Predicting Airline Ticket Prices Using Machine Learning

Anika Ahmed

Problem:

Airfare prices fluctuate based on route, season, and travel behavior. Travelers and airlines struggle to anticipate pricing trends, making planning and decision-making inefficient.

Proposed Solution:

I built a machine learning model to predict airfare prices for U.S. flight routes using historical fare, passenger, and seasonal data.

Estimated Impact:

This model helps forecast ticket prices early, assisting travelers in deciding when to book and airlines in setting competitive fares. It improves transparency and reduces guesswork.

Data & Preprocessing

Data Source:

U.S. DOT Consumer Airfare Reports – Tables 3 & 4

Features Used:

Year, Quarter

ly_fare, cur_passengers, ly_passengers

Percent change in fares and passengers

One-hot encoded city-pair route variables

Preprocessing Steps:

Cleaned nulls and removed outliers

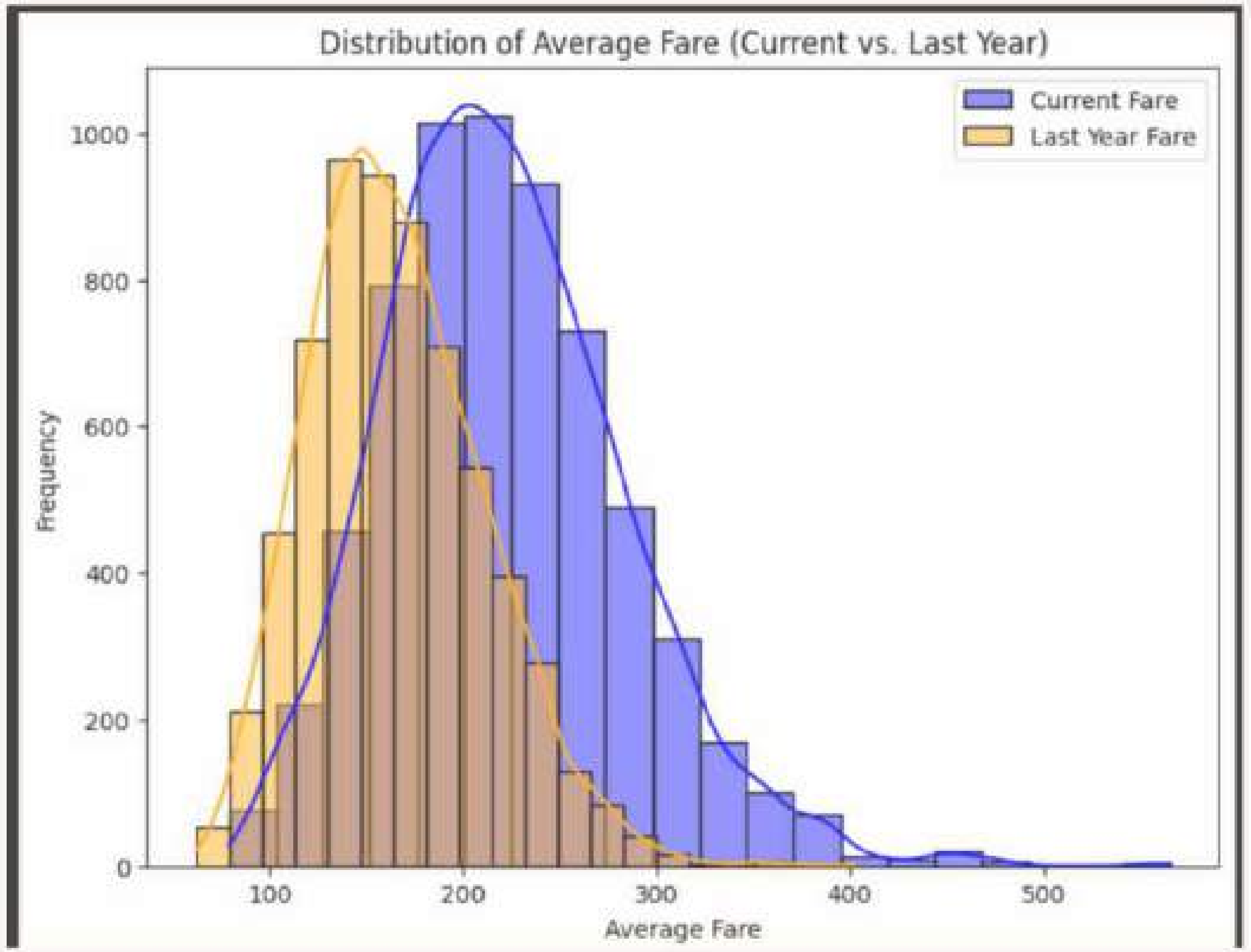
Encoded 50+ unique routes

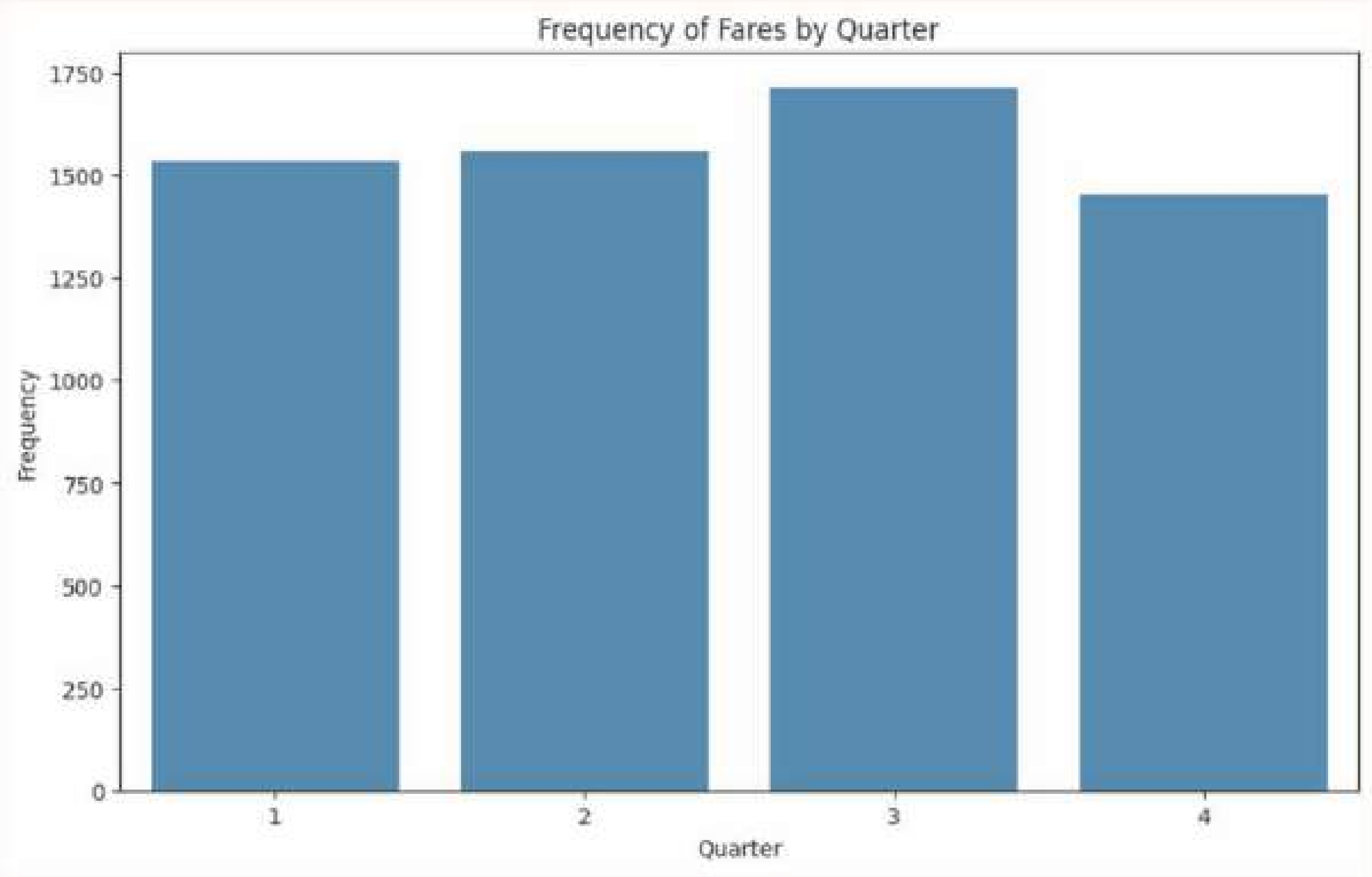
Feature-engineered percentage changes and route indicators

