spark

February 19, 2025

1 1.Verilerin Yüklenmesi

Spark session başlatılarak veriseti çekilir.

[3]:
```
from pyspark.sql import SparkSession


spark = SparkSession.builder.appName("HousingPricePrediction").getOrCreate()

data = spark.read.csv("/content/drive/MyDrive/Colab Notebooks/spark/housing.
 ↪csv", header=True, inferSchema=True)
```

2 2.Verilerin İncelenmesi

Veriseti, sütunlardaki verilerin ortalama, medyan, maximum, minimum gibi değerleri baz alınarak incelenmiştir.

[4]:
```
data.show(5)
```

```
+---------+--------+------------------+-----------+--------------+----------+---
-------+-------------+------------------+---------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|hou
seholds|median_income|median_house_value|ocean_proximity|
+---------+--------+------------------+-----------+--------------+----------+---
-------+-------------+------------------+---------------+
|  -122.23|   37.88|              41.0|      880.0|         129.0|     322.0|
126.0|       8.3252|          452600.0|       NEAR BAY|
|  -122.22|   37.86|              21.0|     7099.0|        1106.0|    2401.0|
1138.0|       8.3014|          358500.0|       NEAR BAY|
|  -122.24|   37.85|              52.0|     1467.0|         190.0|     496.0|
177.0|       7.2574|          352100.0|       NEAR BAY|
|  -122.25|   37.85|              52.0|     1274.0|         235.0|     558.0|
219.0|       5.6431|          341300.0|       NEAR BAY|
|  -122.25|   37.85|              52.0|     1627.0|         280.0|     565.0|
259.0|       3.8462|          342200.0|       NEAR BAY|
+---------+--------+------------------+-----------+--------------+----------+---
-------+-------------+------------------+---------------+
```

```
only showing top 5 rows
```

[5]:
```
data.describe().show()
```

```
+-------+------------------+-----------------+------------------+-------------
----+-----------------+-----------------+----------------+------------------+
------------------+---------------+
|summary|          longitude|         latitude|housing_median_age|
total_rooms|    total_bedrooms|       population|      households|
median_income|median_house_value|ocean_proximity|
+-------+------------------+-----------------+------------------+-------------
----+-----------------+-----------------+----------------+------------------+
------------------+---------------+
|  count|              20640|            20640|             20640|
20640|             20433|            20640|           20640|
20640|             20640|          20640|
|   mean|-119.56970445736148|
35.6318614341087|28.639486434108527|2635.7630813953488| 537.8705525375618|1425.4
767441860465|499.5396802325581|3.8706710029070246|206855.81690891474|
NULL|
| stddev|  2.003531723502584|2.135952397457101|
12.58555761211163|2181.6152515827944|421.38507007403115|
1132.46212176534|382.3297528316098| 1.899821717945263|115395.61587441359|
NULL|
|    min|            -124.35|            32.54|               1.0|
2.0|               1.0|              3.0|             1.0|            0.4999|
14999.0|      <1H OCEAN|
|    max|            -114.31|            41.95|              52.0|
39320.0|            6445.0|           35682.0|          6082.0|
15.0001|          500001.0|     NEAR OCEAN|
+-------+------------------+-----------------+------------------+-------------
----+-----------------+-----------------+----------------+------------------+
------------------+---------------+
```

[7]:
```
data.summary().show()
```

```
+-------+------------------+-----------------+------------------+-------------
----+-----------------+-----------------+----------------+------------------+
------------------+---------------+
|summary|          longitude|         latitude|housing_median_age|
total_rooms|    total_bedrooms|       population|      households|
median_income|median_house_value|ocean_proximity|
+-------+------------------+-----------------+------------------+-------------
----+-----------------+-----------------+----------------+------------------+
------------------+---------------+
|  count|              20640|            20640|             20640|
```

```
20640|             20433|             20640|            20640|
20640|             20640|          20640|
|   mean|-119.56970445736148|
35.6318614341087|28.639486434108527|2635.7630813953488| 537.8705525375618|1425.4
767441860465|499.5396802325581|3.8706710029070246|206855.81690891474|
NULL|
| stddev|  2.003531723502584|2.135952397457101|
12.58555761211163|2181.6152515827944|421.38507007403115|
1132.46212176534|382.3297528316098| 1.899821717945263|115395.61587441359|
NULL|
|    min|            -124.35|            32.54|               1.0|
2.0|               1.0|               3.0|              1.0|            0.4999|
14999.0|      <1H OCEAN|
|    25%|             -121.8|            33.93|              18.0|
1447.0|             296.0|             787.0|             280.0|
2.5625|          119600.0|           NULL|
|    50%|            -118.49|            34.26|              29.0|
2127.0|             435.0|            1166.0|             409.0|
3.5347|          179700.0|           NULL|
|    75%|            -118.01|            37.71|              37.0|
3146.0|             647.0|            1724.0|             605.0|
4.7426|          264700.0|           NULL|
|    max|            -114.31|            41.95|              52.0|
39320.0|            6445.0|           35682.0|            6082.0|
15.0001|          500001.0|     NEAR OCEAN|
+-------+-------------------+-----------------+------------------+--------------
----+------------------+-----------------+----------------+------------------+
------------------+---------------+
```

