


Start coding or [generate](#) with AI.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from datetime import datetime
```

```
from google.colab import files
uploaded = files.upload()
```


  **ola\_bike\_data.csv**  
• **ola\_bike\_data.csv**(text/csv) - 12895 bytes, last modified: 12/6/2025 - 100% done  
Saving ola\_bike\_data.csv to ola\_bike\_data.csv

```
df = pd.read_csv("ola_bike_data.csv")
```

```
df.head()
```

	Date	Hour	Ride_Requests
0	01-05-2025	0	61
1	01-05-2025	1	53
2	01-05-2025	2	51
3	01-05-2025	3	42
4	01-05-2025	4	59




Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
print(df.columns)
```

 `Index(['Date', 'Hour', 'Ride_Requests'], dtype='object')`

```
df.columns = df.columns.str.strip() # Remove any leading/trailing spaces
```




```
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True, errors='coerce')
```

```
df['Day'] = df['Date'].dt.day
df['Month'] = df['Date'].dt.month
df['Weekday'] = df['Date'].dt.weekday
```

```
X = df[['Hour', 'Day', 'Month', 'Weekday']]
y = df['Ride_Requests']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

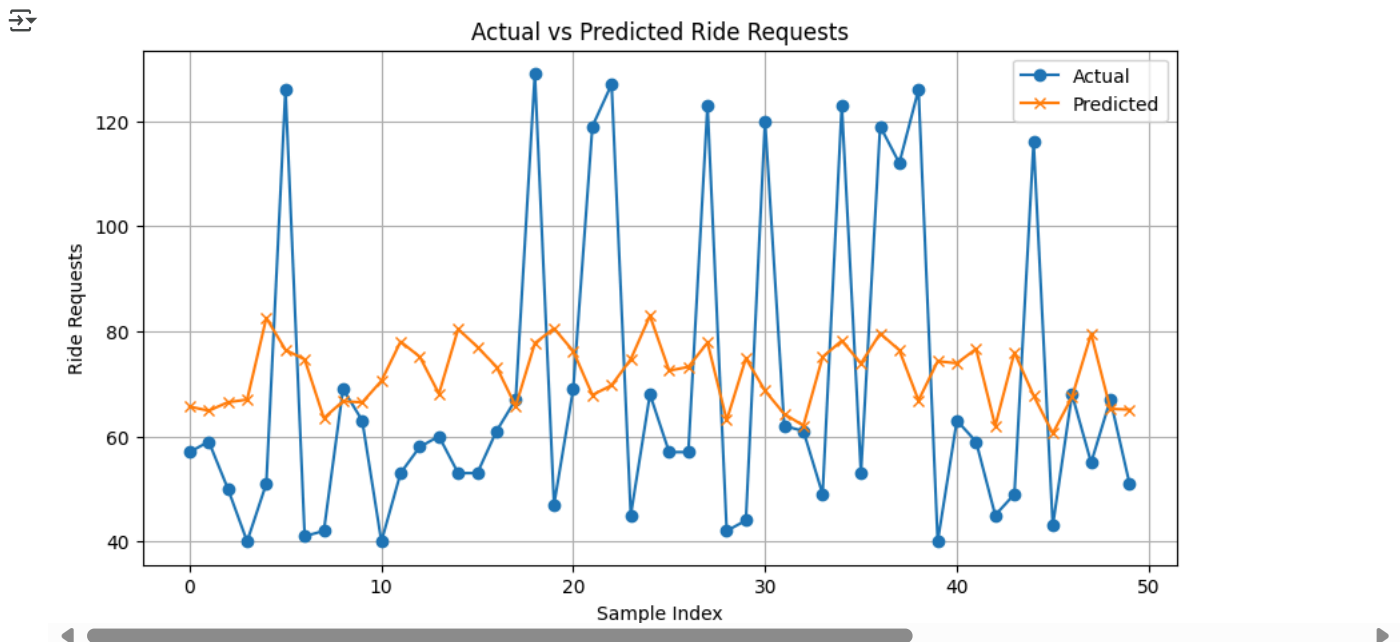
 `LinearRegression`    
`LinearRegression()`

```
y_pred = model.predict(X_test)
```

```
print("R² Score:", r2_score(y_test, y_pred))
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
```

🔗 R<sup>2</sup> Score: 0.0011553607925408604  
Mean Squared Error: 752.6754857718272

```
plt.figure(figsize=(10, 5))
plt.plot(y_test.values[:50], label='Actual', marker='o')
plt.plot(y_pred[:50], label='Predicted', marker='x')
plt.title("Actual vs Predicted Ride Requests")
plt.xlabel("Sample Index")
plt.ylabel("Ride Requests")
plt.legend()
plt.grid(True)
plt.show()
```



```
future_data = pd.DataFrame({
    'Hour': [10],
    'Day': [15],
    'Month': [6],
    'Weekday': [5]
})

future_prediction = model.predict(future_data)
print("🚲 Predicted ride requests for 15 June 2025, 10 AM:", int(future_prediction[0]))
```

🔗 🚲 Predicted ride requests for 15 June 2025, 10 AM: 70