Final dataset used for regression modeling

Key Insights

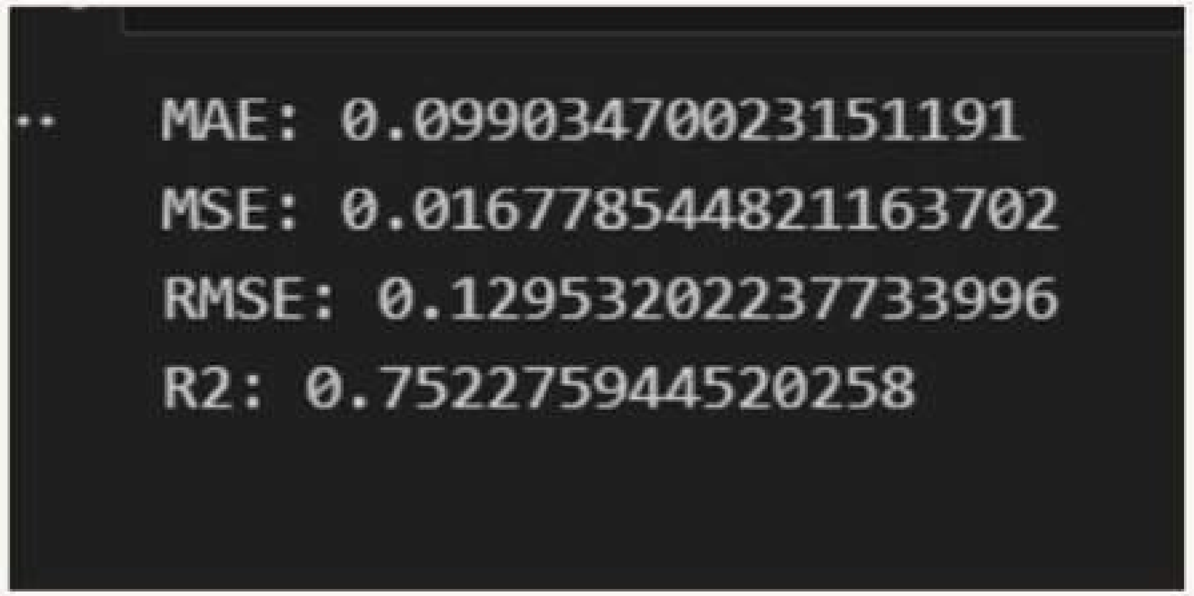
EDA





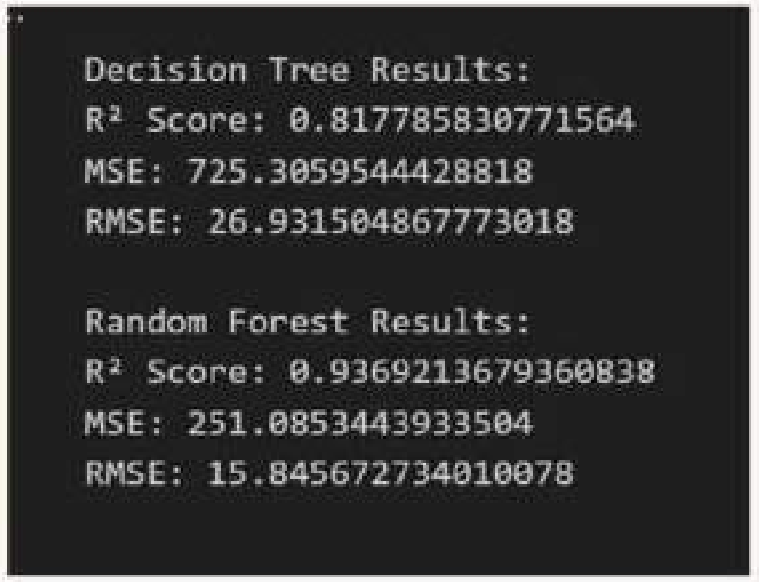
Model Accuracy

Price Linear regression



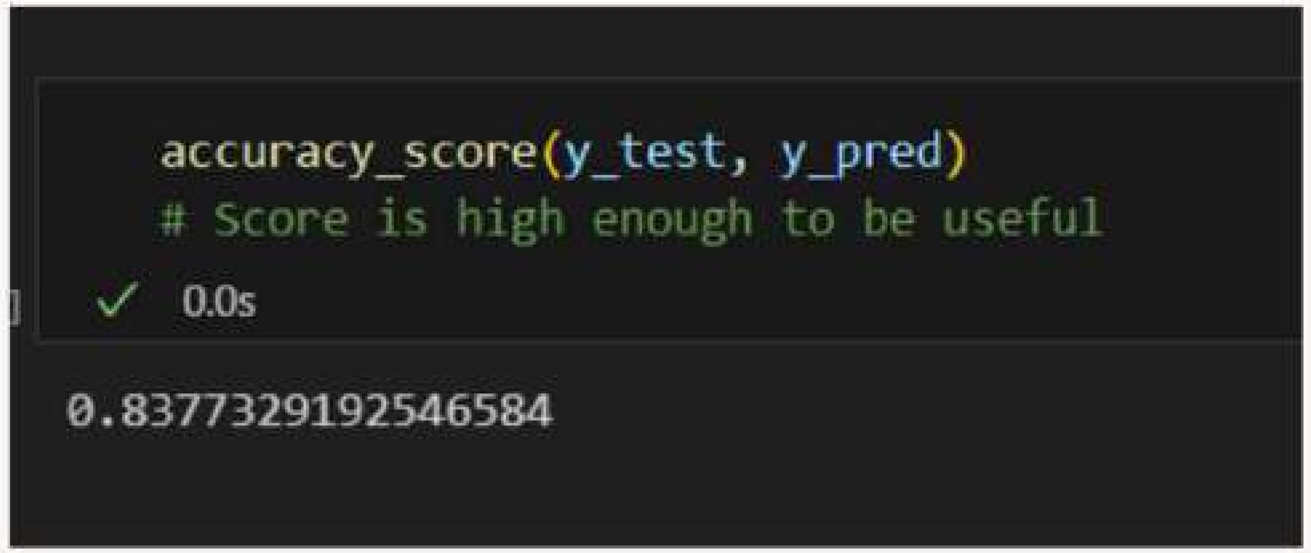
```
MAE: 0.09903470023151191
MSE: 0.016778544821163702
RMSE: 0.12953202237733996
R2: 0.752275944520258
```

