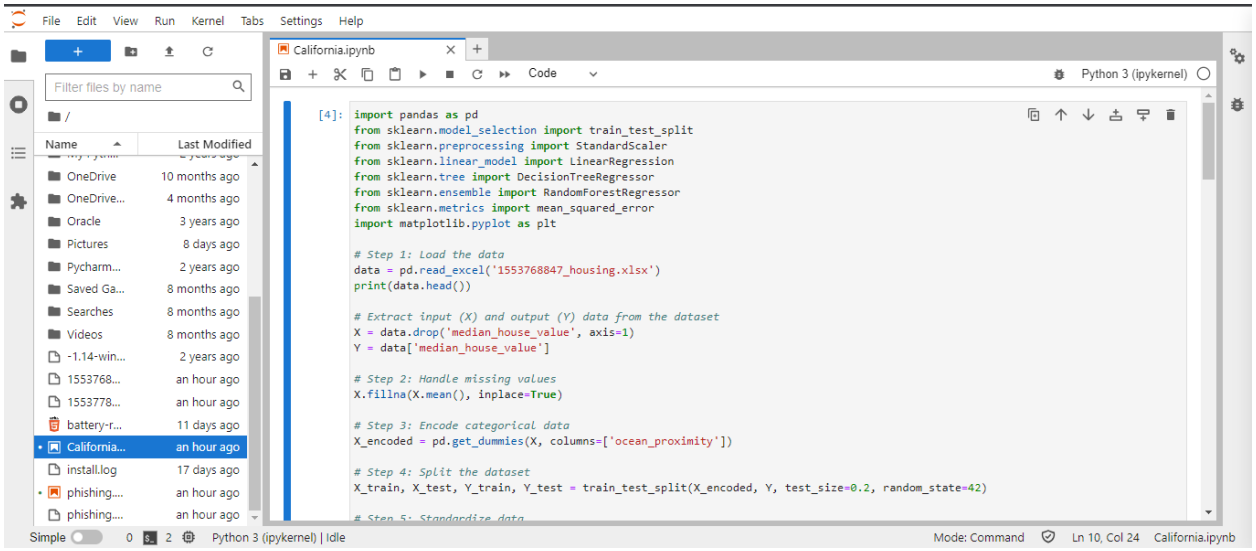
File Edit View Run Kernel Tabs Settings Help
Filter files by name
Name
Last Modified
OneDrive 10 months ago
OneDrive... 4 months ago
Oracle 3 years ago
Pictures 8 days ago
Pycharm... 2 years ago
Saved Ga... 8 months ago
Searches 8 months ago
Videos 8 months ago
-1.14-win... 2 years ago
1553768... an hour ago
1553778... an hour ago
battery-r... 11 days ago
California... an hour ago
install.log 17 days ago
phishing.... an hour ago
phishing.... an hour ago
California.ipynb
Code
Python 3 (ipykernel)
[4]:
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
# Step 1: Load the data
data = pd.read_excel('1553768847_housing.xlsx')
print(data.head())
# Extract input (X) and output (Y) data from the dataset
X = data.drop('median_house_value', axis=1)
Y = data['median_house_value']
# Step 2: Handle missing values
X.fillna(X.mean(), inplace=True)
# Step 3: Encode categorical data
X_encoded = pd.get_dummies(X, columns=['ocean_proximity'])
# Step 4: Split the dataset
X_train, X_test, Y_train, Y_test = train_test_split(X_encoded, Y, test_size=0.2, random_state=42)
Simple 0 2 Python 3 (ipykernel) | Idle
Mode: Command Ln 10, Col 24 California.ipynb


California.ipynb
Code
Python 3 (ipykernel)
# Step 5: Standardize data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Step 6: Perform Linear Regression
linear_reg = LinearRegression()
linear_reg.fit(X_train_scaled, Y_train)
Y_pred_linear = linear_reg.predict(X_test_scaled)
linear_rmse = mean_squared_error(Y_test, Y_pred_linear, squared=False)
print('Linear Regression RMSE:', linear_rmse)
# Step 7: Perform Decision Tree Regression
dt_reg = DecisionTreeRegressor()
dt_reg.fit(X_train_scaled, Y_train)
Y_pred_dt = dt_reg.predict(X_test_scaled)
dt_rmse = mean_squared_error(Y_test, Y_pred_dt, squared=False)
print('Decision Tree Regression RMSE:', dt_rmse)
# Step 8: Perform Random Forest Regression
rf_reg = RandomForestRegressor()
rf_reg.fit(X_train_scaled, Y_train)
Y_pred_rf = rf_reg.predict(X_test_scaled)
rf_rmse = mean_squared_error(Y_test, Y_pred_rf, squared=False)
print('Random Forest Regression RMSE:', rf_rmse)
# Step 9: Bonus exercise - Linear Regression with one independent variable

California.ipynb

Code

Python 3 (ipykernel)

```
# Step 9: Bonus exercise - Linear Regression with one independent variable
X_train_income = X_train_scaled[:, X_encoded.columns.tolist().index('median_income')]
X_test_income = X_test_scaled[:, X_encoded.columns.tolist().index('median_income')]

linear_reg_income = LinearRegression()
linear_reg_income.fit(X_train_income.reshape(-1, 1), Y_train)
Y_pred_income = linear_reg_income.predict(X_test_income.reshape(-1, 1))

# Plot the fitted model for training data and test data
plt.scatter(X_train_income, Y_train, color='blue', label='Training Data')
plt.scatter(X_test_income, Y_test, color='green', label='Test Data')
plt.plot(X_test_income, Y_pred_income, color='red', linewidth=2, label='Fitted Model')
plt.xlabel('Median Income')
plt.ylabel('Median House Value')
plt.legend()
plt.show()
```

```
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -122.23     37.88                  41          880           129.0
1    -122.22     37.86                  21         7099          1106.0
2    -122.24     37.85                  52         1467           190.0
3    -122.25     37.85                  52         1274           235.0
4    -122.25     37.85                  52         1627           280.0

   population  households  median_income ocean_proximity  median_house_value
0         322         126         8.3252        NEAR BAY              452600
1        2401        1138         8.3014        NEAR BAY              358500
2         496         177         7.2574        NEAR BAY              352100
```

```
   population  households  median_income ocean_proximity  median_house_value
0         322         126         8.3252        NEAR BAY              452600
1        2401        1138         8.3014        NEAR BAY              358500
2         496         177         7.2574        NEAR BAY              352100
3         558         219         5.6431        NEAR BAY              341300
4         565         259         3.8462        NEAR BAY              342200
C:\Users\HP\AppData\Local\Temp\ipykernel_23004\1622587805.py:19: FutureWarning: The default value of numeric_only in DataFrame.
mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. S
elect only valid columns or specify the value of numeric_only to silence this warning.
  X.fillna(X.mean(), inplace=True)
Linear Regression RMSE: 70031.41991955665
Decision Tree Regression RMSE: 68752.32914617851
Random Forest Regression RMSE: 49019.543465157796
```

California.ipynb

Code

Python 3 (ipykernel)

```
Decision Tree Regression RMSE: 68752.32914617851
Random Forest Regression RMSE: 49019.543465157796
```