Proiect Big Data

House Sales in King County

Student: Marin Ana-Maria-Ioana

Grupa: 405

Master Anul 1: Baze de date și tehnologii software

-iunie 2022-

Cuprins

Această lucrare are ca scop prezentarea noțiunilor asimilate în cadrul cursului Big Data utilizând exemple practice. Am ales setul de date „House Sales in King County” care face referire la locuințele vândute în regiunea King Country din S.U.A.

Datele se află în fișierul „ kc_house_data”, ce conține următoarele coloane:

id - ID unic pentru fiecare casă vândută

date - Data vânzării casei

price - Prețul fiecărei case vândute

bedrooms – Număr dormitoare

bathrooms - Număr de băi, unde .5 reprezintă o cameră cu toaletă, dar fără duș

sqft_living - Suprafața în metri pătrați a spațiului de locuit interior al locuințelor

sqft_lot - Metri pătrați a spațiului de teren

floors - Numărul de etaje

waterfront - O variabilă care indică dacă locuința are sau nu vedere la malul mării

view – Priveliștea locuinței pe o scară de la 0 la 4

condition – Starea locuinței pe o scară de la 1 la 5

grade - Un indice de la 1 la 13, unde 1-3 nu corespunde construcției și proiectării clădirii, 7 are un nivel mediu de construcție și proiectare, iar 11-13 au un nivel ridicat de calitate a construcției și proiectării.

sqft_above - Suprafața în metri pătrați a spațiului de locuit interior care este deasupra nivelului solului.

sqft_basement -Suprafața în metri pătrați a spațiului interior de locuințe care se află sub nivelul solului.

yr_built - Anul în care a fost construită locuința

yr_renovated – Anul ultimei renovări

zipcode – Codul poștal

lat - Latitudine

long - Longitudine

sqft_living15 - Suprafața interioară a spațiului de locuit pentru cei mai apropiați 15 vecini

sqft_lot15 -Suprafața terenurilor celor mai apropiați 15 vecini

Procesarea și vizualizarea datelor

Inițializarea contextului Spark

```
from pyspark.sql import SparkSession
spark1 = SparkSession.builder.master("local").appName('Operations').getOrCreate()
```

Încărcarea setului de date într-un dataframe:

```
df = spark1.read.csv('kc_house_data.csv',inferSchema=True,header=True)
```

În continuare, se afișează primele 10 înregistrări din setul de date.

```
df.show(10)
```

```
+----------+---------------+---------+--------+---------+-----------+--------+------+----------+----+---------+-----+----------+-----------
-+--------+------------+-------+-------+--------+-------------+----------+
|        id|           date|    price|bedrooms|bathrooms|sqft_living|sqft_lot|floors|waterfront|view|condition|grade|sqft_above|sqft_basemen
t|yr_built|yr_renovated|zipcode|    lat|    long|sqft_living15|sqft_lot15|
+----------+---------------+---------+--------+---------+-----------+--------+------+----------+----+---------+-----+----------+-----------
-+--------+------------+-------+-------+--------+-------------+----------+
|7129300520|20141013T000000| 221900.0|       3|      1.0|       1180|    5650|   1.0|         0|   0|        3|    7|      1180|
0|    1955|           0|  98178|47.5112|-122.257|         1340|      5650|
|6414100192|20141209T000000| 538000.0|       3|     2.25|       2570|    7242|   2.0|         0|   0|        3|    7|      2170|          40
0|    1951|        1991|  98125| 47.721|-122.319|         1690|      7639|
|5631500400|20150225T000000| 180000.0|       2|      1.0|        770|   10000|   1.0|         0|   0|        3|    6|       770|
0|    1933|           0|  98028|47.7379|-122.233|         2720|      8062|
|2487200875|20141209T000000| 604000.0|       4|      3.0|       1960|    5000|   1.0|         0|   0|        5|    7|      1050|          91
0|    1965|           0|  98136|47.5208|-122.393|         1360|      5000|
|1954400510|20150218T000000| 510000.0|       3|      2.0|       1680|    8080|   1.0|         0|   0|        3|    8|      1680|
0|    1987|           0|  98074|47.6168|-122.045|         1800|      7503|
|7237550310|20140512T000000|1225000.0|       4|      4.5|       5420|  101930|   1.0|         0|   0|        3|   11|      3890|         153
0|    2001|           0|  98053|47.6561|-122.005|         4760|    101930|
|1321400060|20140627T000000| 257500.0|       3|     2.25|       1715|    6819|   2.0|         0|   0|        3|    7|      1715|
0|    1995|           0|  98003|47.3097|-122.327|         2238|      6819|
|2008000270|20150115T000000| 291850.0|       3|      1.5|       1060|    9711|   1.0|         0|   0|        3|    7|      1060|
0|    1963|           0|  98198|47.4095|-122.315|         1650|      9711|
|2414600126|20150415T000000| 229500.0|       3|      1.0|       1780|    7470|   1.0|         0|   0|        3|    7|      1050|          73
0|    1960|           0|  98146|47.5123|-122.337|         1780|      8113|
|3793500160|20150312T000000| 323000.0|       3|      2.5|       1890|    6560|   2.0|         0|   0|        3|    7|      1890|
0|    2003|           0|  98038|47.3684|-122.031|         2390|      7570|
+----------+---------------+---------+--------+---------+-----------+--------+------+----------+----+---------+-----+----------+-----------
-+--------+------------+-------+-------+--------+-------------+----------+
only showing top 10 rows
```

Utilizând comanda "printSchema" se poate vizualiza structura setului de date. Se obțin informații despre tipul de date din fiecare coloană.

```
df.printSchema()
```

```
root
 |-- id: long (nullable = true)
 |-- date: string (nullable = true)
 |-- price: double (nullable = true)
 |-- bedrooms: integer (nullable = true)
 |-- bathrooms: double (nullable = true)
 |-- sqft_living: integer (nullable = true)
 |-- sqft_lot: integer (nullable = true)
 |-- floors: double (nullable = true)
 |-- waterfront: integer (nullable = true)
 |-- view: integer (nullable = true)
 |-- condition: integer (nullable = true)
 |-- grade: integer (nullable = true)
 |-- sqft_above: integer (nullable = true)
 |-- sqft_basement: integer (nullable = true)
 |-- yr_built: integer (nullable = true)
 |-- yr_renovated: integer (nullable = true)
 |-- zipcode: integer (nullable = true)
 |-- lat: double (nullable = true)
 |-- long: double (nullable = true)
 |-- sqft_living15: integer (nullable = true)
 |-- sqft_lot15: integer (nullable = true)
```

Cu ajutorul comenzii „summary” putem obține mai multe informații statistice pentru setul de date ales.

```
df.summary().show()
```

```
|summary|                 id|           date|            price|         bedrooms|        bathrooms|       sqft_living|          sqft_lot|
floors|         waterfront|              view|        condition|            grade|       sqft_above|     sqft_basement|          yr_bui
lt|      yr_renovated|           zipcode|                lat|              long|    sqft_living15|        sqft_lot15|
+-------+-------------------+---------------+-----------------+-----------------+-----------------+------------------+------------------+-----------------+
------------------+--------------------+-------------------+-------------------+-------------------+-------------------+------------------+----
-------------+------------------+------------------+-------------------+-------------------+-------------------+------------------+
|  count|              21613|          21613|            21613|            21613|            21613|             21613|             21613|            21613|
21613|               21613|             21613|              21613|              21613|              21613|              21613|              21613|             2161
3|             21613|             21613|              21613|              21613|              21613|              21613|
|   mean| 4.580301520864988E9|           null| 540088.1417665294| 3.37084162309721|2.1147573219821405|2079.8997362698374|15106.967565816869|
1.4943089807060566|0.007541757275713691|0.23430342849211122|3.4094295100171195| 7.656873178179799|1788.3906907879516| 291.5090454818859|197
1.0051357978994| 84.40225790033776| 98077.93980474715|  47.56005251931704|-122.21389640494083|1986.552491556008|12768.455651691113|
| stddev|2.8765655713120522E9|           null|367127.19648270035|0.930061831147451| 0.770163157217741| 918.4408970468096| 41420.51151513551|
0.5399888951423489| 0.08651719772788748| 0.7663175692736114|0.6507430463662044|1.1754587569743344| 828.0909776519175|442.57504267746685|29.3
73410802386243|401.67924001917504|53.505026257472466|0.13856371024192368|0.14082834238139288|685.3913042527788|27304.179631338524|
|    min|            1000102|20140502T000000|          75000.0|                0|              0.0|               290|                 520|
1.0|                 0|                 0|                1|                1|               290|                 0|              1900|
0|           98001|          47.1559|         -122.519|              399|              651|
|    25%|         2123049175|           null|         321500.0|                3|             1.75|              1425|              5040|
1.0|                 0|                 0|                3|                7|             1190|                 0|              1951|
0|           98033|          47.4709|         -122.328|             1490|             5100|
|    50%|         3904921250|           null|         450000.0|                3|             2.25|              1910|              7617|
1.5|                 0|                 0|                3|                7|             1560|                 0|              1975|
0|           98065|          47.5718|         -122.231|             1840|             7620|
|    75%|         7308600050|           null|         645000.0|                4|              2.5|              2550|             10685|
2.0|                 0|                 0|                4|                8|             2210|               560|              1997|
0|           98118|           47.678|         -122.125|             2360|            10083|
|    max|         9900000190|20150527T000000|        7700000.0|               33|              8.0|             13540|           1651359|
3.5|                 1|                 4|                5|               13|             9410|              4820|              2015|
2015|           98199|          47.7776|         -121.315|             6210|           871200|
```

