MUSTAFA ILKER KAMIS

- YZUP Ara Proje-4 .
    - **SOURCE CODE ==>** = https://github.com/ilkerCoder/Pyspark-Predict-housing.csv/blob/main/house-pricing.ipynb

In [ ]:
```
import pyspark
```

In [ ]:
```
#PySpark ı import etmek ve  and pysparkı kullanabilmek icin sparksession ' u baslatma
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("DatawithPySpark").getOrCreate()
```

In [ ]:
```
""" ŞİMDİ SPARK CALISIYOR ,VERI KONUMUNU VE DIGER DETAYLARI BELIRTEBILIRIZ: """
file_location = "./housing.csv"
file_type = "csv"
#inferSchema seçeneğine True değeri verildiğinde, Spark, veri kümesindeki değerlerin veri
#tiplerini otomatik olarak çıkarmaya çalışır.
infer_schema = "true"
first_row_is_header = "true"
delimiter = ","
```

In [ ]:
```
"""
---
spark.read.format() yöntemi kullanılıyor. Bu yöntem, belirtilen dosya türüne göre uygun bir
okuma biçimi sağlar (örneğin: CSV, JSON, parquet, metin dosyası vb.).
---
"""
df=spark.read.format(file_type).option("inferSchema",infer_schema).option("header", first_row_is_header).option
```

In [ ]:
```
"""
---
VERİLERİMİZİ TABULAR DATA OLARAK GOREBİLMEK İCİN. İLK 5 SATIR GOSTERİLİYOR .
---
"""
df.show(5)
```

```
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+----------
--------+---------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|median_hou
se_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
```

In [ ]:
```
"""
BURADA .DTYPES , COLUMNS GİBİ OZELLİKLERİ DE KULLANABİLİRDİK FAKAT KONUMUZ SPARK OLDUGU İCİN
HEP SPARK BU KONULARDA NASIL OZELLESTİRMELER YAPMIS ONU ARASTIRIP BUNU GOSTERMEK İSTİYORUM.
O YUZDEN PREPROCESSİNG ONCESİ PRİNTSCHEMA İLE BİR VERİ CSV KEŞFİNE DEVAM EDELİM .
"""
df.printSchema()
```

```
root
 |-- longitude: double (nullable = true)
 |-- latitude: double (nullable = true)
 |-- housing_median_age: double (nullable = true)
 |-- total_rooms: double (nullable = true)
 |-- total_bedrooms: double (nullable = true)
 |-- population: double (nullable = true)
 |-- households: double (nullable = true)
 |-- median_income: double (nullable = true)
 |-- median_house_value: double (nullable = true)
 |-- ocean_proximity: string (nullable = true)
```

In [ ]:
```
print("TOPLAM COLUMN SAYISI ==>" , len(df.columns))
print("TOPLAM KAYIT SAYISI ==>" , df.count())
df.describe().show()
```

```
TOPLAM COLUMN SAYISI ==> 10
TOPLAM KAYIT SAYISI ==> 20640
+-------+-------------------+------------------+------------------+------------------+------------------+--------
----------+------------------+------------------+------------------+---------------+
|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|1425.476
7441860465|499.5396802325581|3.8706710029070246|206855.81690891474|           NULL|
| stddev|  2.003531723502584|2.135952397457101| 12.58555761211163|2181.6152515827944|421.38507007403115|  1132.4
6212176534|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|
+-------+-------------------+------------------+------------------+------------------+------------------+--------
----------+------------------+------------------+------------------+---------------+
```

In [ ]:
```python
# ARTIK VERİ İSLEMEYE BASLAYABİLİRİZ . ONCE DUPLİCATE DEGERLERİ SİLEREK BASLAYALIM  :
df.count()
df=df.dropDuplicates()
df.count()
```

