Lab_4

March 17, 2024

```
[1]: import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import mean_squared_error
```

```
[ ]: # Load the dataset
```

```
[2]: data = pd.read_csv("/home/princeton/Downloads/College/AI_IN_Enterprise/Fish␣
      ↪market dataset/Fish.csv")
```

```
[3]: data
```

```
[3]:      Species  Weight  Length1  Length2  Length3   Height    Width
     0      Bream   242.0     23.2     25.4     30.0  11.5200   4.0200
     1      Bream   290.0     24.0     26.3     31.2  12.4800   4.3056
     2      Bream   340.0     23.9     26.5     31.1  12.3778   4.6961
     3      Bream   363.0     26.3     29.0     33.5  12.7300   4.4555
     4      Bream   430.0     26.5     29.0     34.0  12.4440   5.1340
     ..       ...     ...      ...      ...      ...      ...      ...
     154    Smelt    12.2     11.5     12.2     13.4   2.0904   1.3936
     155    Smelt    13.4     11.7     12.4     13.5   2.4300   1.2690
     156    Smelt    12.2     12.1     13.0     13.8   2.2770   1.2558
     157    Smelt    19.7     13.2     14.3     15.2   2.8728   2.0672
     158    Smelt    19.9     13.8     15.0     16.2   2.9322   1.8792

     [159 rows x 7 columns]
```

```
[ ]: # Perform one-hot encoding for the 'Species' column
```

```
[4]: data = pd.get_dummies(data, columns=["Species"])
```

```
[14]: print("Number of features after one-hot encoding:", data.shape[1])

     Number of features after one-hot encoding: 13
```

```
[ ]: # Split the data into features (X) and target variable (y)
```

```
[5]: X = data.drop(columns=["Weight"])
     y = data["Weight"]
```

```
[ ]: # Split the data into training and testing sets
```

```
[6]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
     ↪random_state=42)
```

```
[ ]: # Initialize and train a Linear Regression model
```

```
[7]: model = LinearRegression()
     model.fit(X_train, y_train)
```

```
[7]: LinearRegression()
```

```
[ ]: # Make predictions on the testing set
```

```
[8]: predictions = model.predict(X_test)
```

```
[ ]: # Evaluate the model
```

```
[9]: mse = mean_squared_error(y_test, predictions)
     print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 7007.383189853857
```

```
[ ]: # Output actual and predicted weights side by side
```

```
[10]: output = pd.DataFrame({'Actual': y_test, 'Predicted': predictions})
      print(output)
```

```
        Actual    Predicted
78        78.0    18.738254
155       13.4    11.886420
128      200.0   187.711281
55       270.0   332.739895
94       150.0   212.097016
29      1000.0   757.761616
147        7.0   -61.143019
51       180.0   255.772525
98       188.0   260.494792
141     1250.0  1148.322830
19       650.0   600.428130
60      1000.0   852.956669
15       600.0   541.634898
65       150.0   140.057965
24       700.0   680.963249
```

```
30      920.0    818.784983
126    1000.0   1009.771417
101     218.0    289.704760
96      225.0    226.067167
16      700.0    585.138874
151      10.0    -11.022652
18      610.0    585.707041
12      500.0    509.582200
9       500.0    492.021106
31      955.0    814.270866
125    1100.0    964.090051
95      170.0    219.240454
56      270.0    356.938198
145       6.7    -90.406643
152       9.9    -11.874622
135     510.0    575.807996
76       70.0    -18.412527
```

[11]:
```python
from joblib import dump
```

[12]:
```python
dump(model, 'linear_regression_model.pkl')
```

[12]:
```
['linear_regression_model.pkl']
```

[13]:
```python
ls
```

```
archive/        linear_regression_model.pkl
hello.ipynb     New_project/
Lab_1.ipynb    'Ted Petrou - Pandas Cookbook-Packt Publishing (2017).pdf'
Lab_4.ipynb     Untitled.ipynb
```

[ ]: