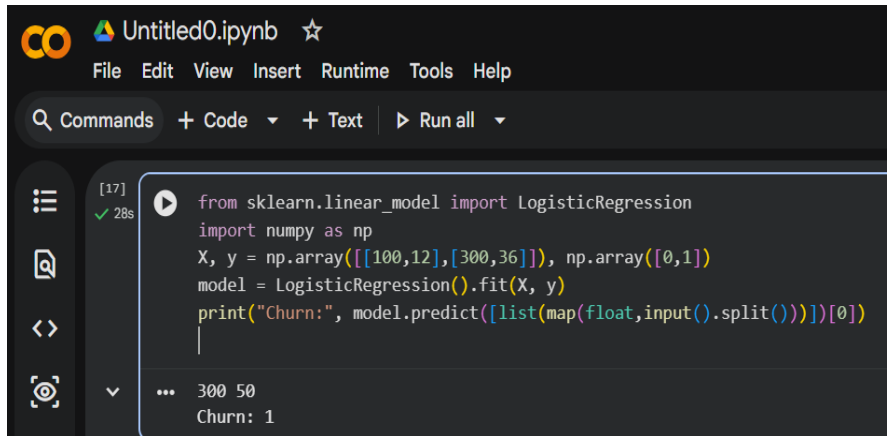


DAY – 7

[11.02.2025]

Exp : 26 – 30



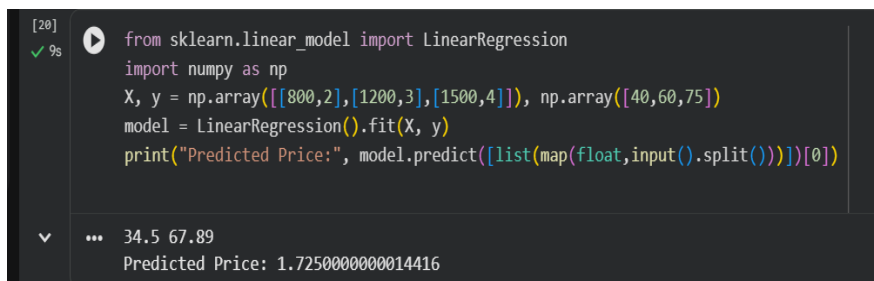
Untitled0.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

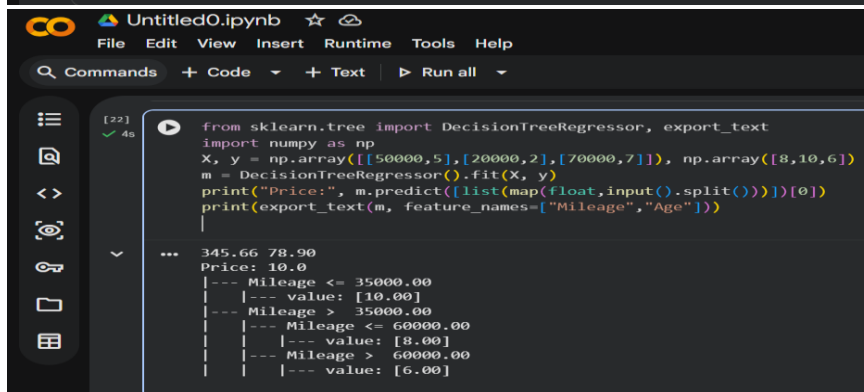
```
[17]
✓ 28s from sklearn.linear_model import LogisticRegression
import numpy as np
X, y = np.array([[100,12],[300,36]]), np.array([0,1])
model = LogisticRegression().fit(X, y)
print("Churn:", model.predict([list(map(float,input().split()))])[0])
```

... 300 50
Churn: 1



```
[20]
✓ 9s from sklearn.linear_model import LinearRegression
import numpy as np
X, y = np.array([[800,2],[1200,3],[1500,4]]), np.array([40,60,75])
model = LinearRegression().fit(X, y)
print("Predicted Price:", model.predict([list(map(float,input().split()))])[0])
```

... 34.5 67.89
Predicted Price: 1.7250000000014416



Untitled0.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
[22]
✓ 4s from sklearn.tree import DecisionTreeRegressor, export_text
import numpy as np
X, y = np.array([[50000,5],[20000,2],[70000,7]]), np.array([8,10,6])
m = DecisionTreeRegressor().fit(X, y)
print("Price:", m.predict([list(map(float,input().split()))])[0])
print(export_text(m, feature_names=["Mileage","Age"]))
```

... 345.66 78.90
Price: 10.0
|--- Mileage <= 35000.00
| |--- value: [10.00]
|--- Mileage > 35000.00
| |--- Mileage <= 60000.00
| | |--- value: [8.00]
| |--- Mileage > 60000.00
| | |--- value: [6.00]

CO

Untitled0.ipynb ☆

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

[18] ✓ 0s

▶

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import numpy as np
X, y = np.array([[1],[2],[3],[4]]), np.array([0,0,1,1])
m = LogisticRegression().fit(X, y); p = m.predict(X)
print(accuracy_score(y,p), precision_score(y,p), recall_score(y,p), f1_score(y,p))
```

... 1.0 1.0 1.0 1.0

CO

Untitled0.ipynb ☆

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

[19] ✓ 54s

▶

```
from sklearn.cluster import KMeans
import numpy as np
X = np.array([[200,2],[800,8],[300,3],[900,9]])
model = KMeans(n_clusters=2).fit(X)
print("Segment:", model.predict([list(map(float,input().split()))])[0])
```

... 55 70
Segment: 1