Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship, meaning that changes in the independent variables correspond to changes in the dependent variable in a straight-line fashion. In simple linear regression, there's one independent variable, while in multiple linear regression, there are multiple independent variables. The goal in both cases is to find the best-fitting line (or plane in the case of multiple variables) that minimizes the difference between the observed values and the values predicted by the model. This is achieved by estimating the coefficients that define the slope and intercept of the line (or plane) in such a way that the sum of squared differences between observed and predicted values is minimized. Linear regression is widely used in various fields for prediction and understanding the relationships between variables, serving as a fundamental tool in statistical analysis and machine learning.

$$y = \beta_0 + \beta_1 \cdot x + \epsilon$$

Y - dependent variable, X - independent variable, beta0 - y intercept and beta1 - coefficient for the independent variable.

There are another formula for multiple linear regression if we have two or more independent variables.

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_n \cdot x_n + \epsilon$$

Lets think about the situation when the original ticket price of the airline company was 124 USD that is increasing by 4% each year already 5 years because of the inflation and predict the ticket price after another 5 year.

Let's assume today's date is January 1st, 2024. Based on this date, we can calculate the prices for the past 5 years, considering a 4% annual inflation rate.

The code below calculates the prices for the past 5 years based on the given original price (124USD) and annual inflation rate (4%). Each year, the price is increased by 4%.

```
# Original price of the airline ticket
original_price = 124
# Annual inflation rate
annual_inflation_rate = 0.04
# Number of years
num_years = 5
# Calculate the prices for the past 5 years
prices = []
current_price = original_price
for _ in range(num_years):
    current_price *= (1 + annual_inflation_rate)
    prices.append(current_price)
# Print the prices for the past 5 years
for i, price in enumerate(prices, start=1):
    print(f"Price after year {i}: ${price:.2f}")
```

So we got prices. And no I would use linear regression to predict the ticket price after another 5 years based on the prices we calculated just now (for the past 5 years).

In my code I use the prices calculated for the past 5 years as the training data.

I have created a linear regression model and fit it to the data and then, predict the price after another 5 years using the trained mode.

```
import numpy as np
from sklearn.linear_model import LinearRegression
# Years
years = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
# Prices calculated for the past 5 years
prices = np.array([128.96, 134.12, 139.48, 145.06, 150.86])
# Create and fit the linear regression model
model = LinearRegression()
model.fit(years, prices)
# Predict prices for the next 5 years
future_years = np.array([6, 7, 8, 9, 10]).reshape(-1, 1)  # Predicting for 5 additional years
predicted_prices = model.predict(future_years)
# Print the predicted prices for the next 5 years
for i, price in enumerate(predicted_prices, start=1):
    print(f"Predicted price after year {i+5}: ${price:.2f}")
```

```
C: > Users > user > Downloads > import numpy as np.py > i
import numpy as np
from sklearn.linear_model import LinearRegression

# Years
years = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)

# Prices
prices = np.array([128.96, 134.12, 139.48, 145.06, 150.86])  # Prices calculated for the past 5 y

# Create and fit the linear regression model
model = LinearRegression()
model.fit(years, prices)

# Predict prices for the next 5 years
future_years = np.array([6, 7, 8, 9, 10]).reshape(-1, 1)  # Predicting for 5 additional years
predicted_prices = model.predict(future_years)

# Print the predicted prices for the next 5 years
for i, price in enumerate(predicted_prices, start=1):
    print(f"Predicted price after year {i+5}: ${price:.2f}")
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\user> & C:/Users/user/anaconda3/python.exe "c:/Users/user/Downloads/import numpy as np.py"
Predicted price after year 6: $156.12
Predicted price after year 7: $161.59
Predicted price after year 8: $167.07
Predicted price after year 9: $172.54
Predicted price after year 10: $178.01
PS C:\Users\user>
```

So we got predicted prices. Now we can show it in a graph (using python code) extending the code to include plotting the predicted prices for the next 5 years on a graph.

```
# Plotting the observed prices and predicted prices
all_years = np.concatenate((years, future_years))
all_prices = np.concatenate((prices, predicted_prices))
plt.plot(all_years, all_prices, marker='o', linestyle='-', color='b')
plt.xlabel('Years')
plt.ylabel('Prices')
plt.title('Predicted Ticket Prices Over Time')
plt.grid(True)
plt.show()
```

```
plt.plot(all_years, all_prices, marker='o', linestyle='-', color='b')
plt.xlabel('Years')
plt.ylabel('Prices')
plt.title('Predicted Ticket Prices Over Time')
plt.grid(True)
plt.show()
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

Predicted price after year 7: $166.09
Predicted price after year 8: $171.64
Predicted price after year 9: $177.20
Predicted price after year 10: $182.76
```