Price prediction advanced model

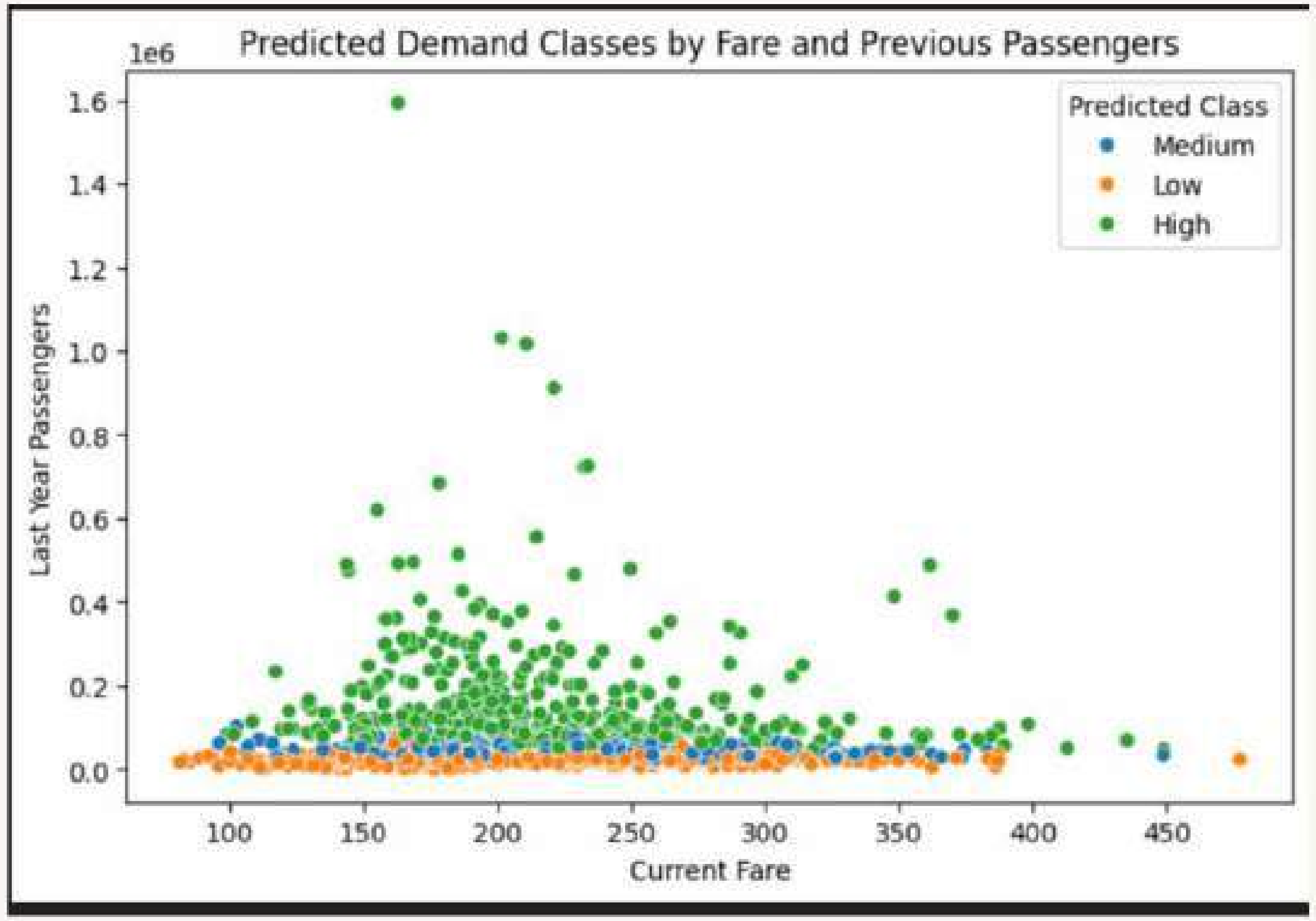


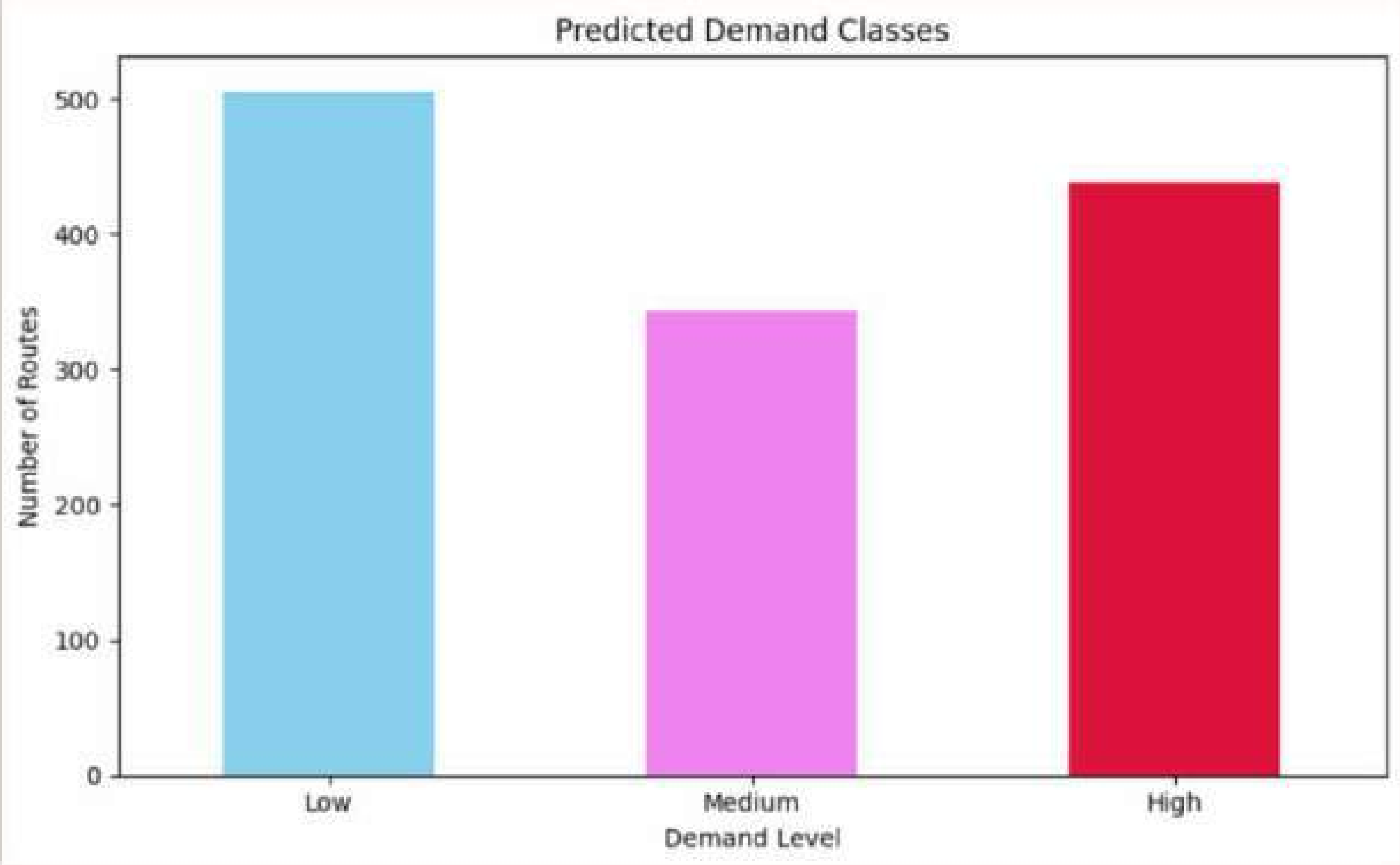
```
Decision Tree Results:
R² Score: 0.817785830771564
MSE: 725.3059544428818
RMSE: 26.931504867773018

Random Forest Results:
R² Score: 0.9369213679360838
MSE: 251.0853443933504
RMSE: 15.845672734010078
```