Out[ ]:
```
20640
```

In [ ]:
```python
#BURADA TAHMİN EDİLMEK İSTENEN DEGER(label) MEDIAN_HOUSE_VALUE OLDUGU İCİN YENIDEN ADLANIRALIM:
df=df.withColumnRenamed('median_house_value','label')
print(df.show(3))
#LINEAR REGRESSION YAPACAGIMIZ ICIN BIR KORELASYON ANALIZI YAPALIM :
from pyspark.sql.functions import corr

correlations = []

for col in df.columns:
    #OCEAN_PROX. STRING OLDUGU İCİN O VE label HARİC DİGER SUTUNLARIN KORELASYONLARINI ALALIM
    if col != 'label' and col != "ocean_proximity":  # 'label' sütunu hariç
        correlation = df.stat.corr(col, 'label')
        correlations.append((col, correlation))

correlations
#EN GUCLU KORELASYON MEDİAN_İNCOME , TOTAL_ROOMS ,  HOUSİNG_MEDİAN_AGE OLARAK GİDİYOR.
```

```
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|   label|o
cean_proximity|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
|  -122.28|   37.81|              52.0|      340.0|          97.0|     200.0|      87.0|       1.5208|112500.0|
NEAR BAY|
|  -122.13|   37.67|              40.0|     1748.0|         318.0|     914.0|     317.0|       3.8676|184000.0|
NEAR BAY|
|  -122.07|   37.67|              27.0|     3239.0|         671.0|    1469.0|     616.0|       3.2465|230600.0|
NEAR BAY|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
only showing top 3 rows

None
```

Out[ ]:
```
[('longitude', -0.045966615117978296),
 ('latitude', -0.1441602768746604),
 ('housing_median_age', 0.10562341249320963),
 ('total_rooms', 0.13415311380656358),
 ('total_bedrooms', 0.049148219599425405),
 ('population', -0.024649678888894945),
 ('households', 0.06584265057005599),
 ('median_income', 0.6880752079585519)]
```

In [ ]:
```python
#PYSPARK DA PANDAS OZELLİKLERİNİ KULLANABİLMEK İCİN PYSPARK DAKİ TOPANDAS OZELLİGİNİ KULLANALIM.
import pandas as pd
pandas_df = df.toPandas()
print("TOPLAM NULL DEGERLER ------ \n" , pandas_df.isnull().sum())

#ORTALAMA DEGER ILE DOLDURACAGIZ . FONKSİYONUMUZU EKLEYELİM
from pyspark.sql.functions import mean

# TOTAL_BEDROOMS ORTALAMA DEGERINI BUL
mean_value = df.select(mean(df["total_bedrooms"])).collect()[0][0]
```

```python
# ORTALAMA DEGERLE DOLDUR
df_filled = df.fillna(mean_value, subset=["total_bedrooms"])
```

```
TOPLAM NULL DEGERLER ------
 longitude              0
latitude               0
housing_median_age     0
total_rooms            0
total_bedrooms       207
population             0
households             0
median_income          0
label                  0
ocean_proximity        0
dtype: int64
```

