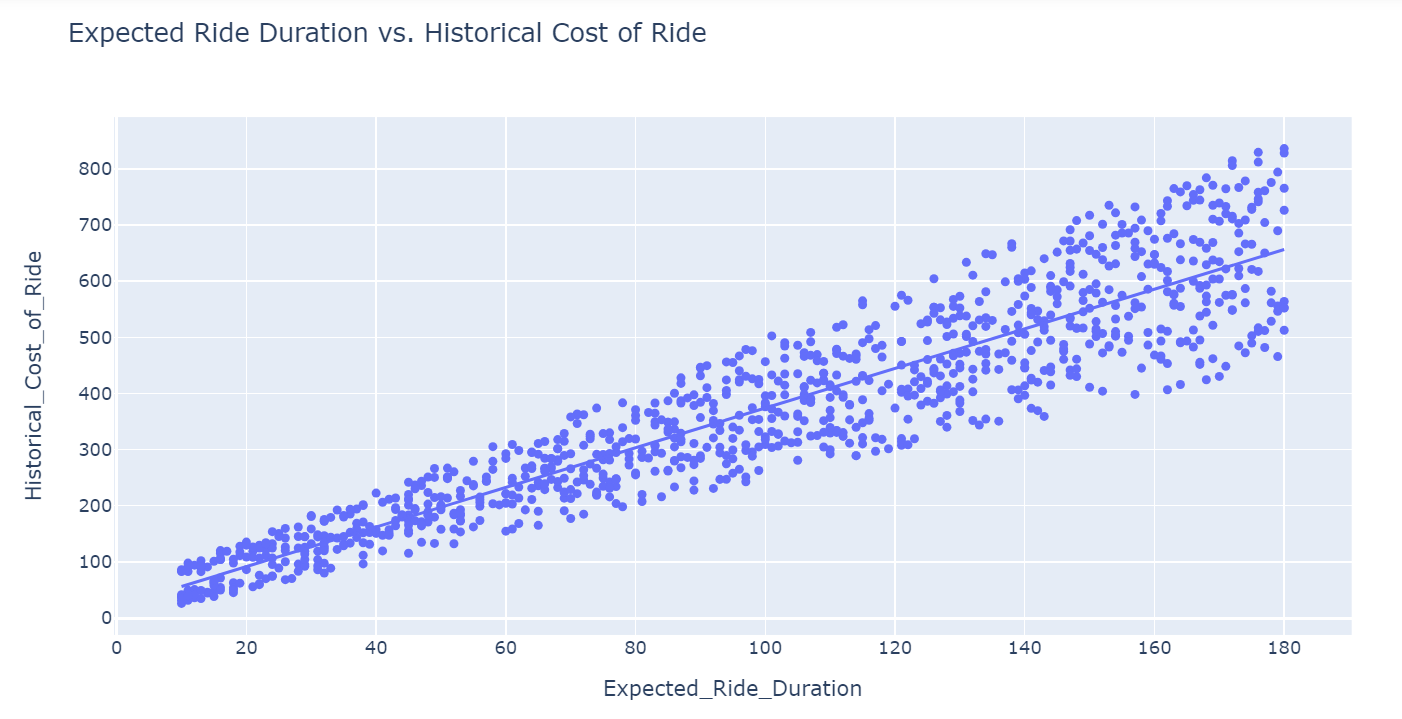
## Exploratory Data Analysis:

Looking into descriptive statistics of the data

****

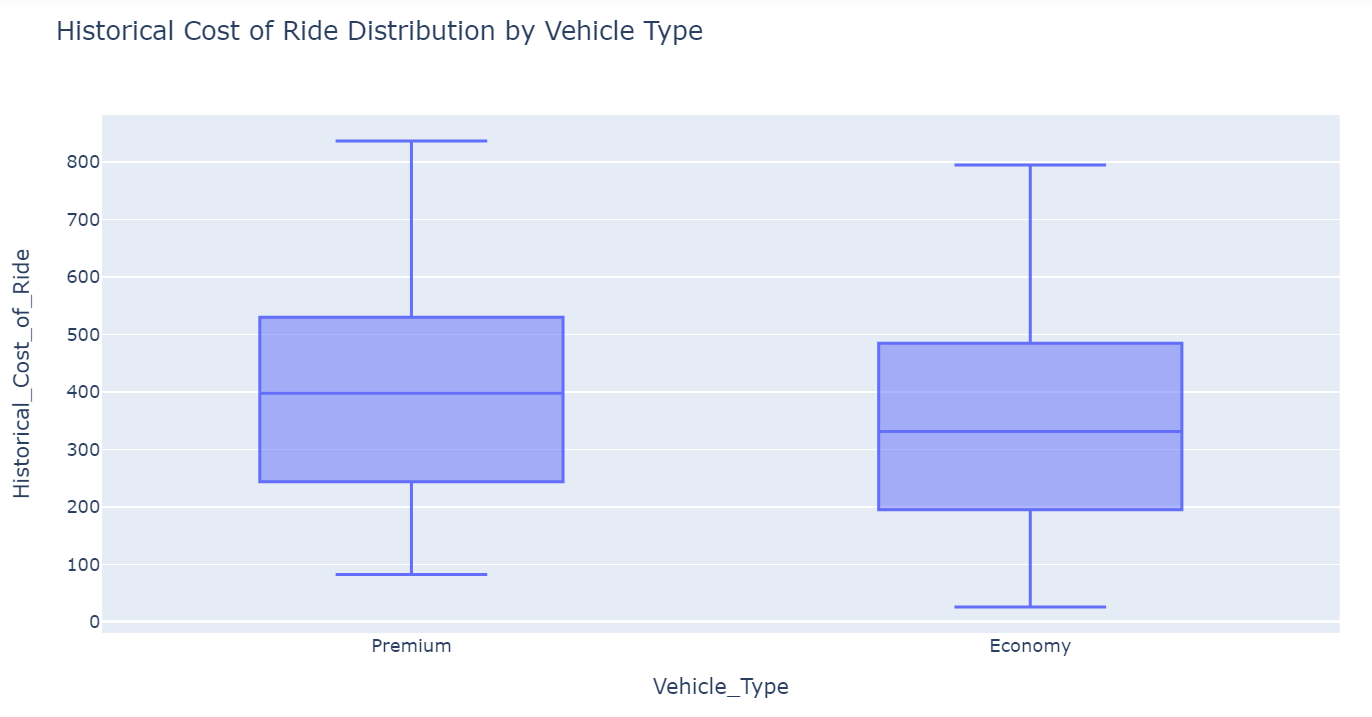
looking at the relationship between expected ride duration and the historical cost of the ride