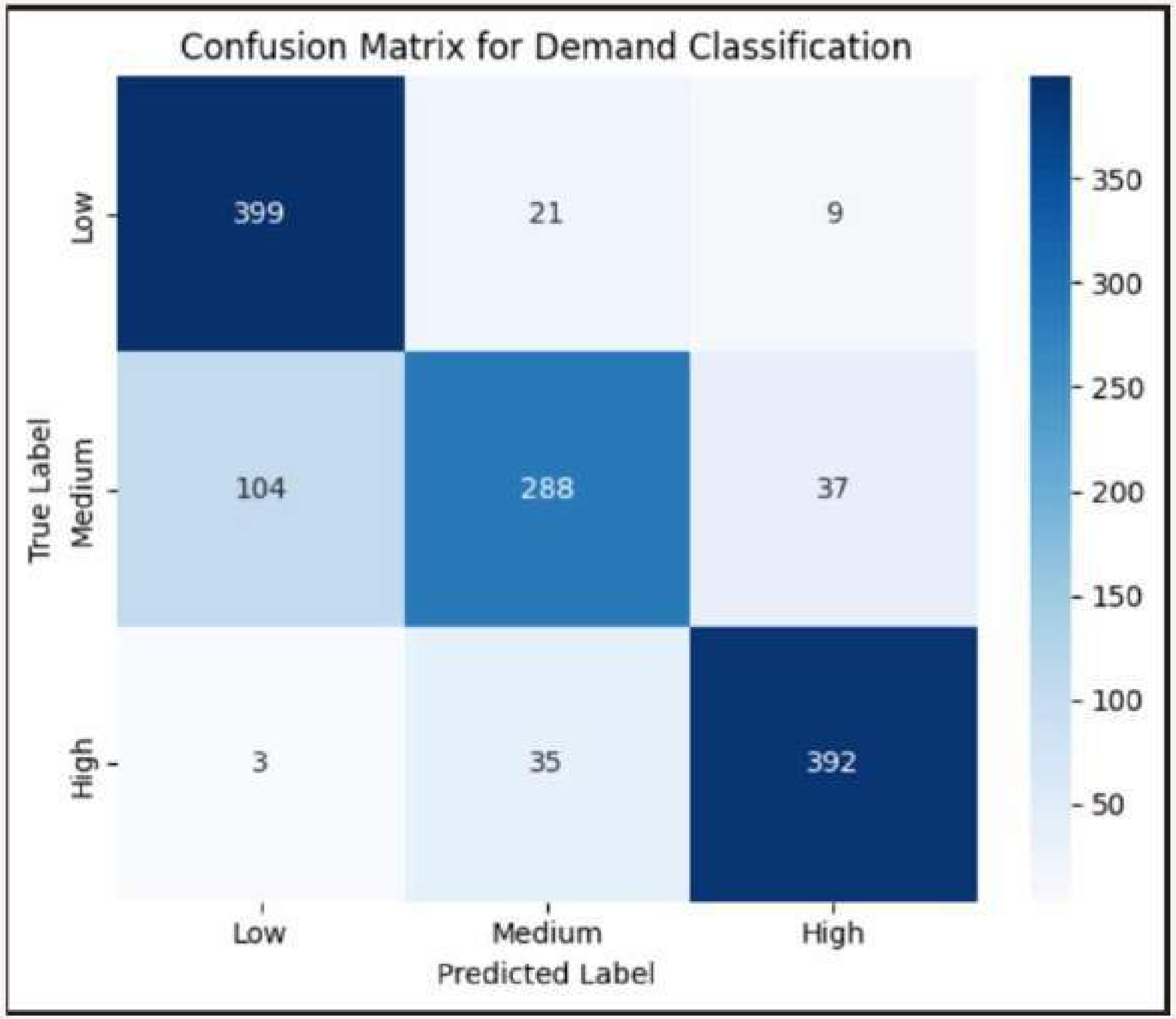
demand Logistic Model



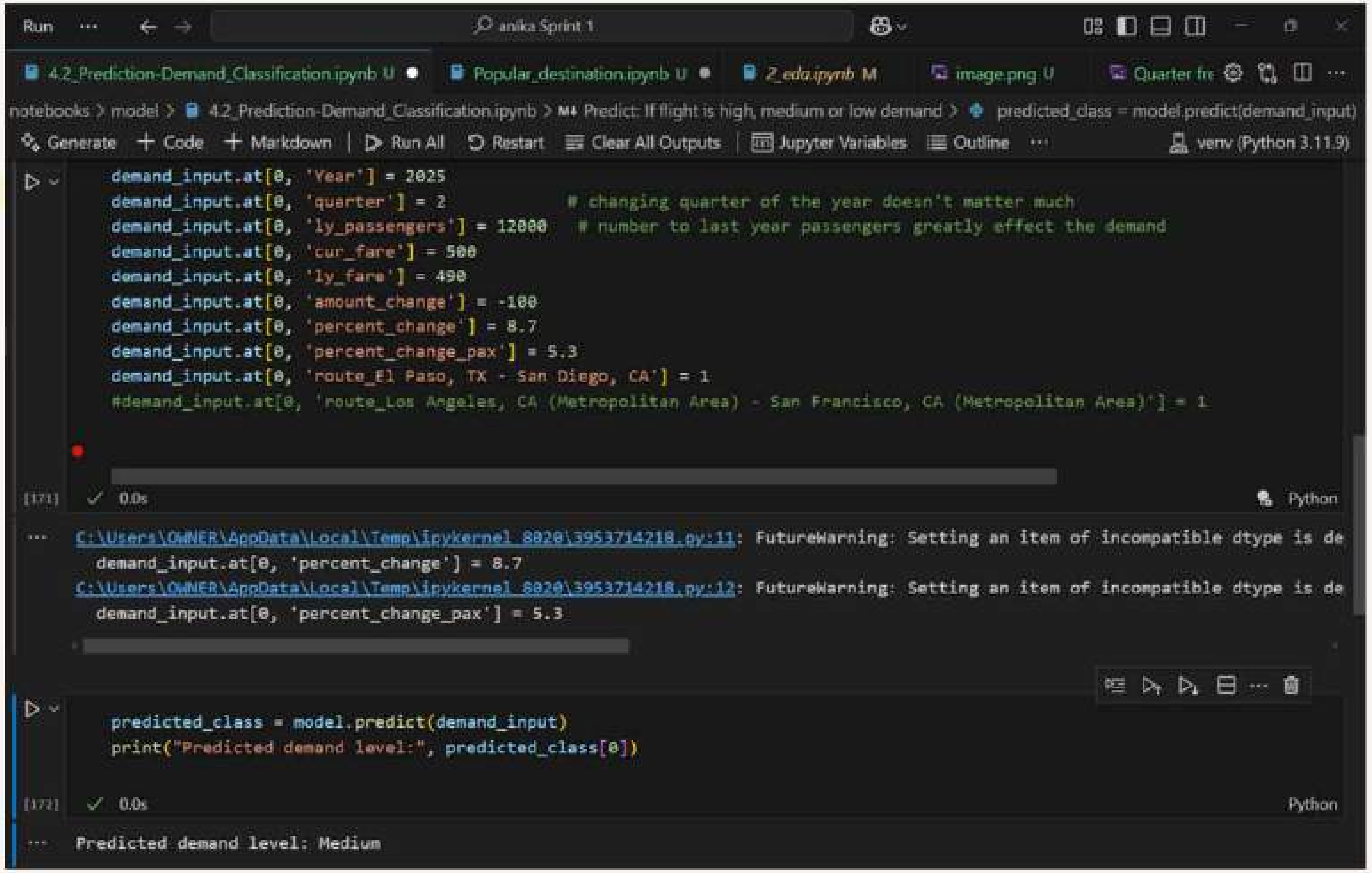
```
accuracy_score(y_test, y_pred)
# Score is high enough to be useful
✓ 0.0s

0.8377329192546584
```


Predicted Demand Classes by Fare and Previous Passengers
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Last Year Passengers
100
150
200
250
300
350
400
450
Current Fare
Predicted Class
● Medium
● Low
● High


Predicted Demand Classes
500
400
300
200
100
0
Number of Routes
Low
Medium
High
Demand Level


Confusion Matrix for Demand Classification
True Label
Low
Medium
High
399
21
9
104
288
37
3
35
392
Low
Medium
High
Predicted Label
350
300
250
200
150
100
50

Predict Demand level mode passenger Increase


Run
anika Sprint 1
4.2_Prediction-Demand_Classification.ipynb U
Popular_destination.ipynb U
2_eda.ipynb M
image.png U
Quarter fr
notebooks > model > 4.2_Prediction-Demand_Classification.ipynb > M↓ Predict: If flight is high, medium or low demand > predicted_class = model.predict(demand_input)
Generate
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
venv (Python 3.11.9)
[171]
0.0s
Python
[172]
0.0s
Python

```python
demand_input.at[0, 'Year'] = 2025
demand_input.at[0, 'quarter'] = 2            # changing quarter of the year doesn't matter much
demand_input.at[0, 'ly_passengers'] = 12000   # number to last year passengers greatly effect the demand
demand_input.at[0, 'cur_fare'] = 500
demand_input.at[0, 'ly_fare'] = 490
demand_input.at[0, 'amount_change'] = -100
demand_input.at[0, 'percent_change'] = 8.7
demand_input.at[0, 'percent_change_pax'] = 5.3
demand_input.at[0, 'route_El Paso, TX - San Diego, CA'] = 1
#demand_input.at[0, 'route_Los Angeles, CA (Metropolitan Area) - San Francisco, CA (Metropolitan Area)'] = 1
```