În continuare, am ales să lucrez cu un număr mai mic de coloane reprezentative pentru setul de date. Am construit un dataframe obținut din alipirea coloanelor de interes din dataframe-ul inițial.

```
df1=df[['price','bedrooms','bathrooms','sqft_living','condition','grade','yr_built']]
```

Afișez primele 10 înregistrări din noul dataframe.

```
df1.show(10)
```

```
+---------+--------+---------+-----------+---------+-----+--------+
|    price|bedrooms|bathrooms|sqft_living|condition|grade|yr_built|
+---------+--------+---------+-----------+---------+-----+--------+
| 221900.0|       3|      1.0|       1180|        3|    7|    1955|
| 538000.0|       3|     2.25|       2570|        3|    7|    1951|
| 180000.0|       2|      1.0|        770|        3|    6|    1933|
| 604000.0|       4|      3.0|       1960|        5|    7|    1965|
| 510000.0|       3|      2.0|       1680|        3|    8|    1987|
|1225000.0|       4|      4.5|       5420|        3|   11|    2001|
| 257500.0|       3|     2.25|       1715|        3|    7|    1995|
| 291850.0|       3|      1.5|       1060|        3|    7|    1963|
| 229500.0|       3|      1.0|       1780|        3|    7|    1960|
| 323000.0|       3|      2.5|       1890|        3|    7|    2003|
+---------+--------+---------+-----------+---------+-----+--------+
only showing top 10 rows
```

Afișez schema datelor. Rezultatul conține denumirea coloanelor și tipul de date ale acestora.

```
df1.printSchema()
```

```
root
 |-- price: double (nullable = true)
 |-- bedrooms: integer (nullable = true)
 |-- bathrooms: double (nullable = true)
 |-- sqft_living: integer (nullable = true)
 |-- condition: integer (nullable = true)
 |-- grade: integer (nullable = true)
 |-- yr_built: integer (nullable = true)
```

Se vizualizează statisticile noului dataframe. Acestea ne ajută să obținem informații esențiale despre datele cu care urmează să lucrăm.

```
df1.summary().show()
```

```
+-------+------------------+-----------------+------------------+-----------------+------------------+-----------------+------------------+
|summary|             price|         bedrooms|         bathrooms|      sqft_living|         condition|            grade|          yr_built|
+-------+------------------+-----------------+------------------+-----------------+------------------+-----------------+------------------+
|  count|             21613|            21613|             21613|            21613|             21613|            21613|             21613|
|   mean| 540088.1417665294| 3.37084162309721|2.1147573219821405|2079.8997362698374|3.4094295100171195| 7.656873178179799|1971.0051357978994|
| stddev|367127.19648270035|0.930061831147451| 0.770163157217741| 918.4408970468096|0.6507430463662044|1.1754587569743344|29.373410802386243|
|    min|           75000.0|                0|               0.0|              290|                 1|                1|              1900|
|    25%|          321500.0|                3|              1.75|             1425|                 3|                7|              1951|
|    50%|          450000.0|                3|              2.25|             1910|                 3|                7|              1975|
|    75%|          645000.0|                4|               2.5|             2550|                 4|                8|              1997|
|    max|         7700000.0|               33|               8.0|            13540|                 5|               13|              2015|
+-------+------------------+-----------------+------------------+-----------------+------------------+-----------------+------------------+
```

Observăm că pentru fiecare coloană se calculează numărul de elemente. Toate coloanele au același număr de elemente, ceea ce inseamnă că nu lipsesc date din celule.

De asemenea, media este un rezultat foarte important. Putem deduce că prețul mediu al unei locuințe este de aproximativ 540.088,142 dolari.

Cu ajutorul informațiilor oferite de valorile maxime și minime calculate pentru fiecare coloană putem deduce dacă există valori aberante care pot afecta procesele de predicție.

Afișarea locuințelor care au prețul mai mic de 200.000,00 dolari și cu un grad mediu de construcție și proiectare.

```
df1.filter((df1['price']<200000)&(df1['grade']>7)).show()
```

```
+--------+--------+---------+-----------+---------+-----+--------+
|   price|bedrooms|bathrooms|sqft_living|condition|grade|yr_built|
+--------+--------+---------+-----------+---------+-----+--------+
|188500.0|       2|     1.75|       1240|        4|    8|    1985|
|189950.0|       2|      1.0|       1030|        3|    8|    1981|
|149900.0|       2|     1.75|       1090|        4|    8|    1982|
|176000.0|       2|      1.0|        920|        4|    8|    1980|
|188000.0|       3|     1.75|       1660|        2|    8|    1979|
|190500.0|       3|      1.5|       1110|        3|    8|    2007|
|199950.0|       3|      3.0|       1530|        3|    8|    1993|
|195000.0|       2|      1.0|       1080|        4|    8|    1972|
|178500.0|       2|      1.0|        930|        4|    8|    1978|
|160000.0|       2|      1.0|       1140|        3|    8|    1980|
|185000.0|       3|      1.5|       1200|        3|    8|    1966|
|168000.0|       2|      1.5|       1220|        4|    8|    1976|
|190000.0|       2|      2.5|       1100|        3|    8|    2006|
|169000.0|       3|     1.75|       1720|        3|    8|    1978|
|194000.0|       1|      1.0|        820|        3|    8|    2007|
|162950.0|       2|      1.0|        950|        4|    8|    1972|
|170000.0|       2|      1.0|       1500|        3|    8|    1950|
|175000.0|       3|     1.75|       1910|        4|    8|    1963|
|187000.0|       2|     1.75|       1050|        4|    8|    1974|
|140000.0|       3|      1.5|       1200|        3|    8|    1966|
+--------+--------+---------+-----------+---------+-----+--------+
```

Afișarea locuințelor construite în anul 2009.

```
df1.filter(df1['yr_built']==2009).show()
```

price	bedrooms	bathrooms	sqft_living	condition	grade	yr_built
266000.0	3	2.5	1805	3	7	2009
930000.0	4	4.0	6050	3	9	2009
320000.0	3	1.5	1240	3	8	2009
359000.0	2	2.75	1370	3	8	2009
363000.0	2	2.0	920	3	8	2009
625000.0	3	3.5	1810	4	8	2009
706000.0	4	2.5	2740	3	9	2009
5350000.0	5	5.0	8000	3	12	2009
411753.0	3	2.5	1710	3	7	2009
369950.0	2	2.75	1370	3	8	2009
275000.0	3	2.25	1260	3	7	2009
247300.0	2	2.0	1140	3	7	2009
579000.0	3	2.5	1640	3	8	2009
750000.0	4	2.5	2680	3	8	2009
912000.0	4	3.75	1980	3	9	2009
305000.0	4	2.0	2470	3	7	2009
656000.0	2	2.5	2270	3	7	2009
455000.0	4	2.5	2811	3	9	2009
440000.0	1	1.0	1160	3	7	2009
435000.0	3	2.25	1380	3	7	2009

Adăugarea și afișarea unei coloane noi care reprezintă vechimea locuinței în ani.

```
df1.withColumn('house_age',2022-df1['yr_built']).show()
```

price	bedrooms	bathrooms	sqft_living	condition	grade	yr_built	house_age
221900.0	3	1.0	1180	3	7	1955	67
538000.0	3	2.25	2570	3	7	1951	71
180000.0	2	1.0	770	3	6	1933	89
604000.0	4	3.0	1960	5	7	1965	57
510000.0	3	2.0	1680	3	8	1987	35
1225000.0	4	4.5	5420	3	11	2001	21
257500.0	3	2.25	1715	3	7	1995	27
291850.0	3	1.5	1060	3	7	1963	59
229500.0	3	1.0	1780	3	7	1960	62
323000.0	3	2.5	1890	3	7	2003	19
662500.0	3	2.5	3560	3	8	1965	57
468000.0	2	1.0	1160	4	7	1942	80
310000.0	3	1.0	1430	4	7	1927	95
400000.0	3	1.75	1370	4	7	1977	45
530000.0	5	2.0	1810	3	7	1900	122
650000.0	4	3.0	2950	3	9	1979	43
395000.0	3	2.0	1890	3	7	1994	28
485000.0	4	1.0	1600	4	7	1916	106
189000.0	2	1.0	1200	4	7	1921	101
230000.0	3	1.0	1250	4	7	1969	53

Adăugarea și afișarea unei coloane noi care reprezintă prețul pentru un metru pătrat din suprafața locuibilă.

```
df1.withColumn('price_mp',df1['price']/df1['sqft_living']).show()
```

price	bedrooms	bathrooms	sqft_living	condition	grade	yr_built	price_mp
221900.0	3	1.0	1180	3	7	1955	188.05084745762713
538000.0	3	2.25	2570	3	7	1951	209.3385214007782
180000.0	2	1.0	770	3	6	1933	233.76623376623377
604000.0	4	3.0	1960	5	7	1965	308.16326530612247
510000.0	3	2.0	1680	3	8	1987	303.57142857142856
1225000.0	4	4.5	5420	3	11	2001	226.01476014760146
257500.0	3	2.25	1715	3	7	1995	150.14577259475217
291850.0	3	1.5	1060	3	7	1963	275.3301886792453
229500.0	3	1.0	1780	3	7	1960	128.93258426966293
323000.0	3	2.5	1890	3	7	2003	170.8994708994709
662500.0	3	2.5	3560	3	8	1965	186.09550561797752
468000.0	2	1.0	1160	4	7	1942	403.44827586206895
310000.0	3	1.0	1430	4	7	1927	216.78321678321677
400000.0	3	1.75	1370	4	7	1977	291.97080291970804
530000.0	5	2.0	1810	3	7	1900	292.81767955801104
650000.0	4	3.0	2950	3	9	1979	220.33898305084745
395000.0	3	2.0	1890	3	7	1994	208.994708994709
485000.0	4	1.0	1600	4	7	1916	303.125
189000.0	2	1.0	1200	4	7	1921	157.5
230000.0	3	1.0	1250	4	7	1969	184.0

Afișarea locuințelor cu un spațiu locuibil mai mic de 310 metri pătrați.

```
df1.createTempView('housepriceKC1')
spark1.sql("SELECT * FROM housepriceKC WHERE sqft_living<310 LIMIT 10").show()
```

price	bedrooms	bathrooms	sqft_living	condition	grade	yr_built
142000.0	0	0.0	290	1	1	1963

Afișarea mediei statistice pentru locuințe grupate în funcție de condiția de locuit.

```
df1.groupby('condition').mean().show()
```

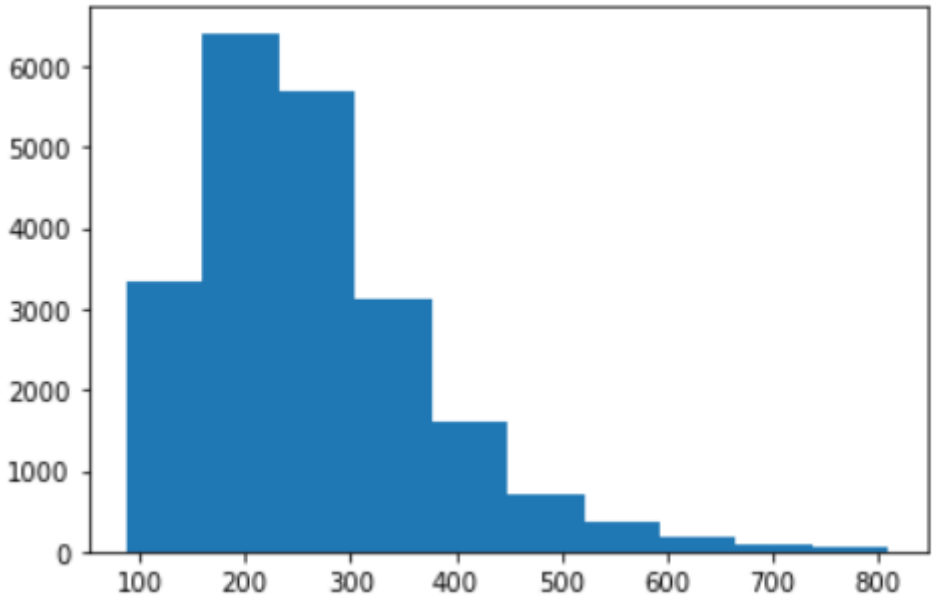
condition	avg(price)	avg(bedrooms)	avg(bathrooms)	avg(sqft_living)	avg(condition)	avg(grade)	avg(yr_built)
1	334431.6666666667	2.466666666666667	1.175	1216.0	1.0	5.8	1931.5333333333333
3	542012.5781483857	3.3741714774428053	2.222632029078469	2149.0420497469886	3.0	7.826740788254579	1979.4631173829378
5	612418.0893592004	3.4603174603174605	2.024397413286302	2022.9112286890065	5.0	7.320987654320987	1946.4485596707818
4	521200.3900334566	3.3569290368022537	1.9004226096143688	1950.991723895052	4.0	7.382461701003698	1958.3402007395669
2	327287.1453488372	2.8313953488372094	1.4491279069767442	1410.0581395348838	2.0	6.505813953488372	1948.9418604651162

Crearea histogramei pentru prețul pe metru pătrat.

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd

df2=pd.read_csv('kc_house_data.csv')

plt.hist(df2['price']/df2['sqft_living']);
```



Regresia liniară

Pornind de la setul de date „House Sales in King County”, se poate observa modul în care atributele unei locuințe influențează prețul de vânzare. Cu ajutorul modelului construit se va putea prezice prețul unei locuințe înainte ca aceasta să fie scoasă la vânzare.

Inițializarea contextului Spark și încărcarea datelor într-un Dataframe.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('linearregression').getOrCreate()
data = spark.read.csv('kc_house_data.csv',inferSchema=True,header=True)
data5=data[['price','bedrooms','bathrooms','sqft_living','condition','grade','yr_built']]
```

Importarea librăriilor utilizate pentru asamblarea coloanelor.

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
```

Crearea unei singure coloane pentru caracteristici, obținută prin alipirea celor 6 coloane care conțin caracteristici numerice.

```
assembler = VectorAssembler(
 inputCols=['bedrooms',
        'bathrooms',
        'sqft_living',
        'condition',
        'grade',
        'yr_built'],
  outputCol="features")
```

Transformarea dataframe-ului.

```
output = assembler.transform(data5)
```

Afișarea dataframe-ului cu coloana de caracteristici adăugată la final.

```
output.show()
```

price	bedrooms	bathrooms	sqft_living	condition	grade	yr_built	features
221900.0	3	1.0	1180	3	7	1955	[3.0,1.0,1180.0,3...
538000.0	3	2.25	2570	3	7	1951	[3.0,2.25,2570.0,...
180000.0	2	1.0	770	3	6	1933	[2.0,1.0,770.0,3....
604000.0	4	3.0	1960	5	7	1965	[4.0,3.0,1960.0,5...
510000.0	3	2.0	1680	3	8	1987	[3.0,2.0,1680.0,3...
1225000.0	4	4.5	5420	3	11	2001	[4.0,4.5,5420.0,3...
257500.0	3	2.25	1715	3	7	1995	[3.0,2.25,1715.0,...
291850.0	3	1.5	1060	3	7	1963	[3.0,1.5,1060.0,3...
229500.0	3	1.0	1780	3	7	1960	[3.0,1.0,1780.0,3...
323000.0	3	2.5	1890	3	7	2003	[3.0,2.5,1890.0,3...
662500.0	3	2.5	3560	3	8	1965	[3.0,2.5,3560.0,3...
468000.0	2	1.0	1160	4	7	1942	[2.0,1.0,1160.0,4...
310000.0	3	1.0	1430	4	7	1927	[3.0,1.0,1430.0,4...
400000.0	3	1.75	1370	4	7	1977	[3.0,1.75,1370.0,...
530000.0	5	2.0	1810	3	7	1900	[5.0,2.0,1810.0,3...
650000.0	4	3.0	2950	3	9	1979	[4.0,3.0,2950.0,3...
395000.0	3	2.0	1890	3	7	1994	[3.0,2.0,1890.0,3...
485000.0	4	1.0	1600	4	7	1916	[4.0,1.0,1600.0,4...
189000.0	2	1.0	1200	4	7	1921	[2.0,1.0,1200.0,4...
230000.0	3	1.0	1250	4	7	1969	[3.0,1.0,1250.0,4...

Crearea și afișarea unui dataframe format din coloana de caracteristici și coloana ce urmează a fi analizată.

