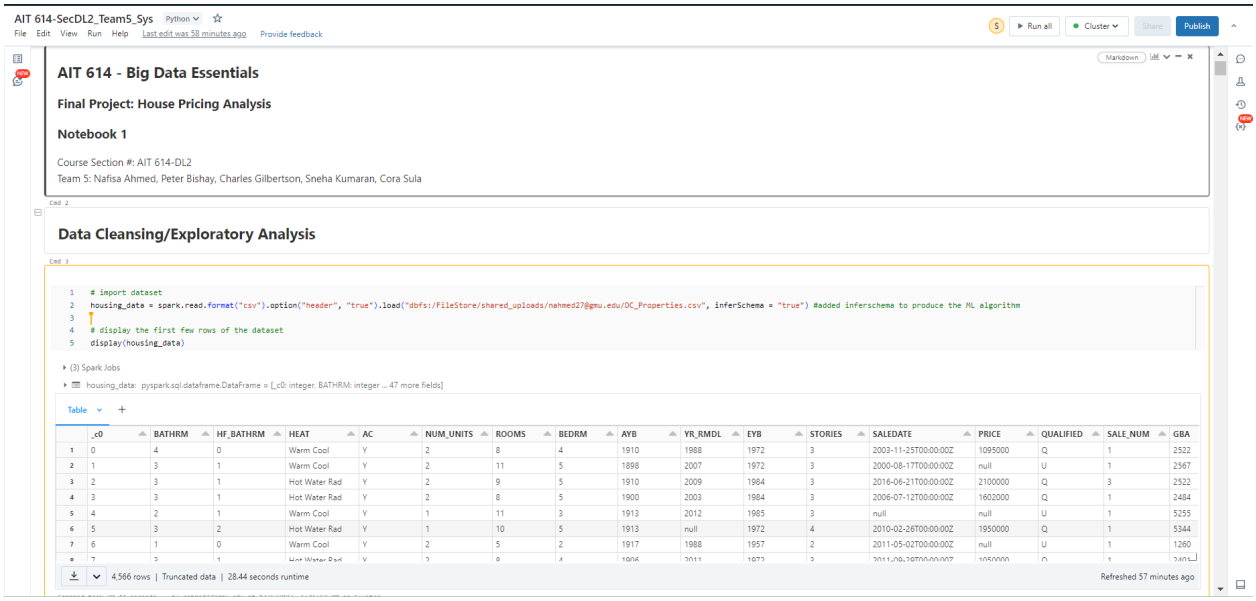
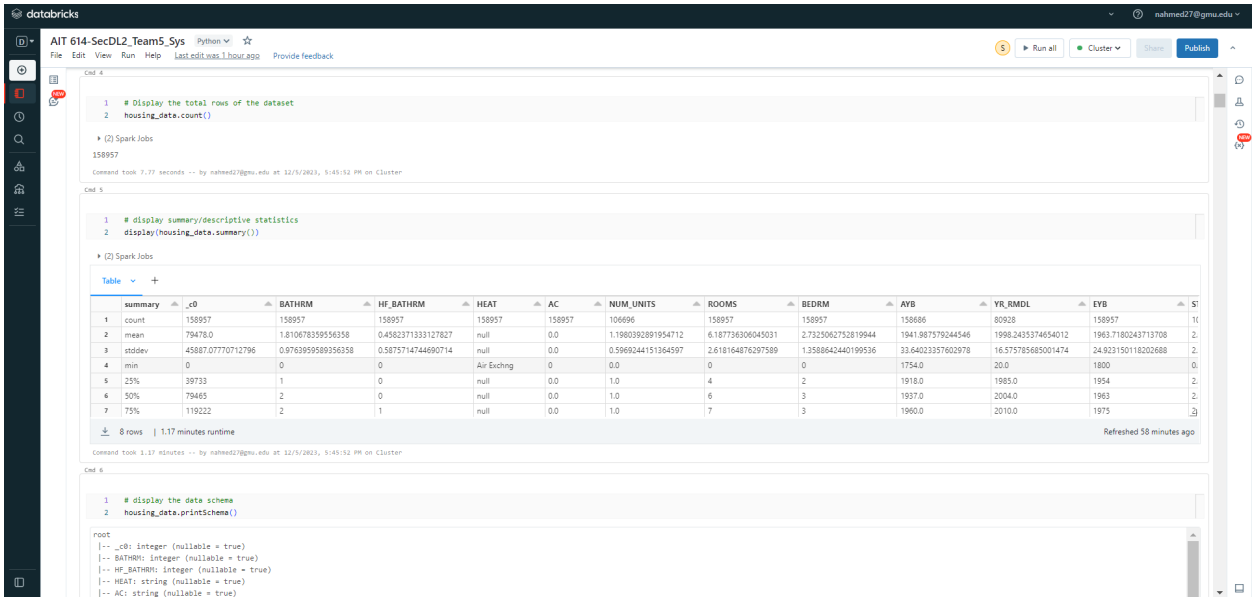
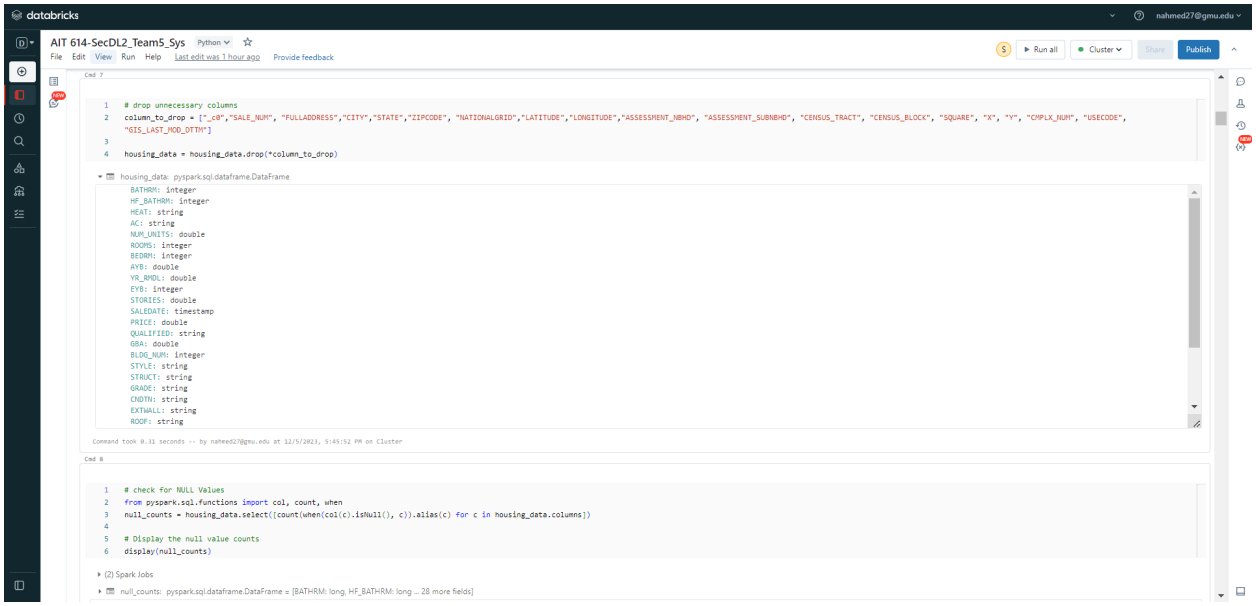
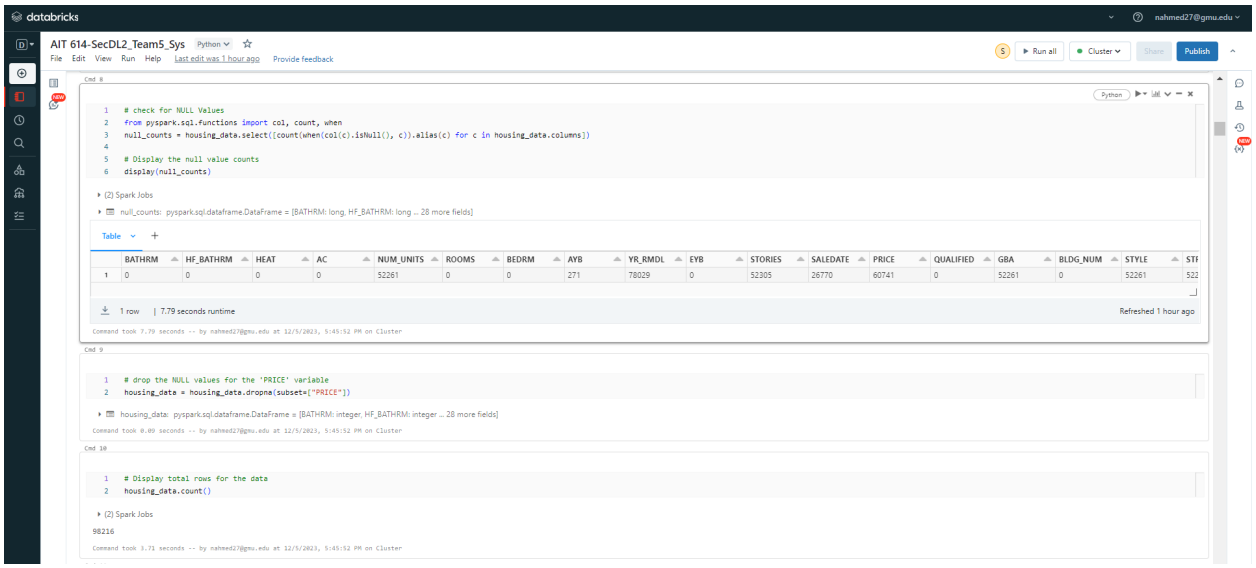
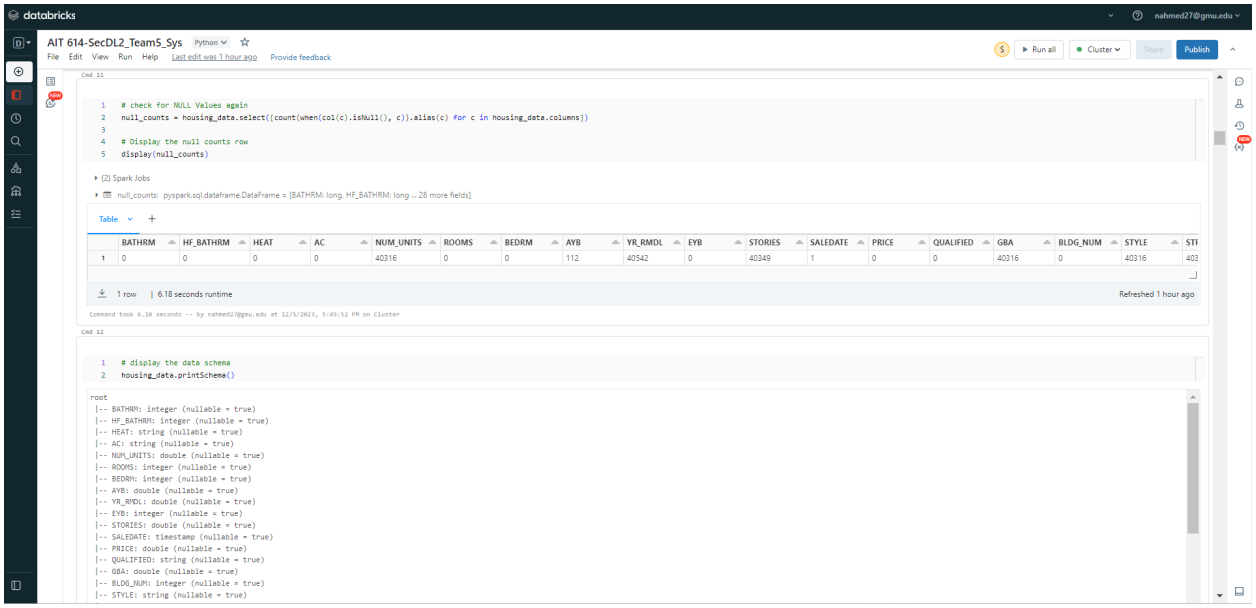
Importing Date/Exploratory Analysis:

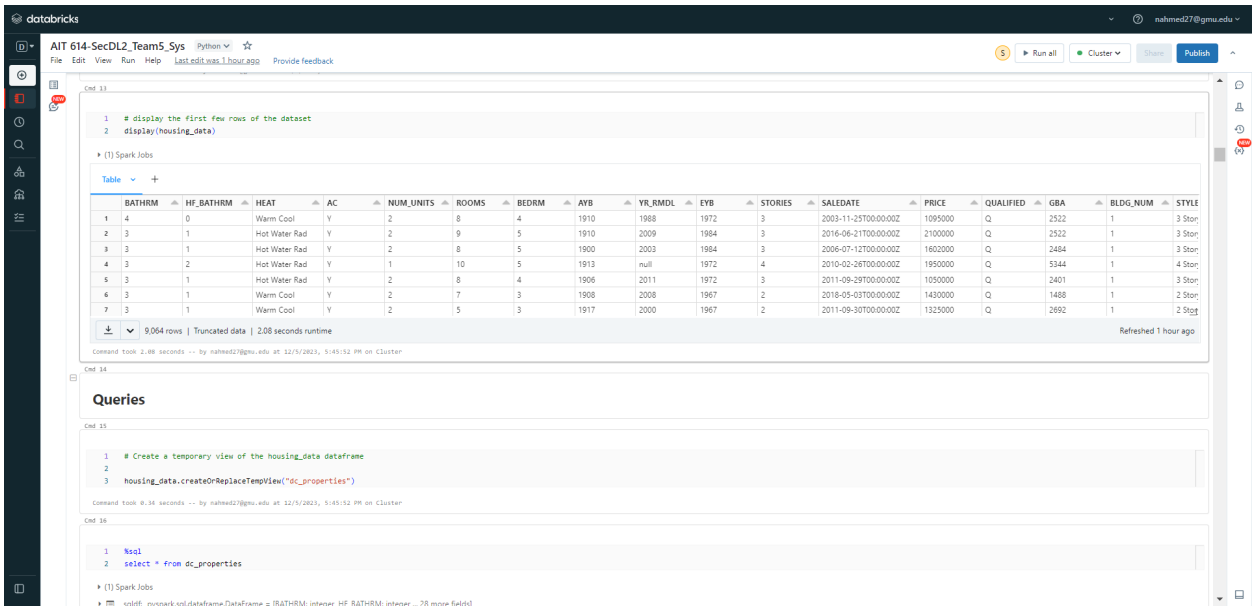




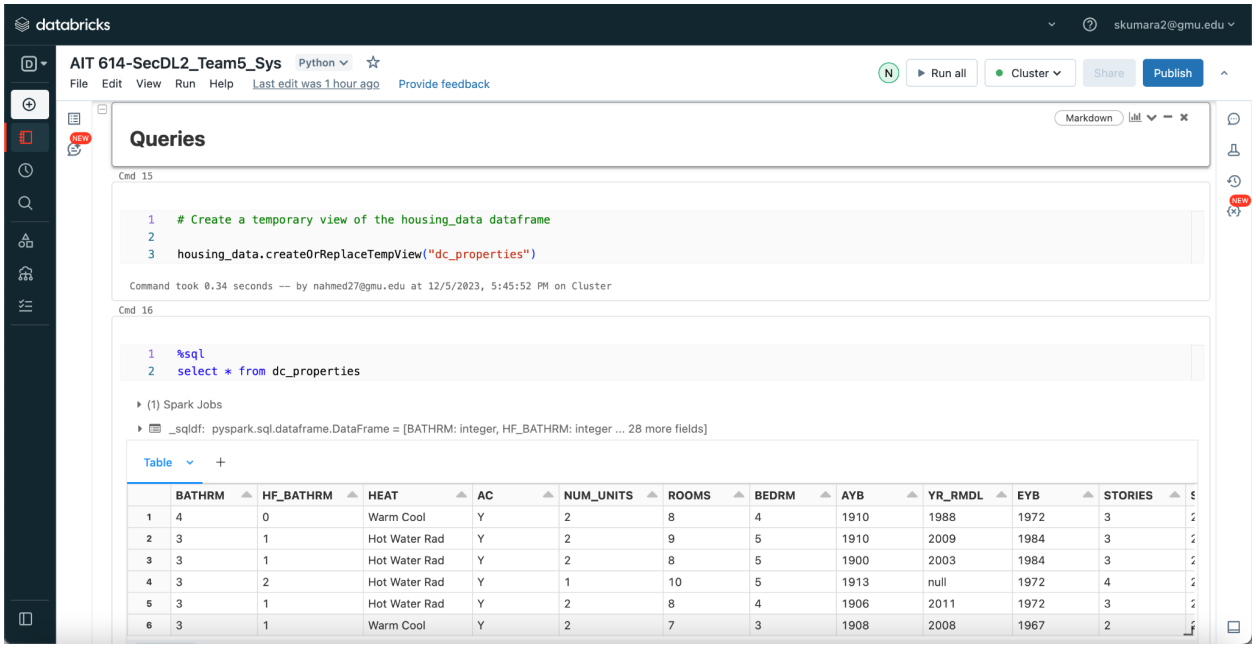

databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 7
# drop unnecessary columns
column_to_drop = ["_c0","SALE_NUM", "FULLADDRESS","CITY","STATE","ZIPCODE", "NATIONALGRID","LATITUDE","LONGITUDE","ASSESSMENT_NBHD", "ASSESSMENT_SUBNBHD", "CENSUS_TRACT", "CENSUS_BLOCK", "SQUARE", "X", "Y", "CMPLX_NUM", "USECODE", "GIS_LAST_MOD_DTTM"]
housing_data = housing_data.drop(*column_to_drop)
housing_data: pyspark.sql.dataframe.DataFrame
BATHRM: integer
HF_BATHRM: integer
HEAT: string
AC: string
NUM_UNITS: double
ROOMS: integer
BEDRM: integer
AYB: double
YR_RMDL: double
EYB: integer
STORIES: double
SALEDATE: timestamp
PRICE: double
QUALIFIED: string
GBA: double
BLDG_NUM: integer
STYLE: string
STRUCT: string
GRADE: string
CNDTN: string
EXTWALL: string
ROOF: string
Command took 0.31 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 8
# check for NULL Values
from pyspark.sql.functions import col, count, when
null_counts = housing_data.select([count(when(col(c).isNull(), c)).alias(c) for c in housing_data.columns])
# Display the null value counts
display(null_counts)
(2) Spark Jobs
null_counts: pyspark.sql.dataframe.DataFrame = [BATHRM: long, HF_BATHRM: long ... 28 more fields]

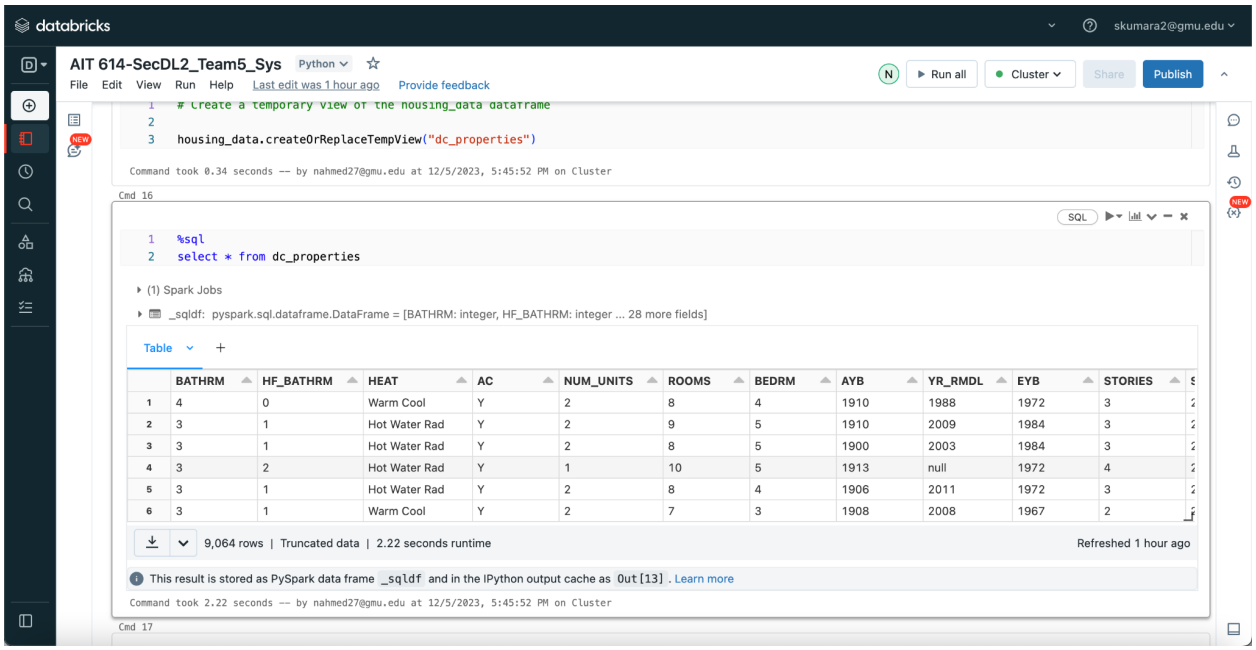

databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 8
# check for NULL Values
from pyspark.sql.functions import col, count, when
null_counts = housing_data.select([count(when(col(c).isNull(), c)).alias(c) for c in housing_data.columns])
# Display the null value counts
display(null_counts)
(2) Spark Jobs
null_counts: pyspark.sql.dataframe.DataFrame = [BATHRM: long, HF_BATHRM: long ... 28 more fields]
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 0 0 52261 0 0 271 78029 0 52305 26770 60741 0 52261 0 52261
1 row | 7.79 seconds runtime
Refreshed 1 hour ago
Command took 7.79 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 9
# drop the NULL values for the 'PRICE' variable
housing_data = housing_data.dropna(subset=["PRICE"])
housing_data: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 28 more fields]
Command took 0.09 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 10
# Display total rows for the data
housing_data.count()
(2) Spark Jobs
98216
Command took 3.71 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

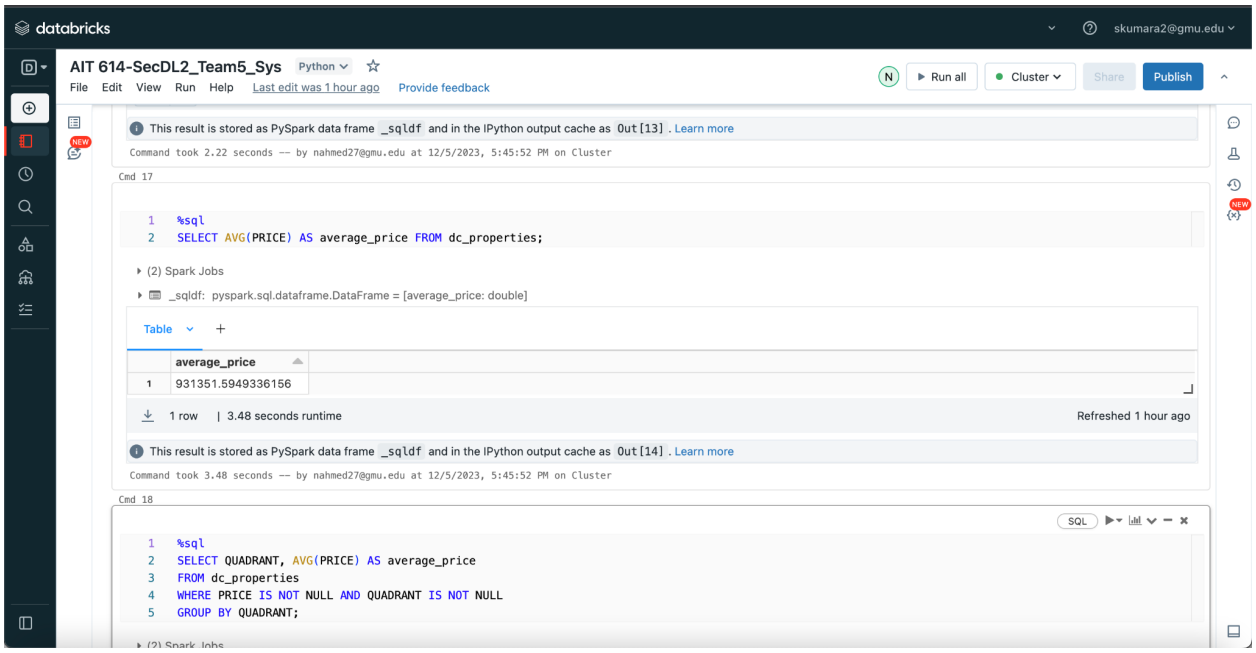

databricks
AIT 614-SecDL2_Team5_Sys
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all Cluster Publish
Cmd 11
# check for NULL Values again
null_counts = housing_data.select([count(when(col(c).isNull(), c)).alias(c) for c in housing_data.columns])
# Display the null counts row
display(null_counts)
(2) Spark Jobs
null_counts: pyspark.sql.dataframe.DataFrame = [BATHRM: long, HF_BATHRM: long ... 28 more fields]
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 0 0 40316 0 0 112 40542 0 40349 1 0 0 40316 0 40316
1 row | 6.18 seconds runtime
Refreshed 1 hour ago
Command took 6.18 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 12
# display the data schema
housing_data.printSchema()
root
|-- BATHRM: integer (nullable = true)
|-- HF_BATHRM: integer (nullable = true)
|-- HEAT: string (nullable = true)
|-- AC: string (nullable = true)
|-- NUM_UNITS: double (nullable = true)
|-- ROOMS: integer (nullable = true)
|-- BEDRM: integer (nullable = true)
|-- AYB: double (nullable = true)
|-- YR_RMDL: double (nullable = true)
|-- EYB: integer (nullable = true)
|-- STORIES: double (nullable = true)
|-- SALEDATE: timestamp (nullable = true)
|-- PRICE: double (nullable = true)
|-- QUALIFIED: string (nullable = true)
|-- GBA: double (nullable = true)
|-- BLDG_NUM: integer (nullable = true)
|-- STYLE: string (nullable = true)

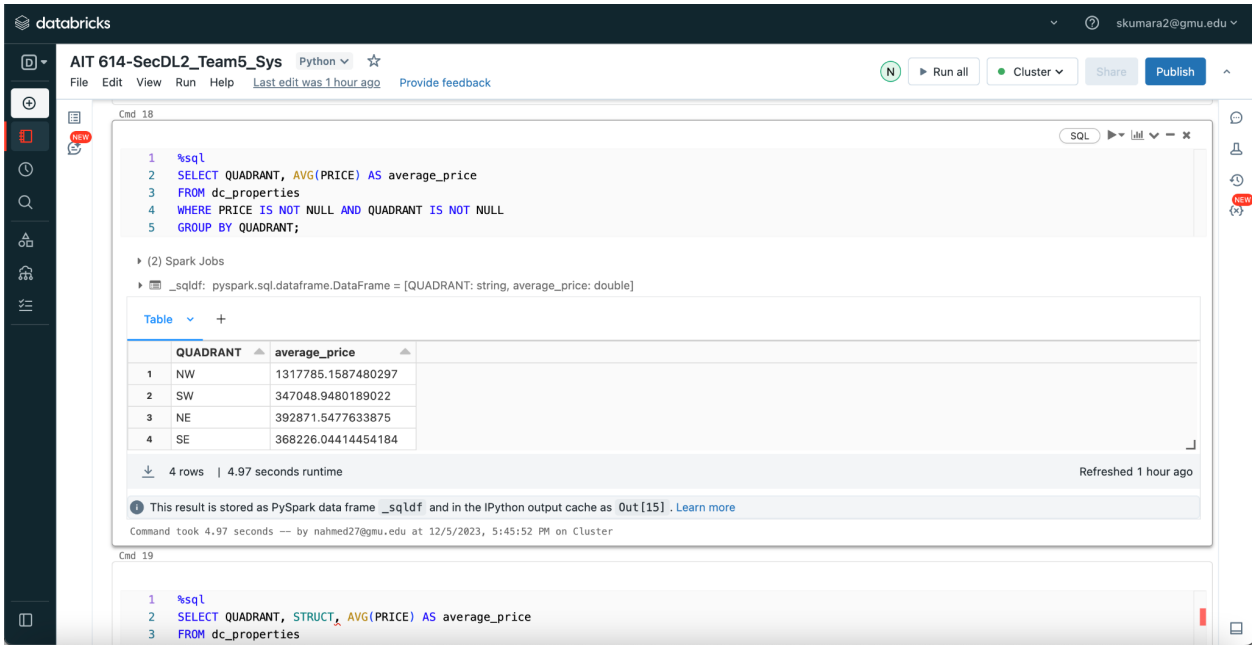

databricks
AIT 614-SecDL2_Team5_Sys
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all Cluster Publish
Cmd 13
# display the first few rows of the dataset
display(housing_data)
(1) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 4 0 Warm Cool Y 2 8 4 1910 1988 1972 3 2003-11-25T00:00:00Z 1095000 Q 2522 1 3 Stor
2 3 1 Hot Water Rad Y 2 9 5 1910 2009 1984 3 2016-06-21T00:00:00Z 2100000 Q 2522 1 3 Stor
3 3 1 Hot Water Rad Y 2 8 5 1900 2003 1984 3 2006-07-12T00:00:00Z 1602000 Q 2484 1 3 Stor
4 3 2 Hot Water Rad Y 1 10 5 1913 null 1972 4 2010-02-26T00:00:00Z 1950000 Q 5344 1 4 Stor
5 3 1 Hot Water Rad Y 2 8 4 1906 2011 1972 3 2011-09-29T00:00:00Z 1050000 Q 2401 1 3 Stor
6 3 1 Warm Cool Y 2 7 3 1908 2008 1967 2 2018-05-03T00:00:00Z 1430000 Q 1488 1 2 Stor
7 3 1 Warm Cool Y 2 5 3 1917 2000 1967 2 2011-09-30T00:00:00Z 1325000 Q 2692 1 2 Stor
9,064 rows | Truncated data | 2.08 seconds runtime
Refreshed 1 hour ago
Command took 2.08 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 14
Queries
Cmd 15
# Create a temporary view of the housing_data dataframe
housing_data.createOrReplaceTempView("dc_properties")
Command took 0.34 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 16
%sql
select * from dc_properties
(1) Spark Jobs

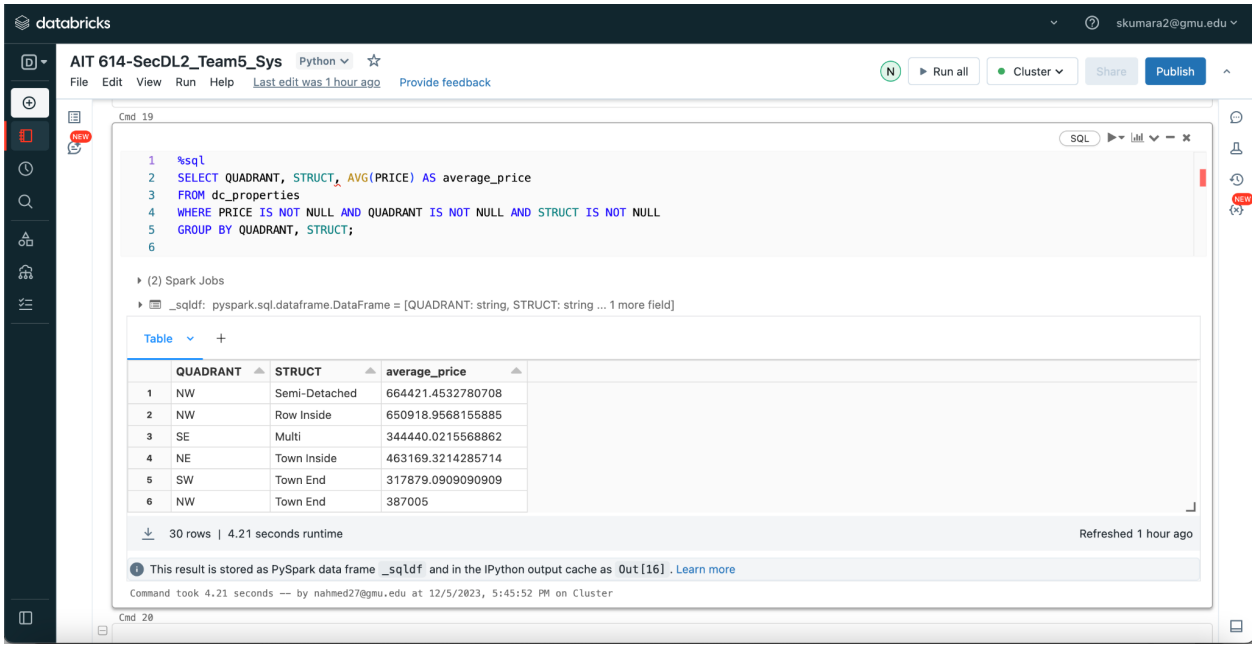
Queries:


databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
Markdown
Queries
Cmd 15
1 # Create a temporary view of the housing_data dataframe
2
3 housing_data.createOrReplaceTempView("dc_properties")
Command took 0.34 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 16
1 %sql
2 select * from dc_properties
(1) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 28 more fields]
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES
1 4 0 Warm Cool Y 2 8 4 1910 1988 1972 3
2 3 1 Hot Water Rad Y 2 9 5 1910 2009 1984 3
3 3 1 Hot Water Rad Y 2 8 5 1900 2003 1984 3
4 3 2 Hot Water Rad Y 1 10 5 1913 null 1972 4
5 3 1 Hot Water Rad Y 2 8 4 1906 2011 1972 3
6 3 1 Warm Cool Y 2 7 3 1908 2008 1967 2

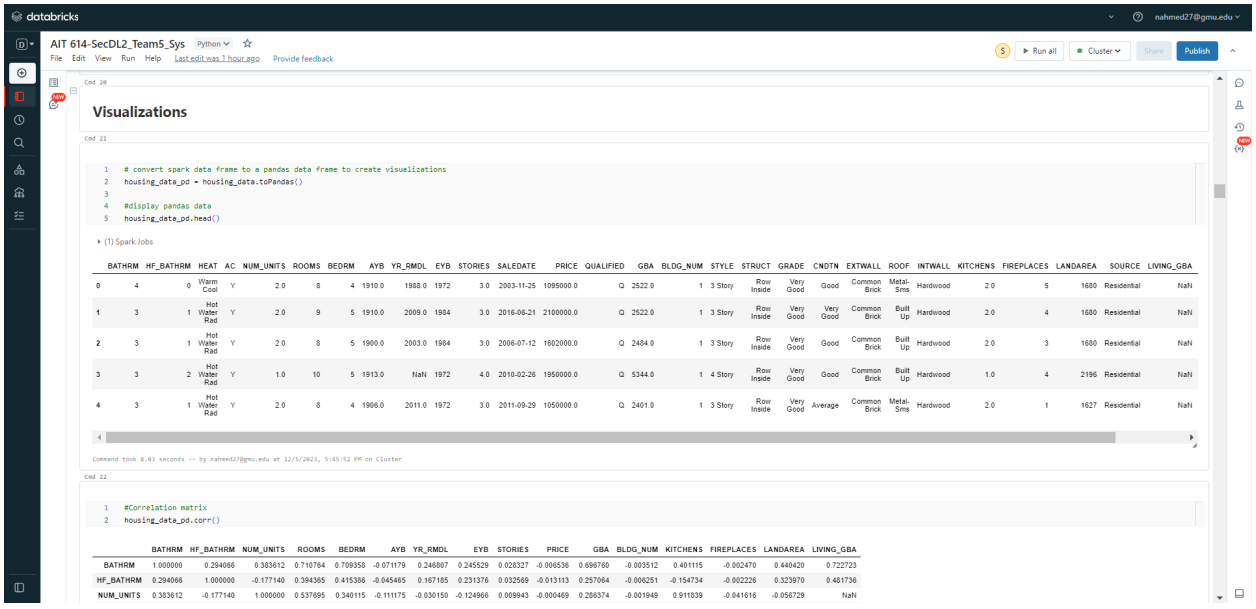

databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
# Create a temporary view of the housing_data dataframe
housing_data.createOrReplaceTempView("dc_properties")
Command took 0.34 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 16
SQL
%sql
select * from dc_properties
(1) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 28 more fields]
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES
1 4 0 Warm Cool Y 2 8 4 1910 1988 1972 3
2 3 1 Hot Water Rad Y 2 9 5 1910 2009 1984 3
3 3 1 Hot Water Rad Y 2 8 5 1900 2003 1984 3
4 3 2 Hot Water Rad Y 1 10 5 1913 null 1972 4
5 3 1 Hot Water Rad Y 2 8 4 1906 2011 1972 3
6 3 1 Warm Cool Y 2 7 3 1908 2008 1967 2
9,064 rows | Truncated data | 2.22 seconds runtime
Refreshed 1 hour ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[13] . Learn more
Command took 2.22 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 17


databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[13] . Learn more
Command took 2.22 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 17
%sql
SELECT AVG(PRICE) AS average_price FROM dc_properties;
(2) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [average_price: double]
Table
average_price
1 931351.5949336156
1 row | 3.48 seconds runtime
Refreshed 1 hour ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[14] . Learn more
Command took 3.48 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 18
SQL
%sql
SELECT QUADRANT, AVG(PRICE) AS average_price
FROM dc_properties
WHERE PRICE IS NOT NULL AND QUADRANT IS NOT NULL
GROUP BY QUADRANT;
(2) Spark Jobs


databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
Cmd 18
SQL
%sql
SELECT QUADRANT, AVG(PRICE) AS average_price
FROM dc_properties
WHERE PRICE IS NOT NULL AND QUADRANT IS NOT NULL
GROUP BY QUADRANT;
(2) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [QUADRANT: string, average_price: double]
Table
QUADRANT average_price
1 NW 1317785.1587480297
2 SW 347048.9480189022
3 NE 392871.5477633875
4 SE 368226.04414454184
4 rows | 4.97 seconds runtime
Refreshed 1 hour ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[15] . Learn more
Command took 4.97 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 19
%sql
SELECT QUADRANT, STRUCT, AVG(PRICE) AS average_price
FROM dc_properties



Visualizations:



databricks

nahmed27@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

Cmd 22

```
#Correlation matrix
housing_data_pd.corr()
```

	BATHRM	HF_BATHRM	NUM_UNITS	ROOMS	BEDRM	AYB	YR_RMDL	EYB	STORIES	PRICE	GBA	BLDG_NUM	KITCHENS	FIREPLACES	LANDAREA	LIVING_GBA
BATHRM	1.000000	0.294066	0.383612	0.710764	0.709358	-0.071179	0.246807	0.245529	0.028327	-0.006536	0.696760	-0.003512	0.401115	-0.002470	0.440420	0.722723
HF_BATHRM	0.294066	1.000000	-0.177140	0.394365	0.415386	-0.045465	0.167185	0.231376	0.032569	-0.013113	0.257064	-0.006251	-0.154734	-0.002226	0.323970	0.481736
NUM_UNITS	0.383612	-0.177140	1.000000	0.537695	0.340115	-0.111175	-0.030150	-0.124966	0.009943	-0.000469	0.286374	-0.001949	0.911839	-0.041616	-0.056729	NaN
ROOMS	0.710764	0.394365	0.537695	1.000000	0.847760	-0.225328	0.202394	0.164268	0.025786	-0.038241	0.680618	-0.004819	0.528950	-0.004489	0.514334	0.759645
BEDRM	0.709358	0.415386	0.340115	0.847760	1.000000	-0.248506	0.253108	0.170732	0.026312	-0.043331	0.645725	-0.007643	0.352900	-0.005711	0.514365	0.717004
AYB	-0.071179	-0.045465	-0.111175	-0.225328	-0.248506	1.000000	0.003346	0.684361	-0.004003	0.056041	0.035662	-0.007500	-0.118075	0.001507	-0.090143	0.115434
YR_RMDL	0.246807	0.167185	-0.030150	0.202394	0.253108	0.003346	1.000000	0.256893	-0.003363	0.030072	-0.019663	-0.000731	-0.013625	0.001339	0.118823	0.122423
EYB	0.245529	0.231376	-0.124966	0.164268	0.170732	0.684361	0.256893	1.000000	0.033950	0.049627	0.284936	0.000756	-0.102144	-0.000198	0.116174	0.106563
STORIES	0.028327	0.032569	0.009943	0.025786	0.026312	-0.004003	-0.003363	0.033950	1.000000	0.042337	0.042322	-0.000702	0.011565	0.027901	-0.014466	NaN
PRICE	-0.006536	-0.013113	-0.000469	-0.038241	-0.043331	0.056041	0.030072	0.049627	0.042337	1.000000	0.600576	0.003048	0.045213	0.023989	-0.005457	-0.027106
GBA	0.696760	0.257064	0.286374	0.680618	0.645725	0.035662	-0.019663	0.284936	0.042322	0.600576	1.000000	-0.009831	0.296983	0.502694	0.494548	NaN
BLDG_NUM	-0.003512	-0.006251	-0.001949	-0.004819	-0.007643	-0.007500	-0.000731	0.000756	-0.000702	0.003048	-0.009831	1.000000	-0.002541	-0.000045	0.040238	NaN
KITCHENS	0.401115	-0.154734	0.911839	0.528950	0.352900	-0.118075	-0.013625	-0.102144	0.011565	0.045213	0.296983	-0.002541	1.000000	-0.015238	-0.039189	NaN
FIREPLACES	-0.002470	-0.002226	-0.041616	-0.004489	-0.005711	0.001507	0.001339	-0.000198	0.027901	0.023989	0.502694	-0.000045	-0.015238	1.000000	-0.001991	-0.004179
LANDAREA	0.440420	0.323970	-0.056729	0.514334	0.514365	-0.090143	0.118823	0.116174	-0.014466	-0.005457	0.494548	0.040238	-0.039189	-0.001991	1.000000	0.455107
LIVING_GBA	0.722723	0.481736	NaN	0.759645	0.717004	0.115434	0.122423	0.106563	NaN	-0.027106	NaN	NaN	NaN	-0.004179	0.455107	1.000000

Command took 0.19 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

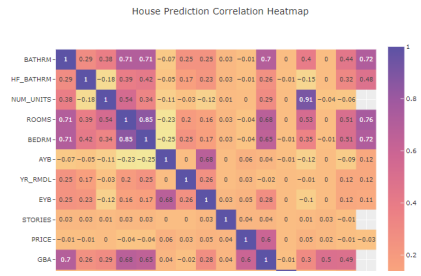
nahmed27@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

```
# Correlation Heat Map
import plotly.express as px

fig_cm = px.imshow(
    housing_data_pd.corr().round(2),
    color_continuous_scale= 'sunset',
    text_auto=True,
    template= 'ggplot2',
    title= "House Prediction Correlation Heatmap"
)

# Make the figure bigger
fig_cm.update_layout(
    height=800,  # set the height in pixels
    width=800    # set the width in pixels
)

# Show the figure
fig_cm.show()
```



databricks

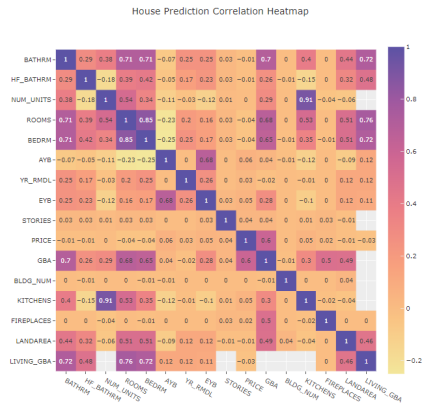
nahmed27@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

```
    width=800    # set the width in pixels
)

# Show the figure
fig_cm.show()
```



Command took 3.40 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

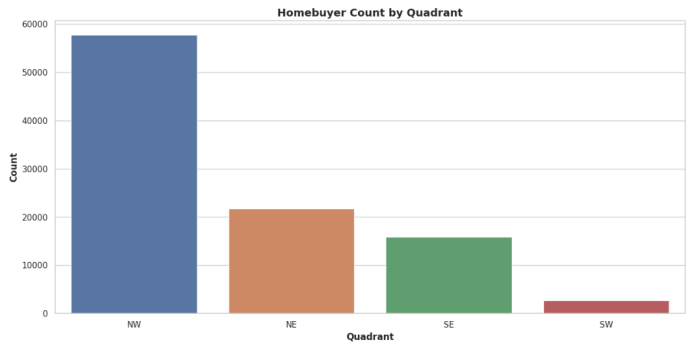
nahmed27@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

```python
# Visualize which Quadrant in DC had the most home buyers
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(style="whitegrid")
plt.figure(figsize=(15, 7))
sns.countplot(data=housing_data_pd, x='QUADRANT', order=housing_data_pd['QUADRANT'].value_counts(ascending=False).index)

plt.xlabel( "Quadrant" , size = 12, weight='bold') # Set label for x-axis

plt.ylabel( "Count" , size = 12, weight='bold') # Set label for y-axis

plt.title( "Homebuyer Count by Quadrant" , size = 14, weight='bold') # Set title for plot

plt.show()
```