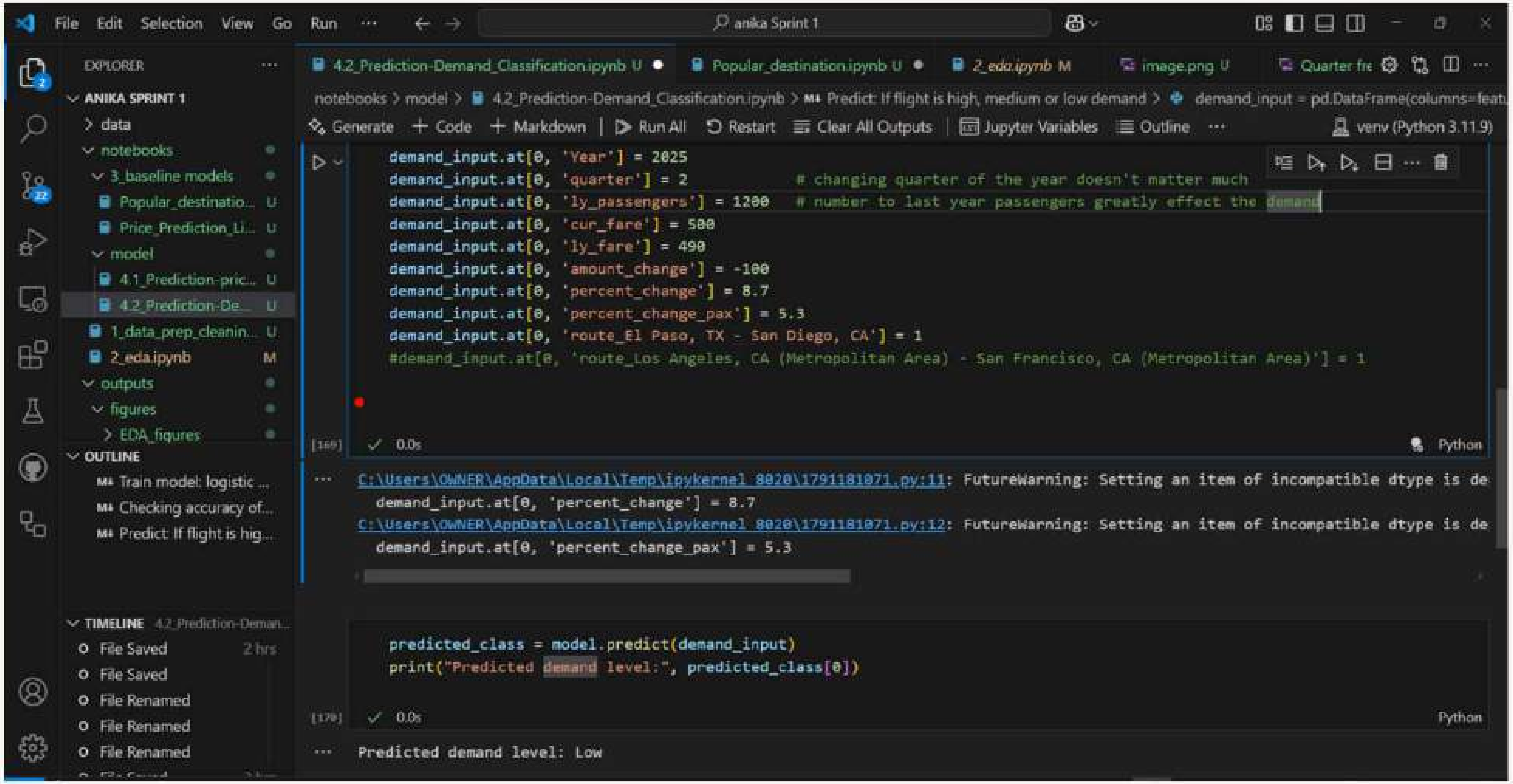
```
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\3953714218.py:11: FutureWarning: Setting an item of incompatible dtype is de
  demand_input.at[0, 'percent_change'] = 8.7
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\3953714218.py:12: FutureWarning: Setting an item of incompatible dtype is de
  demand_input.at[0, 'percent_change_pax'] = 5.3
```

```python
predicted_class = model.predict(demand_input)
print("Predicted demand level:", predicted_class[0])
```

```
Predicted demand level: Medium
```

Predict Demand level mode

passenger decrease



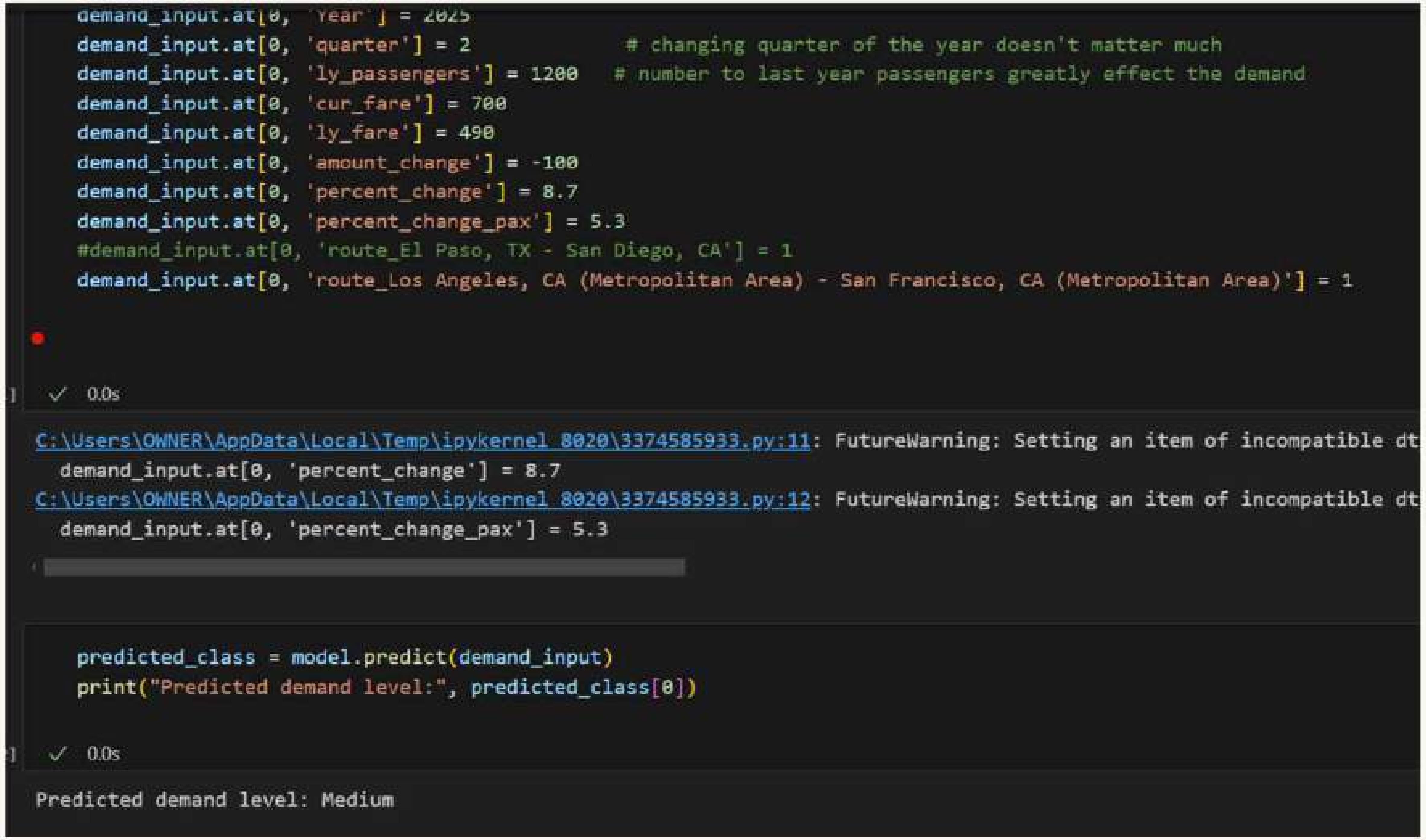
```python
demand_input.at[0, 'Year'] = 2025
demand_input.at[0, 'quarter'] = 2          # changing quarter of the year doesn't matter much
demand_input.at[0, 'ly_passengers'] = 1200  # number to last year passengers greatly effect the demand
demand_input.at[0, 'cur_fare'] = 500
demand_input.at[0, 'ly_fare'] = 490
demand_input.at[0, 'amount_change'] = -100
demand_input.at[0, 'percent_change'] = 8.7
demand_input.at[0, 'percent_change_pax'] = 5.3
demand_input.at[0, 'route_El Paso, TX - San Diego, CA'] = 1
#demand_input.at[0, 'route_Los Angeles, CA (Metropolitan Area) - San Francisco, CA (Metropolitan Area)'] = 1
```

```
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\1791181071.py:11: FutureWarning: Setting an item of incompatible dtype is de
  demand_input.at[0, 'percent_change'] = 8.7
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\1791181071.py:12: FutureWarning: Setting an item of incompatible dtype is de
  demand_input.at[0, 'percent_change_pax'] = 5.3
```

```python
predicted_class = model.predict(demand_input)
print("Predicted demand level:", predicted_class[0])
```

```
Predicted demand level: Low
```