****

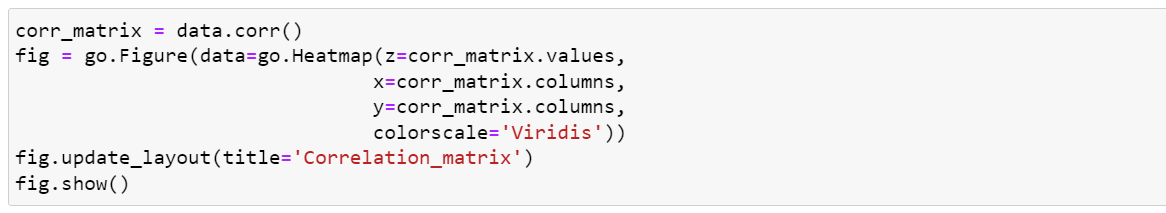
****

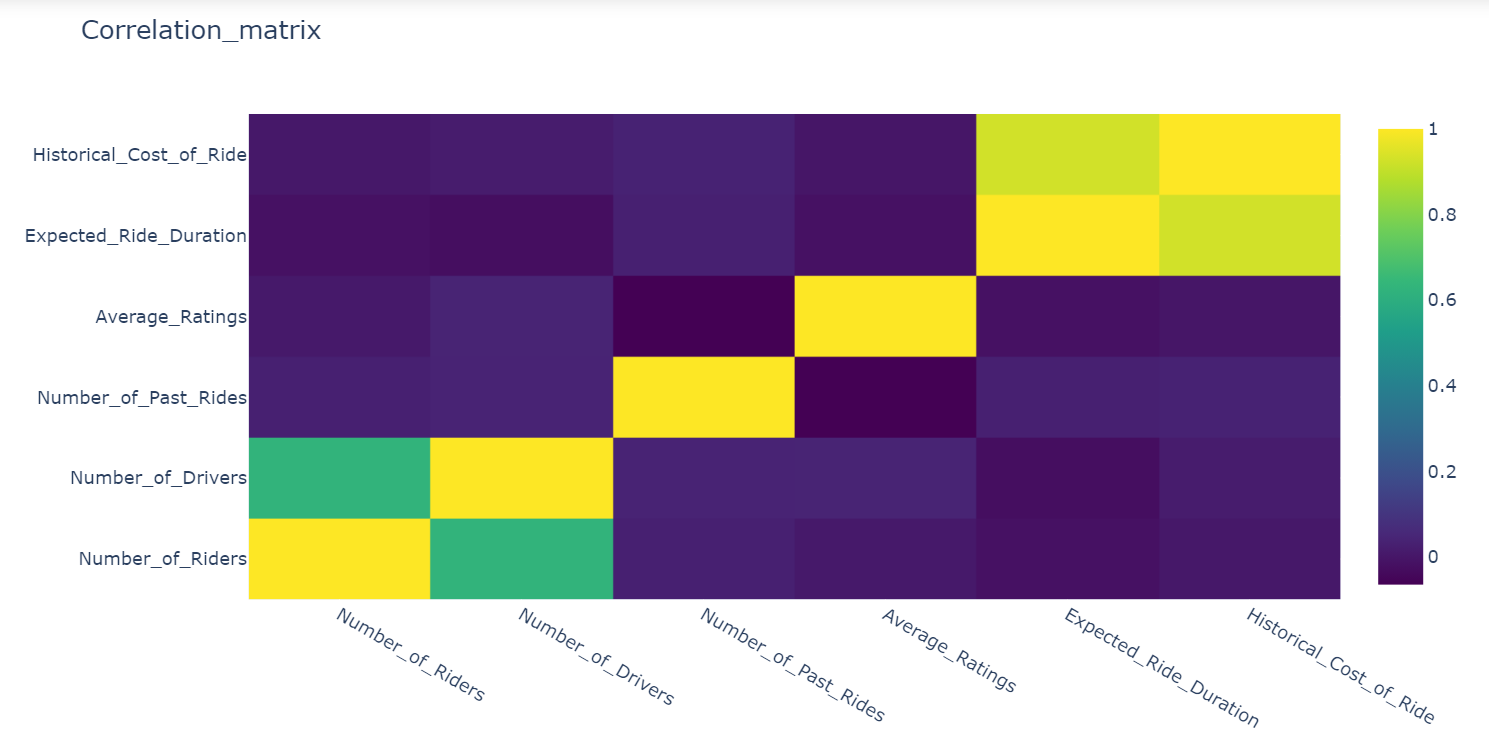
looking at the distribution of the historical cost of rides based on the vehicle type