Sütunlardaki eşşiz değerlerin sayısı incelenmiştir.

[10]:
```
for col_name in data.columns:
    distinct_count = data.select(col_name).distinct().count()
    print(f"{col_name}: {distinct_count} unique values")
```

```
longitude: 844 unique values
latitude: 862 unique values
housing_median_age: 52 unique values
total_rooms: 5926 unique values
total_bedrooms: 1924 unique values
population: 3888 unique values
households: 1815 unique values
median_income: 12928 unique values
median_house_value: 3842 unique values
ocean_proximity: 5 unique values
```

Her sütundaki boş değerler incelelmiştir.

[13]:
```
from pyspark.sql.functions import col, count, when
data.select([count(when(col(c).isNull(), c)).alias(c) for c in data.columns]).
 ↪show()
```

```
+---------+--------+------------------+-----------+-------------+----------+---
-------+-------------+------------------+---------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|hou
seholds|median_income|median_house_value|ocean_proximity|
+---------+--------+------------------+-----------+-------------+----------+---
-------+-------------+------------------+---------------+
|        0|       0|                 0|          0|          207|         0|
0|            0|                 0|              0|
+---------+--------+------------------+-----------+-------------+----------+---
-------+-------------+------------------+---------------+
```

3 3.Özniteliklerin Seçimi ve Verilerin Makine Öğrenmesi İçin Hazırlanması

StringIndexer kullanılarak kategorik değişken olan "ocean_proximity" değişkeni sayısal bir değişkene dönüştürülmüştür.

[15]:
```
from pyspark.ml.feature import StringIndexer, VectorAssembler

indexer =␣
 ↪StringIndexer(inputCol="ocean_proximity",outputCol="ocean_proximity_index")
data = indexer.fit(data).transform(data)
```

[19]:
```
data.show()
```

```
+---------+--------+------------------+-----------+-------------+----------+---
-------+-------------+------------------+---------------+---------------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|hou
seholds|median_income|median_house_value|ocean_proximity|ocean_proximity_index|
+---------+--------+------------------+-----------+-------------+----------+---
-------+-------------+------------------+---------------+---------------------+
|  -122.23|   37.88|              41.0|      880.0|         129.0|     322.0|
126.0|       8.3252|          452600.0|       NEAR BAY|                  3.0|
|  -122.22|   37.86|              21.0|     7099.0|        1106.0|    2401.0|
1138.0|       8.3014|          358500.0|       NEAR BAY|                  3.0|
|  -122.24|   37.85|              52.0|     1467.0|         190.0|     496.0|
177.0|       7.2574|          352100.0|       NEAR BAY|                  3.0|
|  -122.25|   37.85|              52.0|     1274.0|         235.0|     558.0|
219.0|       5.6431|          341300.0|       NEAR BAY|                  3.0|
|  -122.25|   37.85|              52.0|     1627.0|         280.0|     565.0|
259.0|       3.8462|          342200.0|       NEAR BAY|                  3.0|
|  -122.25|   37.85|              52.0|      919.0|         213.0|     413.0|
```

```
193.0|       4.0368|          269700.0|       NEAR BAY|                3.0|
|  -122.25|   37.84|              52.0|     2535.0|         489.0|    1094.0|
514.0|       3.6591|          299200.0|       NEAR BAY|                3.0|
|  -122.25|   37.84|              52.0|     3104.0|         687.0|    1157.0|
647.0|         3.12|          241400.0|       NEAR BAY|                3.0|
|  -122.26|   37.84|              42.0|     2555.0|         665.0|    1206.0|
595.0|       2.0804|          226700.0|       NEAR BAY|                3.0|
|  -122.25|   37.84|              52.0|     3549.0|         707.0|    1551.0|
714.0|       3.6912|          261100.0|       NEAR BAY|                3.0|
|  -122.26|   37.85|              52.0|     2202.0|         434.0|     910.0|
402.0|       3.2031|          281500.0|       NEAR BAY|                3.0|
|  -122.26|   37.85|              52.0|     3503.0|         752.0|    1504.0|
734.0|       3.2705|          241800.0|       NEAR BAY|                3.0|
|  -122.26|   37.85|              52.0|     2491.0|         474.0|    1098.0|
468.0|        3.075|          213500.0|       NEAR BAY|                3.0|
|  -122.26|   37.84|              52.0|      696.0|         191.0|     345.0|
174.0|       2.6736|          191300.0|       NEAR BAY|                3.0|
|  -122.26|   37.85|              52.0|     2643.0|         626.0|    1212.0|
620.0|       1.9167|          159200.0|       NEAR BAY|                3.0|
|  -122.26|   37.85|              50.0|     1120.0|         283.0|     697.0|
264.0|        2.125|          140000.0|       NEAR BAY|                3.0|
|  -122.27|   37.85|              52.0|     1966.0|         347.0|     793.0|
331.0|        2.775|          152500.0|       NEAR BAY|                3.0|
|  -122.27|   37.85|              52.0|     1228.0|         293.0|     648.0|
303.0|       2.1202|          155500.0|       NEAR BAY|                3.0|
|  -122.26|   37.84|              50.0|     2239.0|         455.0|     990.0|
419.0|       1.9911|          158700.0|       NEAR BAY|                3.0|
|  -122.27|   37.84|              52.0|     1503.0|         298.0|     690.0|
275.0|       2.6033|          162900.0|       NEAR BAY|                3.0|
+---------+--------+------------------+-----------+--------------+----------+---
-------+-------------+------------------+---------------+----------------------+
only showing top 20 rows
```