PRICE INCREASE



```
    demand_input.at[0, 'Year'] = 2025
    demand_input.at[0, 'quarter'] = 2             # changing quarter of the year doesn't matter much
    demand_input.at[0, 'ly_passengers'] = 1200   # number to last year passengers greatly effect the demand
    demand_input.at[0, 'cur_fare'] = 700
    demand_input.at[0, 'ly_fare'] = 490
    demand_input.at[0, 'amount_change'] = -100
    demand_input.at[0, 'percent_change'] = 8.7
    demand_input.at[0, 'percent_change_pax'] = 5.3
    #demand_input.at[0, 'route_El Paso, TX - San Diego, CA'] = 1
    demand_input.at[0, 'route_Los Angeles, CA (Metropolitan Area) - San Francisco, CA (Metropolitan Area)'] = 1
```

✓ 0.0s

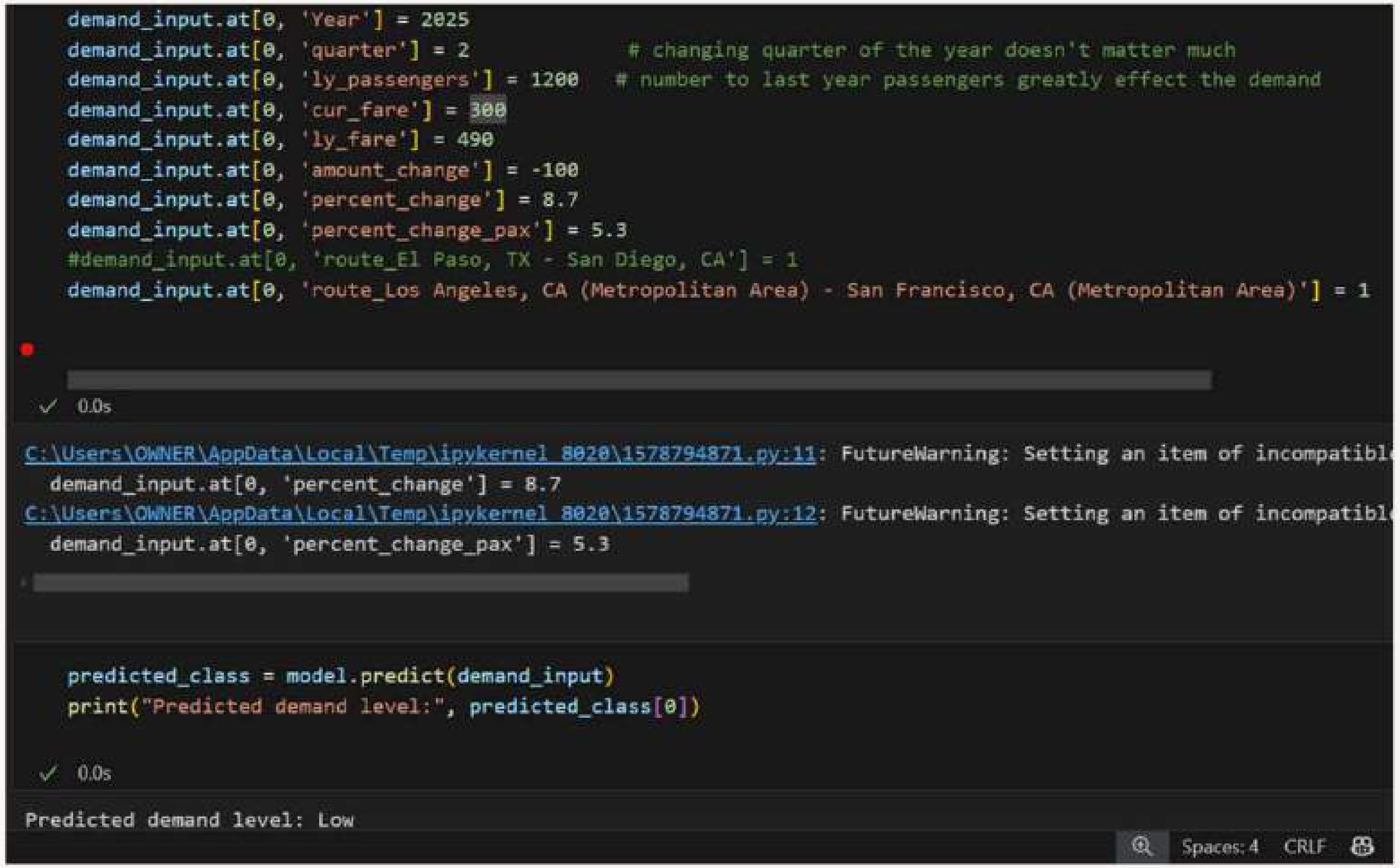
```
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\3374585933.py:11: FutureWarning: Setting an item of incompatible dt
  demand_input.at[0, 'percent_change'] = 8.7
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\3374585933.py:12: FutureWarning: Setting an item of incompatible dt
  demand_input.at[0, 'percent_change_pax'] = 5.3
```

```
    predicted_class = model.predict(demand_input)
    print("Predicted demand level:", predicted_class[0])
```