Command took 4.60 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

nahmed27@gmu.edu

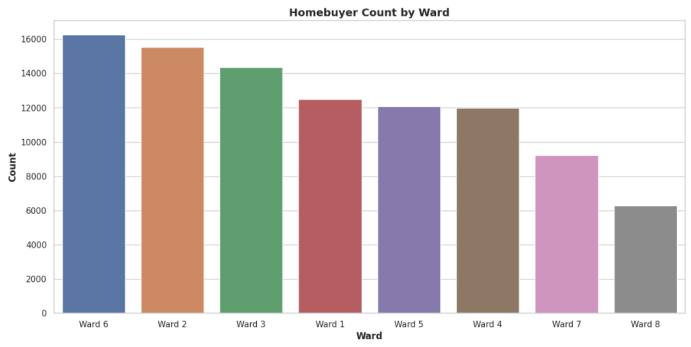
AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

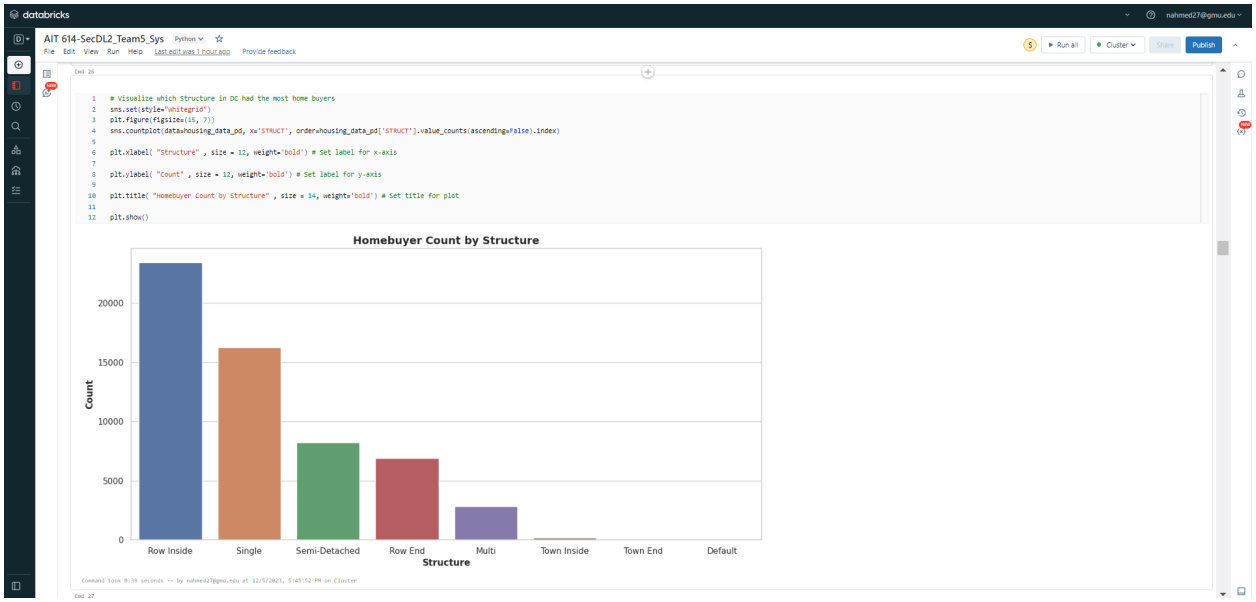
Cmd 25

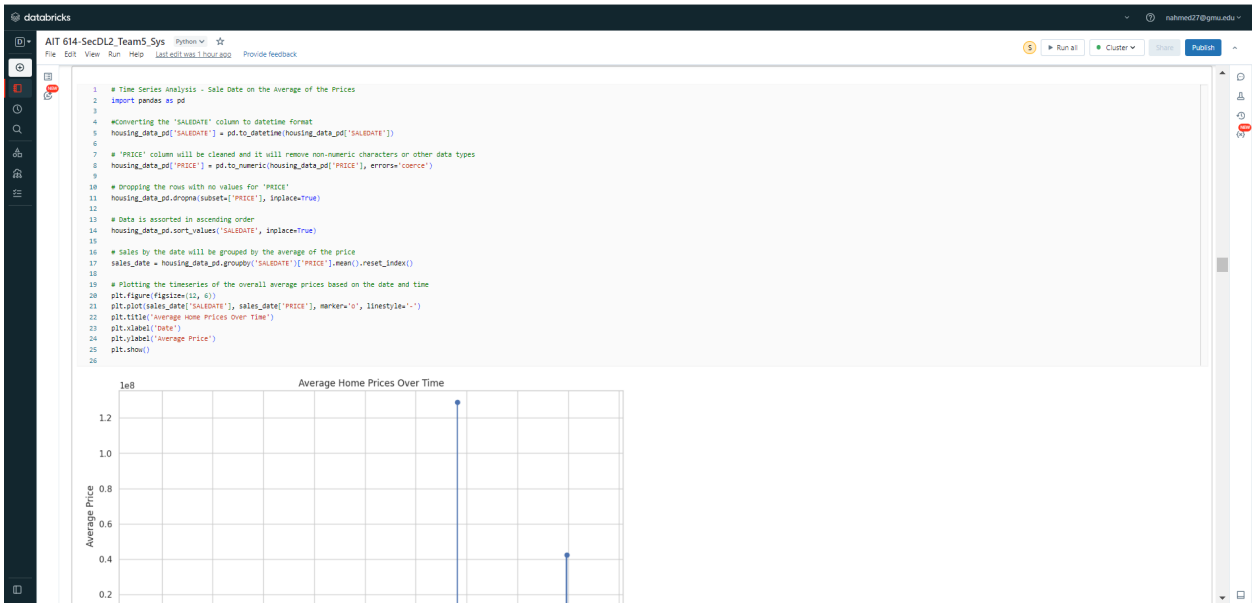
```python
# Visualize which Ward in DC had the most home buyers
sns.set(style="whitegrid")
plt.figure(figsize=(15, 7))
sns.countplot(data=housing_data_pd, x='WARD', order=housing_data_pd['WARD'].value_counts(ascending=False).index)

plt.xlabel( "Ward" , size = 12, weight='bold') # Set label for x-axis

plt.ylabel( "Count" , size = 12, weight='bold') # Set label for y-axis

plt.title( "Homebuyer Count by Ward" , size = 14, weight='bold') # Set title for plot

plt.show()
```