****

****

Now let’s have a look at the correlation matrix

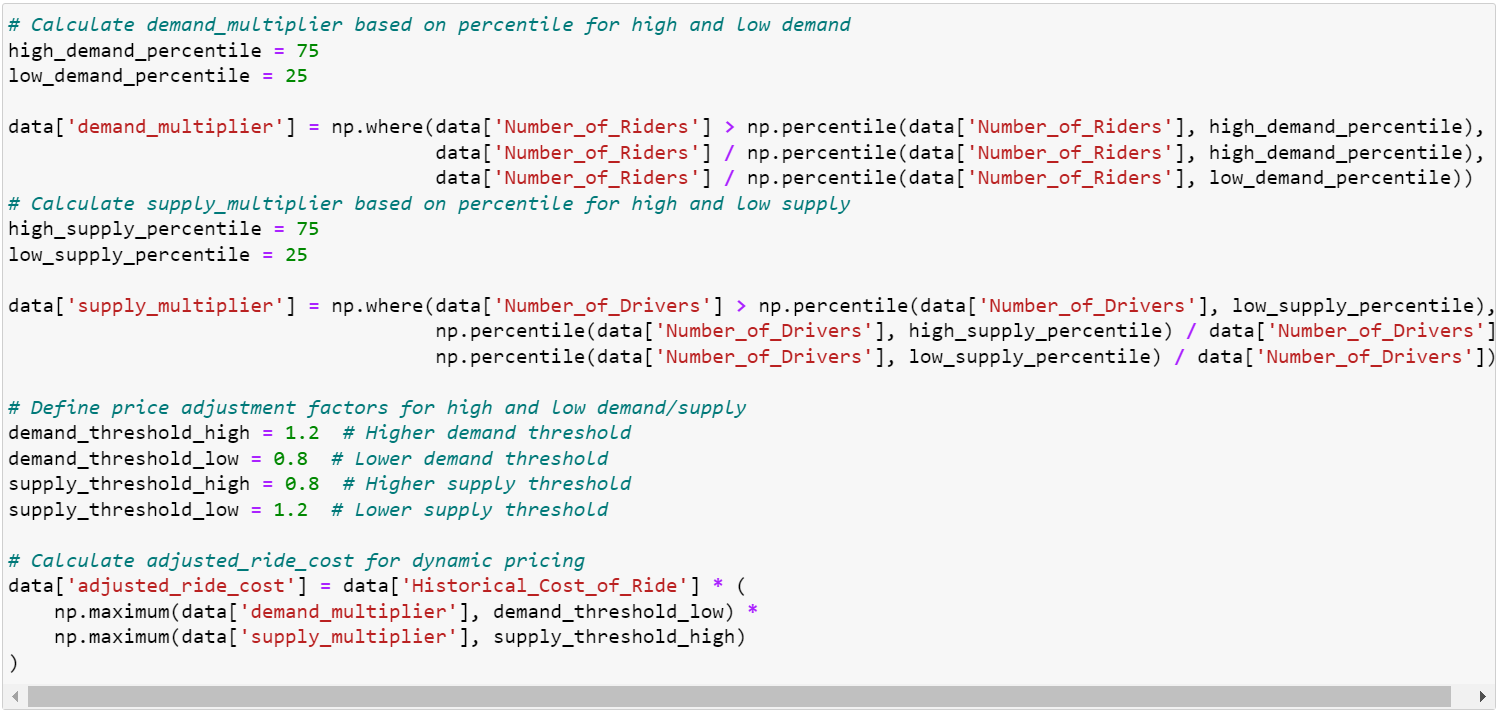
****

****

## Implementing a Dynamic Pricing Strategy:

The data provided by the company states that the company uses a pricing model that only takes the expected ride duration as a factor to determine the price for a ride. Now, we will implement a dynamic pricing strategy aiming to adjust the ride costs dynamically based on the demand and supply levels observed in the data. It will capture high-demand periods and low-supply scenarios to increase prices, while low-demand periods and high-supply situations will lead to price reductions.

Implementing this dynamic pricing strategy using Python:

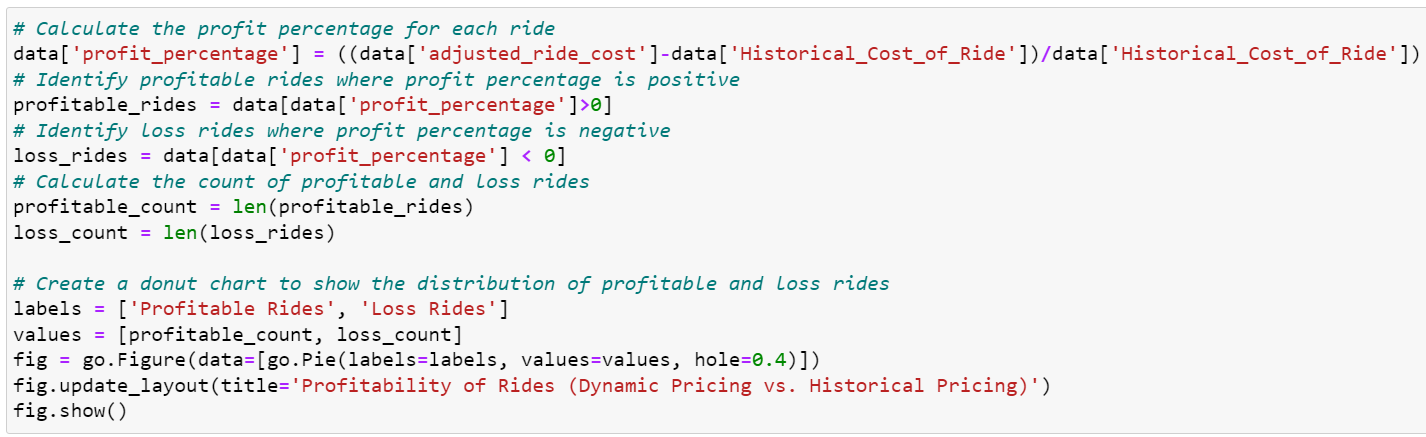
****

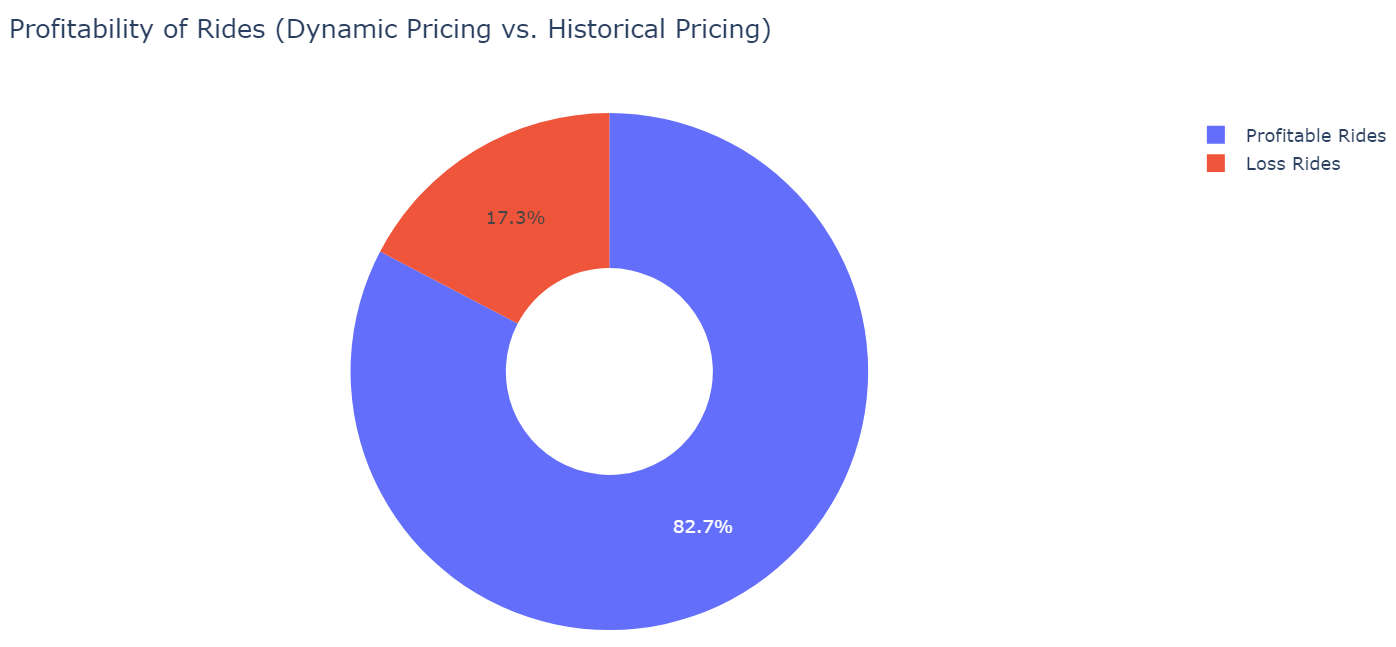
In the above code, first calculated the demand multiplier by comparing the number of riders to percentiles representing high and low demand levels. If the number of riders exceeds the percentile for high demand, the demand multiplier is set as the number of riders divided by the high-demand percentile. Otherwise, if the number of riders falls below the percentile for low demand, the demand multiplier is set as the number of riders divided by the low-demand percentile.

Next, calculated the supply multiplier by comparing the number of drivers to percentiles representing high and low supply levels. If the number of drivers exceeds the low-supply percentile, the supply multiplier is set as the high-supply percentile divided by the number of drivers. On the other hand, if the number of drivers is below the low-supply percentile, the supply multiplier is set as the low-supply percentile divided by the number of drivers.

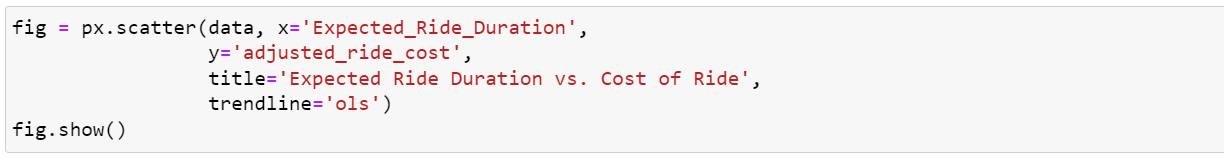
Finally, calculated the adjusted ride cost for dynamic pricing. It multiplies the historical cost of the ride by the maximum of the demand multiplier and a lower threshold (demand\_threshold\_low), and also by the maximum of the supply multiplier and an upper threshold (supply\_threshold\_high). This multiplication ensures that the adjusted ride cost captures the combined effect of demand and supply multipliers, with the thresholds serving as caps or floors to control the price adjustments.