```
final_data=output[['features','price']]
final_data.show()
```

features	price
[3.0,1.0,1180.0,3...	221900.0
[3.0,2.25,2570.0,...	538000.0
[2.0,1.0,770.0,3....	180000.0
[4.0,3.0,1960.0,5...	604000.0
[3.0,2.0,1680.0,3...	510000.0
[4.0,4.5,5420.0,3...	1225000.0
[3.0,2.25,1715.0,...	257500.0
[3.0,1.5,1060.0,3...	291850.0
[3.0,1.0,1780.0,3...	229500.0
[3.0,2.5,1890.0,3...	323000.0
[3.0,2.5,3560.0,3...	662500.0
[2.0,1.0,1160.0,4...	468000.0
[3.0,1.0,1430.0,4...	310000.0
[3.0,1.75,1370.0,...	400000.0
[5.0,2.0,1810.0,3...	530000.0
[4.0,3.0,2950.0,3...	650000.0
[3.0,2.0,1890.0,3...	395000.0
[4.0,1.0,1600.0,4...	485000.0
[2.0,1.0,1200.0,4...	189000.0
[3.0,1.0,1250.0,4...	230000.0

Crearea unui obiect de tipul Linear Regression.

```
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(labelCol='price')
```

Construirea modelului de regresie.

```
lrModel = lr.fit(final_data)
```

Afișarea coeficienților de regresie și interceptiei.

```
print("Coefficients: {}".format(str(lrModel.coefficients)))
print('\n')
print("Intercept:{}".format(str(lrModel.intercept)))
```

```
Coefficients: [-48667.690649562515,58972.60582141921,182.6793589009037,17029.908878949656,132864.24949848352,-3926.578459195764]
```

```
Intercept:6863391.045730483
```

Sumarizarea modelului pentru setul de training și afișarea unor metrici

```
trainingSummary = lrModel.summary
```

```
trainingSummary.residuals.show(10)
print("RMSE: {}".format(trainingSummary.rootMeanSquaredError))
print("r2: {}".format(trainingSummary.r2))
```

```
+-------------------+
|          residuals|
+-------------------+
| -74700.80850479566|
| -101947.1884906087|
| -43890.43860881124|
| 100837.73739328142|
|  55873.16741911415|
|  -354732.108840961|
|-53486.884425722994|
| 19097.039326169528|
|-157075.53154935967|
|-3286.2960151694715|
+-------------------+
only showing top 10 rows

RMSE: 227541.91665103188
r2: 0.6158420698333276
```

Regresia logistică

Se dorește crearea unui model pentru predicția locuințelor care au o stare bună din punct de vedere al construcției și proiectării având la bază celelalte atribute ale locuinței.

Inițializarea contextului Spark și încărcarea datelor într-un Dataframe.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('logregconsult').getOrCreate()
datalr = spark.read.csv('kc_house_data.csv',inferSchema=True,header=True)
```

Definirea unui dataframe care urmează a fi utilizat pentru exemplificarea modelului de regresie logistică.

```
datalr2=datalr[['price','bedrooms','bathrooms','sqft_living','grade','yr_built','condition']]
```

Definesc o functie care să încarce în coloana nou creată „good” valoarea 0 dacă o locuință are gradul mai mic decât 7 sau valoarea 1 dacă o locuință are gradul mai mare decât 7.

```
import pyspark.sql.functions as F
datalr2 = datalr2.withColumn('good', F.when(F.col('grade').isin(1,2,3,4,5,6), 0).otherwise(1))
datalr2.show(5)
```

```
+--------+--------+---------+-----------+-----+--------+---------+----+
|   price|bedrooms|bathrooms|sqft_living|grade|yr_built|condition|good|
+--------+--------+---------+-----------+-----+--------+---------+----+
|221900.0|       3|      1.0|       1180|    7|    1955|        3|   1|
|538000.0|       3|     2.25|       2570|    7|    1951|        3|   1|
|180000.0|       2|      1.0|        770|    6|    1933|        3|   0|
|604000.0|       4|      3.0|       1960|    7|    1965|        5|   1|
|510000.0|       3|      2.0|       1680|    8|    1987|        3|   1|
+--------+--------+---------+-----------+-----+--------+---------+----+
only showing top 5 rows
```

```
datalr2.columns
```

```
['price',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'condition',
 'grade',
 'yr_built',
 'price_mp',
 'good']
```