VectorAssembler ile özellikler (bağımsız değişkenler) belirlenerek hepsi bir vektör formatına dönüştürülmüştür. "median_house_value" sütunu da tahmin edilecek değişken olarak belirlenmiştir.(bağımlı değişken)

[20]:
```
assembler = VectorAssembler(
    ␣
  ↪inputCols=["longitude","latitude","housing_median_age","total_rooms","total_bedrooms","popu
    outputCol="features", handleInvalid="skip"
)
data = assembler.transform(data)

final_data= data.select("features","median_house_value")
final_data.show()
```

```
+--------------------+-----------------+
|            features|median_house_value|
+--------------------+-----------------+
|[-122.23,37.88,41…|         452600.0|
|[-122.22,37.86,21…|         358500.0|
|[-122.24,37.85,52…|         352100.0|
|[-122.25,37.85,52…|         341300.0|
|[-122.25,37.85,52…|         342200.0|
|[-122.25,37.85,52…|         269700.0|
|[-122.25,37.84,52…|         299200.0|
|[-122.25,37.84,52…|         241400.0|
|[-122.26,37.84,42…|         226700.0|
|[-122.25,37.84,52…|         261100.0|
|[-122.26,37.85,52…|         281500.0|
|[-122.26,37.85,52…|         241800.0|
|[-122.26,37.85,52…|         213500.0|
|[-122.26,37.84,52…|         191300.0|
|[-122.26,37.85,52…|         159200.0|
|[-122.26,37.85,50…|         140000.0|
|[-122.27,37.85,52…|         152500.0|
|[-122.27,37.85,52…|         155500.0|
|[-122.26,37.84,50…|         158700.0|
|[-122.27,37.84,52…|         162900.0|
+--------------------+-----------------+
only showing top 20 rows
```

4 4.PySpark ile Makine Öğrenmesi Modelinin Oluşturulması

final_data, eğitim ve test verisi olarak ikiye bölünür. Doğrusal Regresyon modeli tanımlanır ve fit fonksiyonu ile model eğitilir.

[21]:
```
from pyspark.ml.regression import LinearRegression

train_data, test_data = final_data.randomSplit([0.8, 0.2])
lr = LinearRegression(featuresCol="features", labelCol="median_house_value")
lr_model = lr.fit(train_data)
```

Model verisi evaluate fonksiyonu ile test verisi üzerinden değerlendirilir. Root mean squared error ve R2 değerleri hesaplanır.

[22]:
```
test_results = lr_model.evaluate(test_data)
print("Root Mean Squared Error (RMSE):", test_results.rootMeanSquaredError)
print("R2:", test_results.r2)
```

```
Root Mean Squared Error (RMSE): 71602.65298060827
R2: 0.6239338349430412
```