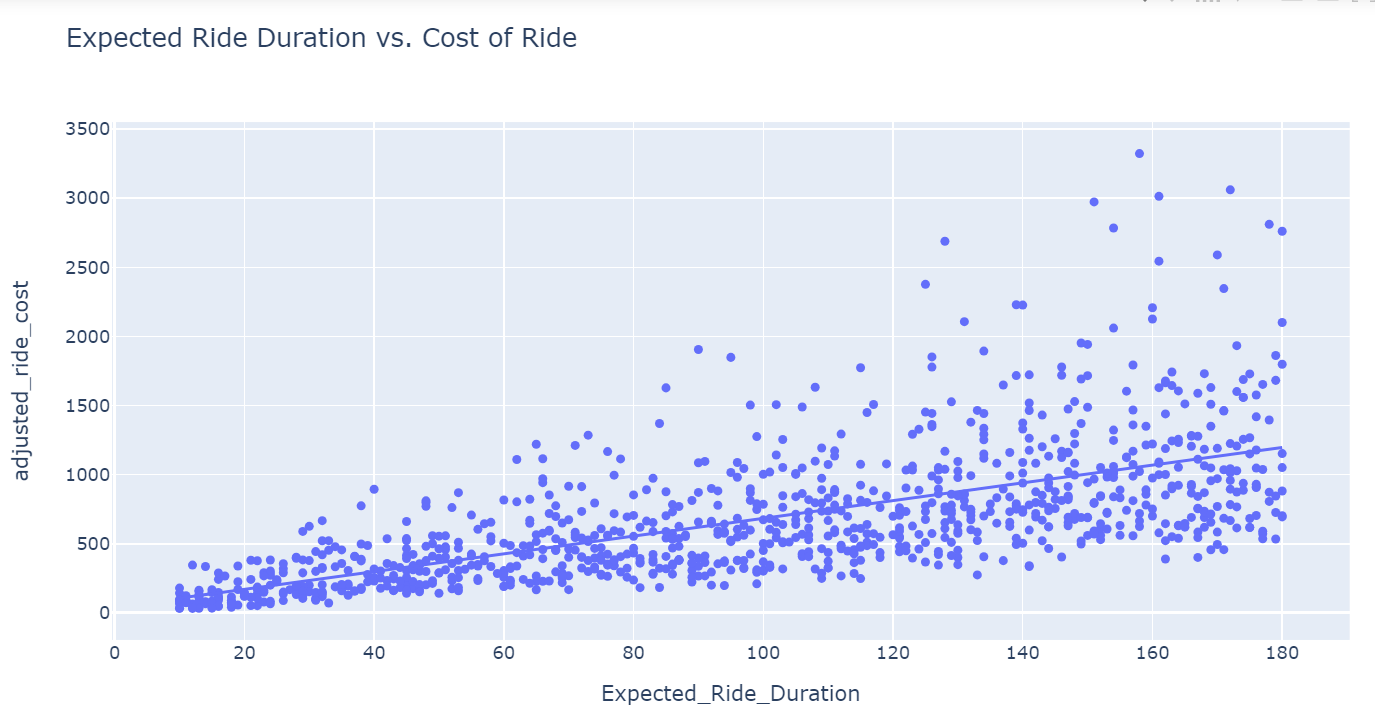
Now calculating the profit percentage we got after implementing this dynamic pricing strategy:

****

****

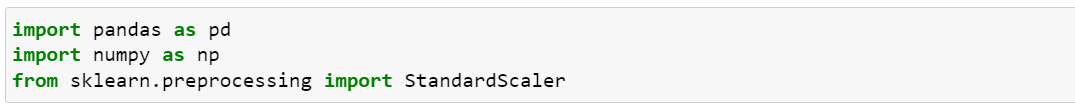
look at the relationship between the expected ride duration and the cost of the ride based on the dynamic pricing strategy:

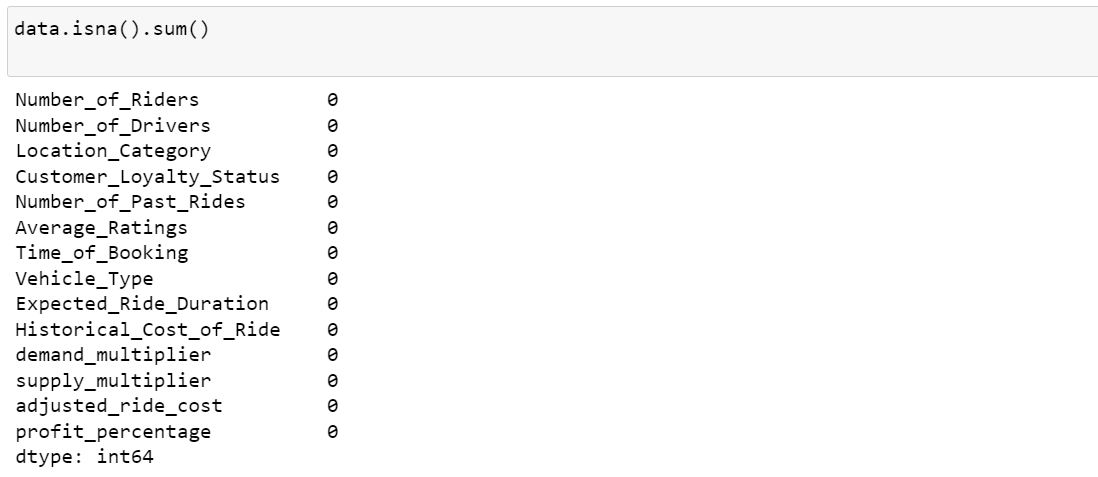
****

****

## Training a Predictive Model

Now, as implemented a dynamic pricing strategy, let’s train a Machine Learning model. Before training the model, let’s pre-process the data:

****

****

****

In the above code, implemented a data pre-processing pipeline to pre-process the data. As vehicle type is a valuable factor, convert it into a numerical feature before moving forward.

****

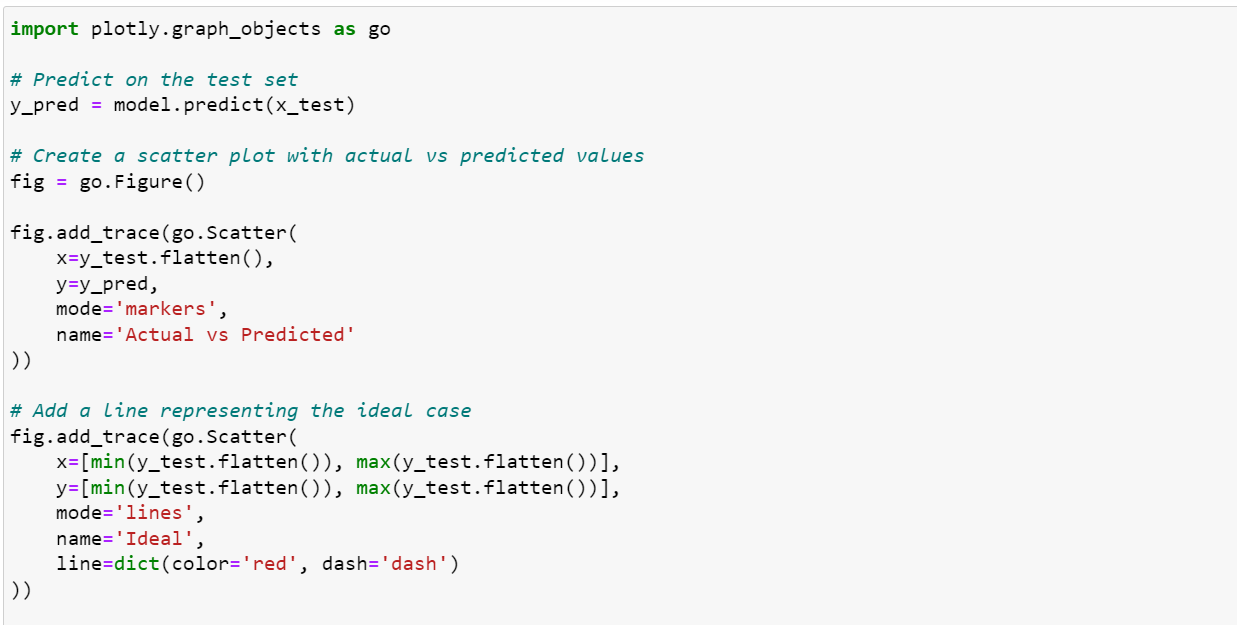
Now split the data and train a Machine Learning model to predict the cost of a ride