✓ 0.0s

```
Predicted demand level: Medium
```

PRICE DECREASE



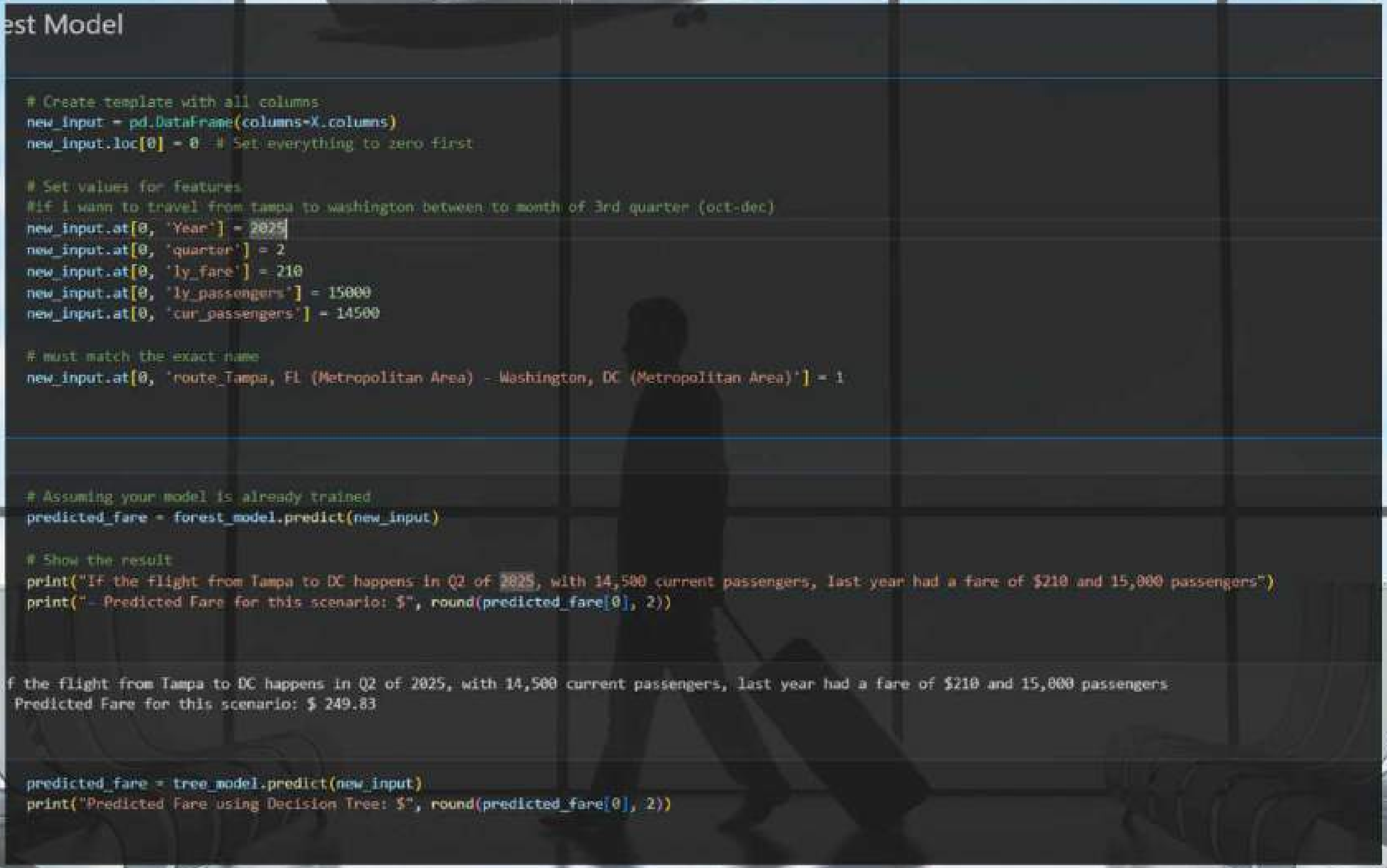
```
    demand_input.at[0, 'Year'] = 2025
    demand_input.at[0, 'quarter'] = 2              # changing quarter of the year doesn't matter much
    demand_input.at[0, 'ly_passengers'] = 1200   # number to last year passengers greatly effect the demand
    demand_input.at[0, 'cur_fare'] = 300
    demand_input.at[0, 'ly_fare'] = 490
    demand_input.at[0, 'amount_change'] = -100
    demand_input.at[0, 'percent_change'] = 8.7
    demand_input.at[0, 'percent_change_pax'] = 5.3
    #demand_input.at[0, 'route_El Paso, TX - San Diego, CA'] = 1
    demand_input.at[0, 'route_Los Angeles, CA (Metropolitan Area) - San Francisco, CA (Metropolitan Area)'] = 1
```

```
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\1578794871.py:11: FutureWarning: Setting an item of incompatibl
  demand_input.at[0, 'percent_change'] = 8.7
C:\Users\OWNER\AppData\Local\Temp\ipykernel_8020\1578794871.py:12: FutureWarning: Setting an item of incompatibl
  demand_input.at[0, 'percent_change_pax'] = 5.3
```

```
    predicted_class = model.predict(demand_input)
    print("Predicted demand level:", predicted_class[0])
```

```
Predicted demand level: Low
```

Predict Price Model



est Model

```
# Create template with all columns
new_input = pd.DataFrame(columns=X.columns)
new_input.loc[0] = 0  # Set everything to zero first

# Set values for features
#if i wann to travel from tampa to washington between to month of 3rd quarter (oct-dec)
new_input.at[0, 'Year'] = 2025
new_input.at[0, 'quarter'] = 2
new_input.at[0, 'ly_fare'] = 210
new_input.at[0, 'ly_passengers'] = 15000
new_input.at[0, 'cur_passengers'] = 14500

# must match the exact name
new_input.at[0, 'route_Tampa, FL (Metropolitan Area) - Washington, DC (Metropolitan Area)'] = 1
```

```
# Assuming your model is already trained
predicted_fare = forest_model.predict(new_input)

# Show the result
print("If the flight from Tampa to DC happens in Q2 of 2025, with 14,500 current passengers, last year had a fare of $210 and 15,000 passengers")
print("- Predicted Fare for this scenario: $", round(predicted_fare[0], 2))
```

```
f the flight from Tampa to DC happens in Q2 of 2025, with 14,500 current passengers, last year had a fare of $210 and 15,000 passengers
 Predicted Fare for this scenario: $ 249.83
```

```
predicted_fare = tree_model.predict(new_input)
print("Predicted Fare using Decision Tree: $", round(predicted_fare[0], 2))
```

Thank You