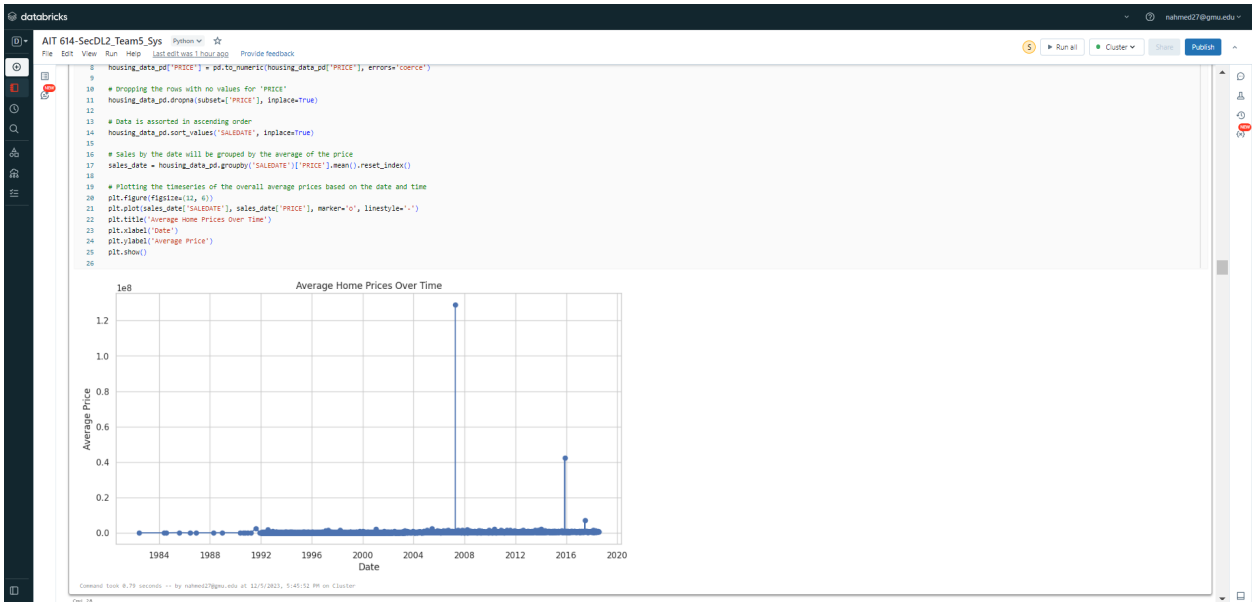


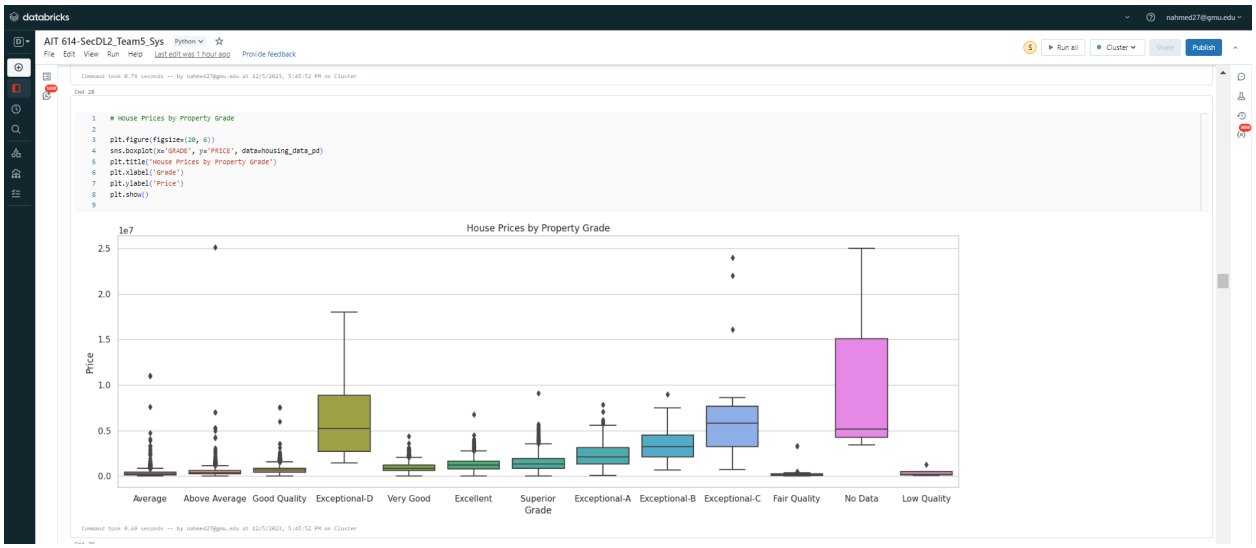
Command took 0.49 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

Cmd 26


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
Cmd 26
1 # Visualize which Structure in OC had the most home buyers
2 sns.set(style="whitegrid")
3 plt.figure(figsize=(15, 7))
4 sns.countplot(data=housing_data_pd, x='STRUCT', order=housing_data_pd['STRUCT'].value_counts(ascending=False).index)
5
6 plt.xlabel("Structure" , size = 12, weight='bold') # Set label for x-axis
7
8 plt.ylabel("Count" , size = 12, weight='bold') # Set label for y-axis
9
10 plt.title("Homebuyer Count by Structure" , size = 14, weight='bold') # Set title for plot
11
12 plt.show()
Homebuyer Count by Structure
Count
20000
15000
10000
5000
0
Row Inside
Single
Semi-Detached
Row End
Multi
Town Inside
Town End
Default
Structure
Command took 0.39 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 27


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help
Last edit was 1 hour ago
Provide feedback
Run all
Cluster
Share
Publish
1 # Time Series Analysis - Sale Date on the Average of the Prices
2 import pandas as pd
3
4 #Converting the 'SALEDATE' column to datetime format
5 housing_data_pd['SALEDATE'] = pd.to_datetime(housing_data_pd['SALEDATE'])
6
7 # 'PRICE' column will be cleaned and it will remove non-numeric characters or other data types
8 housing_data_pd['PRICE'] = pd.to_numeric(housing_data_pd['PRICE'], errors='coerce')
9
10 # Dropping the rows with no values for 'PRICE'
11 housing_data_pd.dropna(subset=['PRICE'], inplace=True)
12
13 # Data is assorted in ascending order
14 housing_data_pd.sort_values('SALEDATE', inplace=True)
15
16 # Sales by the date will be grouped by the average of the price
17 sales_date = housing_data_pd.groupby('SALEDATE')['PRICE'].mean().reset_index()
18
19 # Plotting the timeseries of the overall average prices based on the date and time
20 plt.figure(figsize=(12, 6))
21 plt.plot(sales_date['SALEDATE'], sales_date['PRICE'], marker='o', linestyle='-')
22 plt.title('Average Home Prices Over Time')
23 plt.xlabel('Date')
24 plt.ylabel('Average Price')
25 plt.show()
26
Average Home Prices Over Time
1e8
1.2
1.0
0.8
0.6
0.4
0.2
Average Price


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
housing_data_pd['PRICE'] = pd.to_numeric(housing_data_pd['PRICE'], errors='coerce')
# Dropping the rows with no values for 'PRICE'
housing_data_pd.dropna(subset=['PRICE'], inplace=True)
# Data is assorted in ascending order
housing_data_pd.sort_values('SALEDATE', inplace=True)
# Sales by the date will be grouped by the average of the price
sales_date = housing_data_pd.groupby('SALEDATE')['PRICE'].mean().reset_index()
# Plotting the timeseries of the overall average prices based on the date and time
plt.figure(figsize=(12, 6))
plt.plot(sales_date['SALEDATE'], sales_date['PRICE'], marker='o', linestyle='-')
plt.title('Average Home Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Average Price')
plt.show()
Average Home Prices Over Time
1e8
Average Price
Date
Command took 0.79 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

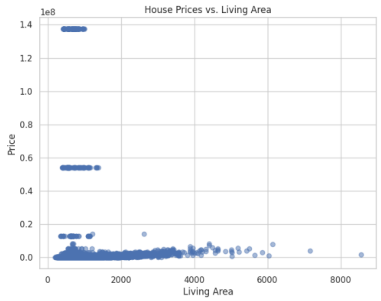

databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Command took 0.79 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 28
# House Prices by Property Grade
plt.figure(figsize=(20, 6))
sns.boxplot(x='GRADE', y='PRICE', data=housing_data_pd)
plt.title('House Prices by Property Grade')
plt.xlabel('Grade')
plt.ylabel('Price')
plt.show()
House Prices by Property Grade
1e7
Price
Average
Above Average
Good Quality
Exceptional-D
Very Good
Excellent
Superior Grade
Exceptional-A
Exceptional-B
Exceptional-C
Fair Quality
No Data
Low Quality
Command took 0.69 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 29
# Houses Prices by Property Structure
plt.figure(figsize=(20, 6))
sns.boxplot(x='STRUCT', y='PRICE', data=housing_data_pd)
plt.title('House Prices by Property Structure')
plt.xlabel('Structure')
plt.ylabel('Price')
plt.show()
House Prices by Property Structure
1e7
Price
Semi-Detached
Single
Row Inside
Row End
Multi
Default
Town Inside
Town End
Structure
Command took 0.59 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

AIT 614-SecDL2_Team5_Sys

```python
# House Prices by Living Area

plt.figure(figsize=(8, 6))
plt.scatter(housing_data_pd['LIVING_GRA'], housing_data_pd['PRICE'], alpha=0.5)
plt.title('House Prices vs. Living Area')
plt.xlabel('Living Area')
plt.ylabel('Price')
plt.show()
```

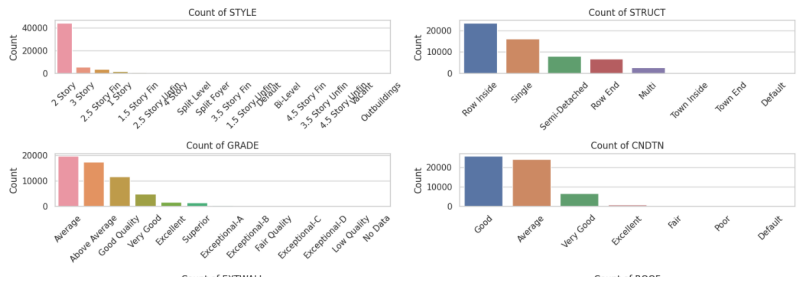


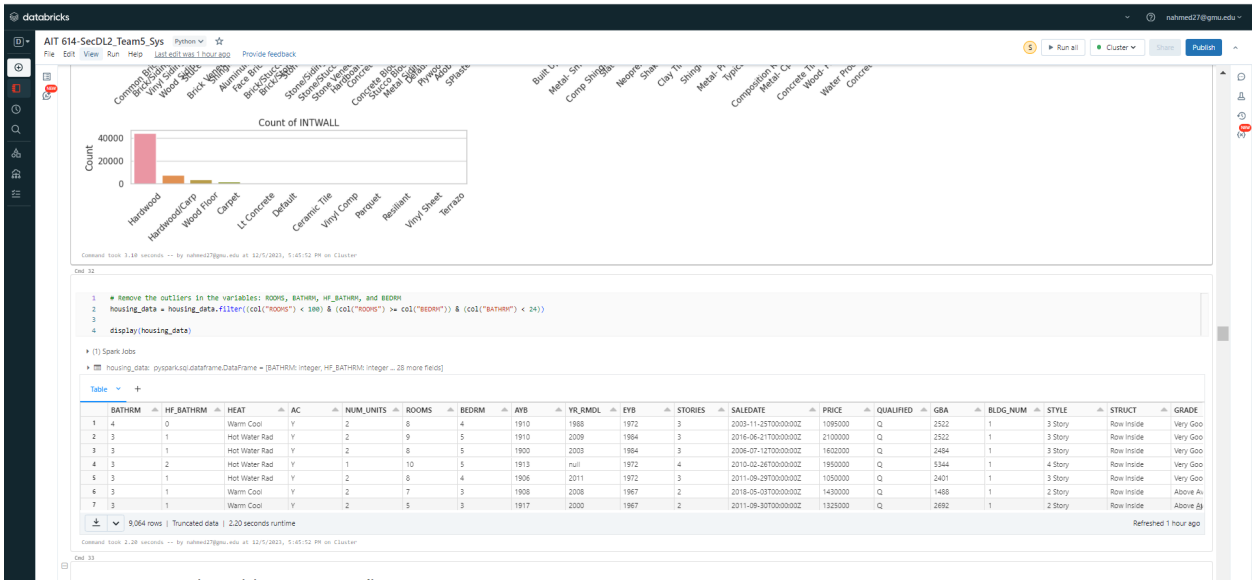
Command took 0.69 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