****

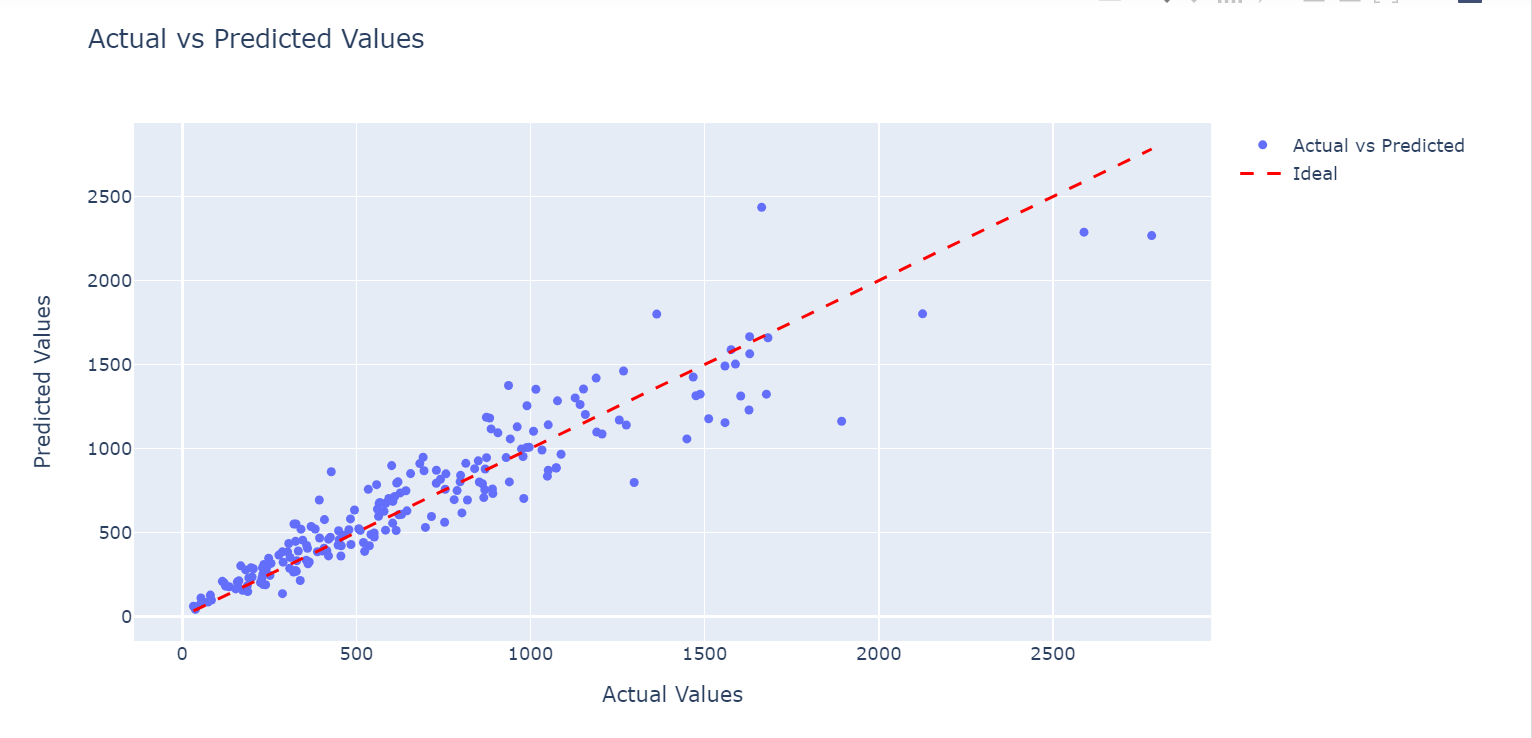
Now test this Machine Learning model using some input values:

****

Here’s a comparison of the actual and predicted results

****

****

****

So this is how you can use Machine Learning to implement a data-driven dynamic pricing strategy using Python.

### **Summary**

In a dynamic pricing strategy, the aim is to maximize revenue and profitability by pricing items at the right level that balances supply and demand dynamics. It allows businesses to adjust prices dynamically based on factors like time of day, day of the week, customer segments, inventory levels, seasonal fluctuations, competitor pricing, and market conditions.