Crearea unei singure coloane pentru caracteristici, obținută prin alipirea celor 7 coloane care conțin caracteristici numerice.

```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=['price',
                                       'bedrooms',
                                       'bathrooms',
                                       'sqft_living',
                                       'yr_built',
                                      'condition'],outputCol='features')
```

```
output = assembler.transform(datalr2)
```

```
final_data = output.select('features','good')
```

```
train_good,test_good = final_data.randomSplit([0.7,0.3])
```

```
from pyspark.ml.classification import LogisticRegression
lr_good = LogisticRegression(labelCol='good')
fitted_good_model = lr_good.fit(train_good)
training_sum = fitted_good_model.summary
```

Se afișează caracteristicile statistice ale modelului obținut.

```
training_sum.predictions.describe().show()
```

```
+-------+------------------+------------------+
|summary|              good|        prediction|
+-------+------------------+------------------+
|  count|             15189|             15189|
|   mean|0.8936730528672066| 0.922048851142274|
| stddev|0.3082657681315293|0.2681035240138647|
|    min|               0.0|               0.0|
|    max|               1.0|               1.0|
+-------+------------------+------------------+
```

Evaluăm rezultatul pe setul de date primit, folosind datele de test.

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
pred_and_labels = fitted_good_model.evaluate(test_good)
pred_and_labels.predictions.show()
```

```
+--------------------+----+--------------------+--------------------+----------+
|            features|good|       rawPrediction|         probability|prediction|
+--------------------+----+--------------------+--------------------+----------+
|[75000.0,1.0,0.0,...|   0|[1.98458898537745...|[0.87916950112534...|       0.0|
|[83000.0,2.0,1.0,...|   0|[3.26539338867429...|[0.96322232995649...|       0.0|
|[84000.0,2.0,1.0,...|   0|[2.14984695305280...|[0.89565447435637...|       0.0|
|[89950.0,1.0,1.0,...|   0|[2.65152680096304...|[0.93410503172731...|       0.0|
|[92000.0,2.0,1.0,...|   0|[2.07237155370779...|[0.88818869625099...|       0.0|
|[95000.0,2.0,1.0,...|   0|[3.06096229316955...|[0.95525344752108...|       0.0|
|[96500.0,3.0,1.0,...|   0|[1.3386535145181,...|[0.79226842518842...|       0.0|
|[99000.0,2.0,1.0,...|   0|[1.44416324683308...|[0.80909852628114...|       0.0|
|[100000.0,2.0,0.7...|   0|[4.13623635451806...|[0.98426854181521...|       0.0|
|[100000.0,2.0,1.0...|   0|[1.31538341049139...|[0.78841260091796...|       0.0|
|[100000.0,4.0,1.0...|   0|[1.76769024761654...|[0.85417019558120...|       0.0|
|[105000.0,3.0,1.0...|   0|[1.26950809101192...|[0.78065852957635...|       0.0|
|[109500.0,2.0,1.0...|   0|[2.09484566082514...|[0.89040119401838...|       0.0|
|[110000.0,1.0,1.0...|   0|[3.23438206733867...|[0.96210783102062...|       0.0|
|[110000.0,2.0,1.0...|   0|[1.87253893747613...|[0.86675178044721...|       0.0|
|[114975.0,2.0,1.0...|   0|[2.15712190035692...|[0.89633241865121...|       0.0|
|[115000.0,1.0,2.0...|   1|[0.11658772113415...|[0.52911395965015...|       0.0|
|[118125.0,2.0,1.0...|   0|[2.11607898087022...|[0.89245617605318...|       0.0|
|[119500.0,3.0,1.0...|   0|[-0.5953080088839...|[0.35541787954524...|       1.0|
|[120000.0,3.0,1.0...|   0|[1.35094766911450...|[0.79428451731651...|       0.0|
+--------------------+----+--------------------+--------------------+----------+
only showing top 20 rows
```

```
good_eval = BinaryClassificationEvaluator(rawPredictionCol='prediction',labelCol='good')
```

```
auc=good_eval.evaluate(pred_and_labels.predictions)
auc
```

```
0.7505639584982591
```

AUC este un rezultat care evaluează cât de bine un model de regresie logistică va clasifica rezultatele pozitive și negative la toate limitele posibile. Poate varia între 0.5 și 1. Cu cât este mai apropiată de 1, cu atât modelul este mai eficient.