In [ ]:

```python
# BIR TANE KATEGORİK DEGİSKENIMIZ VAR(OCEAN_PROXİMİTY). BU ATTRİBUTE ORDİNAL OLMADIGI İCİN
# ONE HOT ENCODING YAPALIM :
print(df_filled.show(5))
from pyspark.ml.feature import OneHotEncoder , StandardScaler , StringIndexer , VectorAssembler
from pyspark.ml import Pipeline
# Pyspark OneHotEncoder STRING DEGERLERLE ISLEM YAPAMADIGI ICIN ONCE INDEXER ILE SAYISAL
#DEGERLERE DONUSTURUP DAHA SONRA ONE HOT ENCODİNG YAPACAGIZ
indexer = StringIndexer(inputCol="ocean_proximity", outputCol="ocean_proximity_index")
encoder = OneHotEncoder(inputCol="ocean_proximity_index", outputCol="ocean_proximity_encoded")
pipeline = Pipeline(stages=[indexer, encoder])

# Pipeline'ı veri üzerinde uygulama
model = pipeline.fit(df_filled)
encoded_df = model.transform(df_filled)
encoded_df = encoded_df.drop("ocean_proximity" , "ocean_proximity_index")
# Sonuçları gösterme
encoded_df.show()
```

```
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|   label|o
cean_proximity|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
|  -122.28|   37.81|              52.0|      340.0|          97.0|     200.0|      87.0|       1.5208|112500.0|
NEAR BAY|
|  -122.13|   37.67|              40.0|     1748.0|         318.0|     914.0|     317.0|       3.8676|184000.0|
NEAR BAY|
|  -122.07|   37.67|              27.0|     3239.0|         671.0|    1469.0|     616.0|       3.2465|230600.0|
NEAR BAY|
|  -122.13|   37.66|              19.0|      862.0|         167.0|     407.0|     183.0|       4.3456|163000.0|
NEAR BAY|
|  -121.85|   39.73|              52.0|      444.0|          80.0|    1107.0|      98.0|       3.4191|137500.0|
INLAND|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
--------------+
only showing top 5 rows

None
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
----------------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|   label|o
cean_proximity_encoded|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
----------------------+
|  -122.28|   37.81|              52.0|      340.0|          97.0|     200.0|      87.0|       1.5208|112500.0|
(4,[3],[1.0])|
|  -122.13|   37.67|              40.0|     1748.0|         318.0|     914.0|     317.0|       3.8676|184000.0|
(4,[3],[1.0])|
|  -122.07|   37.67|              27.0|     3239.0|         671.0|    1469.0|     616.0|       3.2465|230600.0|
(4,[3],[1.0])|
|  -122.13|   37.66|              19.0|      862.0|         167.0|     407.0|     183.0|       4.3456|163000.0|
(4,[3],[1.0])|
|  -121.85|   39.73|              52.0|      444.0|          80.0|    1107.0|      98.0|       3.4191|137500.0|
(4,[1],[1.0])|
|  -121.85|    38.0|              26.0|     3364.0|         570.0|    1806.0|     566.0|       4.2647|133400.0|
(4,[1],[1.0])|
|  -121.87|   37.99|              15.0|     2203.0|         312.0|    1051.0|     311.0|       4.9783|163900.0|
(4,[1],[1.0])|
|  -119.98|   38.93|              25.0|     1262.0|         293.0|     534.0|     226.0|       2.6607| 90400.0|
(4,[1],[1.0])|
|  -119.78|   36.76|              47.0|     1425.0|         323.0|     949.0|     325.0|       1.7344| 51300.0|
(4,[1],[1.0])|
|   -119.6|   36.57|              42.0|     2311.0|         439.0|    1347.0|     436.0|       2.5556| 69700.0|
(4,[1],[1.0])|
|  -124.05|   40.85|              31.0|     2414.0|         428.0|    1005.0|     401.0|       3.5156|143000.0|
(4,[2],[1.0])|
|  -119.77|    36.3|              24.0|     2202.0|         471.0|    1052.0|     439.0|       2.1038| 62000.0|
(4,[1],[1.0])|
|  -122.68|   38.76|              29.0|      994.0|         226.0|     302.0|     117.0|       2.3125| 67900.0|
(4,[1],[1.0])|
|  -118.36|   34.16|              45.0|     1755.0|         335.0|     822.0|     342.0|       5.1423|322900.0|
(4,[0],[1.0])|
|  -118.35|   34.09|              47.0|     1800.0|         546.0|     921.0|     478.0|       2.8021|280600.0|
(4,[0],[1.0])|
|  -118.19|   34.08|              35.0|     1554.0|         381.0|    1487.0|     374.0|       1.9038|139500.0|
(4,[0],[1.0])|
|  -118.33|   34.03|              46.0|     2312.0|         625.0|    1552.0|     603.0|       1.6429|125000.0|
(4,[0],[1.0])|
|  -118.31|   34.02|              46.0|     1976.0|         469.0|    1409.0|     431.0|       2.2981|112100.0|
(4,[0],[1.0])|
|  -118.25|   34.01|              45.0|      782.0|         270.0|    1030.0|     235.0|       1.0898| 93400.0|
(4,[0],[1.0])|
|  -118.28|   33.97|              31.0|     2017.0|         566.0|    2063.0|     521.0|       1.9219|107000.0|
(4,[0],[1.0])|
+---------+--------+------------------+-----------+--------------+----------+----------+-------------+--------+-
----------------------+
only showing top 20 rows
```

In [ ]:

```python
train_df, test_df = encoded_df.randomSplit([.75, .25] , seed=100)
features = ["longitude" ,"latitude" , "housing_median_age", "total_rooms" ,"total_bedrooms", "population" , "ho
from pyspark.ml.regression import LinearRegression

stage_1 = VectorAssembler(inputCols=features, outputCol="out_features")

stage_2 = StandardScaler(inputCol="out_features", outputCol="features")

stage_3 = LinearRegression(featuresCol = 'features', labelCol='label', maxIter=10,
                        regParam=0.8, elasticNetParam=0.1) # It is always a )
```

```
stages = [stage_1, stage_2, stage_3]

pipeline = Pipeline(stages=stages)
model = pipeline.fit(train_df)
pred_result= model.transform(test_df)
pred_result.show(5)
```

```
+---------+--------+------------------+-----------+-------------+----------+----------+-------------+-------+--
--------------------+--------------------+--------------------+------------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|  label|oc
ean_proximity_encoded|        out_features|            features|        prediction|
+---------+--------+------------------+-----------+-------------+----------+----------+-------------+-------+--
--------------------+--------------------+--------------------+------------------+
|  -124.27|   40.69|              36.0|     2349.0|         528.0|    1194.0|     465.0|       2.5179|79000.0|
(4,[2],[1.0])|[-124.27,40.69,36...|[-62.004759084060...|197851.07491546567|
|  -124.21|   40.75|              32.0|     1218.0|         331.0|     620.0|     268.0|       1.6528|58100.0|
(4,[2],[1.0])|[-124.21,40.75,32...|[-61.974821966935...|156920.35540752695|
|  -124.21|   41.75|              20.0|     3810.0|         787.0|    1993.0|     721.0|       2.0074|66900.0|
(4,[2],[1.0])|[-124.21,41.75,20...|[-61.974821966935...|149568.51469413703|
|  -124.19|   40.77|              30.0|     2975.0|         634.0|    1367.0|     583.0|        2.442|69000.0|
(4,[2],[1.0])|[-124.19,40.77,30...|[-61.964842927894...|194327.39536100044|
|  -124.18|   40.78|              37.0|     1453.0|         293.0|     867.0|     310.0|       2.5536|70200.0|
(4,[2],[1.0])|[-124.18,40.78,37...|[-61.959853408373...| 185712.5337922892|
+---------+--------+------------------+-----------+-------------+----------+----------+-------------+-------+--
--------------------+--------------------+--------------------+------------------+
only showing top 5 rows
```

ARTIK TEST DATAMIZDA PREDİCTİON SÜTUNU MEVCUT, DOLAYISIYLA EVALUATE EDEBILIRIZ LINEAR REGRESYON MODELİ R2 VE RMSE ' Yİ KULLANIYOR

In [ ]:

```
from pyspark.ml.evaluation import RegressionEvaluator
regeval = RegressionEvaluator(labelCol="label",
predictionCol="prediction", metricName="rmse")
acc = regeval.evaluate(pred_result, {regeval.metricName: "r2"})
print(acc)
rmse = regeval.evaluate(pred_result)
print(rmse)
```

```
0.6394263010189293
68925.24178667089
```