AIT 614-SecDL2_Team5_Sys

```python
# Visualize which the type of house features have the highest value count

# categorical columns chosen to portray the type of house features that has the highest value of count
categorical_columns = ['STYLE', 'STRUCT', 'GRADE', 'CNDTN', 'EXTWALL', 'ROOF', 'INTWALL']

# Filtering
categorical_data = housing_data_pd[categorical_columns]

# Subplots
num_plots = len(categorical_columns)
plt.figure(figsize=(15, 10))

for i, column in enumerate(categorical_data):
    plt.subplot((num_plots // 2) + 1, 2, i + 1)
    sns.countplot(data=categorical_data, x=column, order=categorical_data[column].value_counts().index)
    plt.title(f'Count of {column}')
    plt.xlabel('')
    plt.ylabel('Count')
    plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

plt.tight_layout()
plt.show()
```






```
# Remove the outliers in the variables: ROOMS, BATHRM, HF_BATHRM, and BEDRM
housing_data = housing_data.filter((col("ROOMS") < 100) & (col("ROOMS") >= col("BEDRM")) & (col("BATHRM") < 24))

display(housing_data)
```

(1) Spark Jobs

housing_data: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 28 more fields]

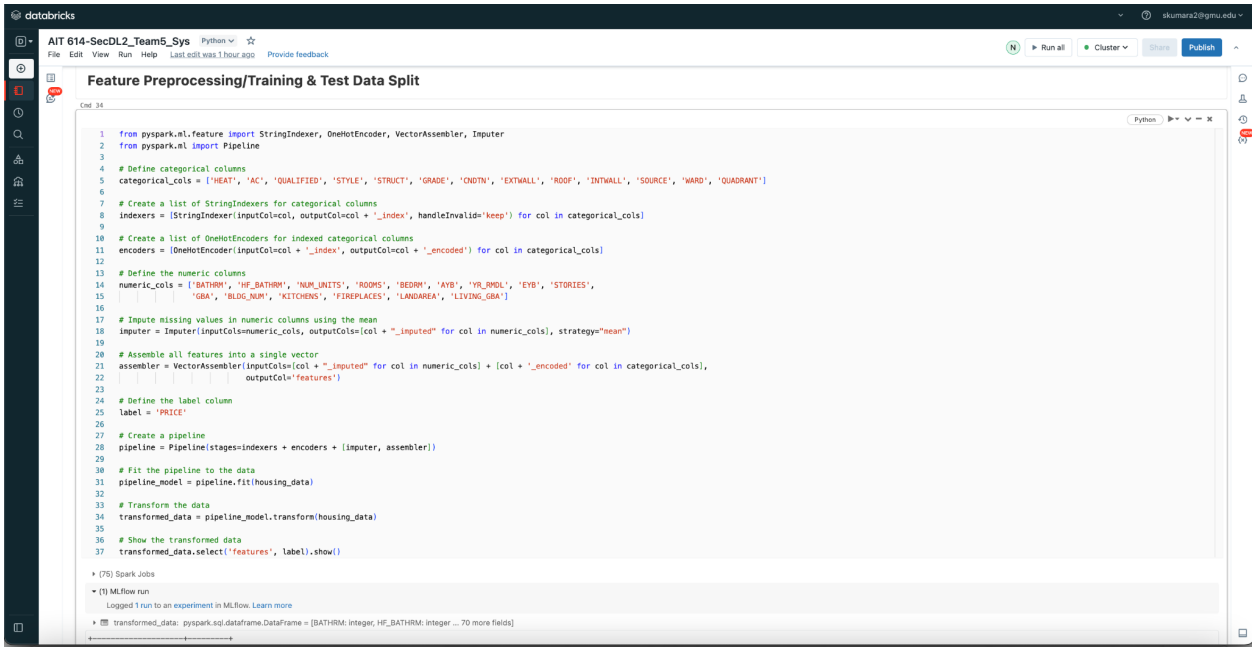
Table

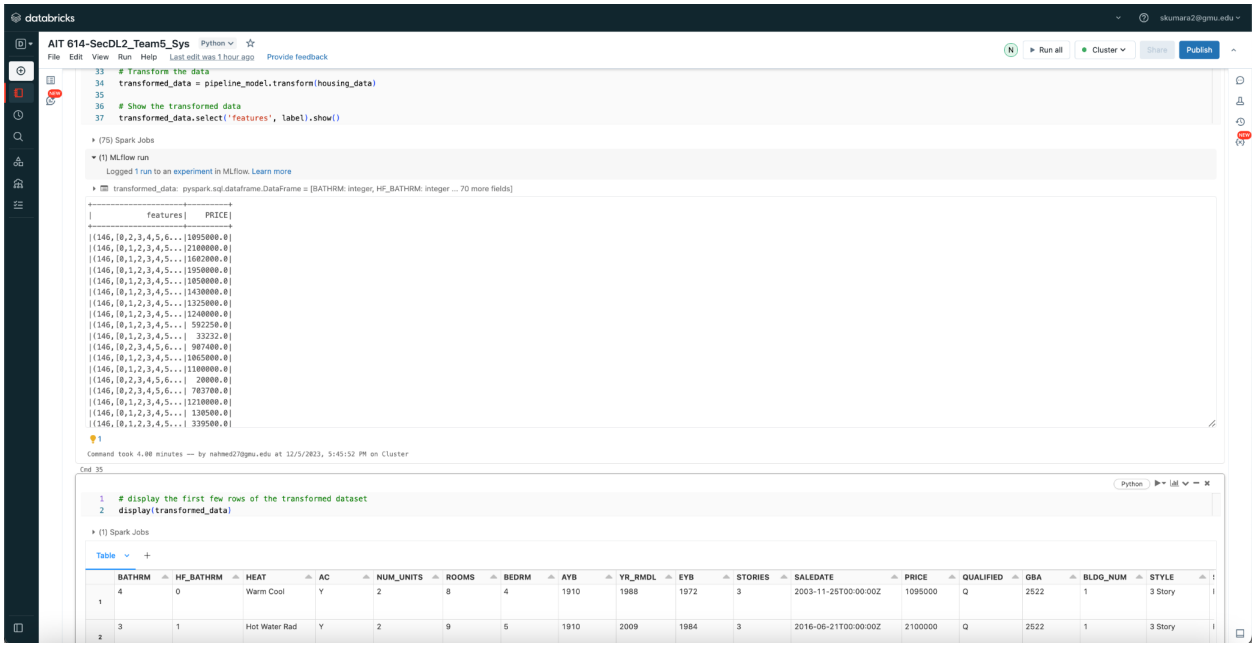
	BATHRM	HF_BATHRM	HEAT	AC	NUM_UNITS	ROOMS	BEDRM	AYB	YR_RMDL	EYB	STORIES	SALEDATE	PRICE	QUALIFIED	GBA	BLDG_NUM	STYLE	STRUCT	GRADE
1	4	0	Warm Cool	Y	2	8	4	1910	1988	1972	3	2003-11-25T00:00:00Z	1095000	Q	2522	1	3 Story	Row Inside	Very Goo
2	3	1	Hot Water Rad	Y	2	9	5	1910	2009	1964	3	2016-06-21T00:00:00Z	2100000	Q	2522	1	3 Story	Row Inside	Very Goo
3	3	1	Hot Water Rad	Y	2	8	5	1900	2003	1964	3	2006-07-12T00:00:00Z	1602000	Q	2484	1	3 Story	Row Inside	Very Goo
4	3	2	Hot Water Rad	Y	1	10	5	1913	null	1972	4	2010-02-26T00:00:00Z	1950000	Q	5344	1	4 Story	Row Inside	Very Goo
5	3	1	Hot Water Rad	Y	2	8	4	1906	2011	1972	3	2011-09-29T00:00:00Z	1050000	Q	2401	1	3 Story	Row Inside	Very Goo
6	3	1	Warm Cool	Y	2	7	3	1908	2008	1967	2	2018-05-03T00:00:00Z	1430000	Q	1488	1	2 Story	Row Inside	Above Av
7	3	1	Warm Cool	Y	2	5	3	1917	2000	1967	2	2011-09-30T00:00:00Z	1325000	Q	2692	1	2 Story	Row Inside	Above Av

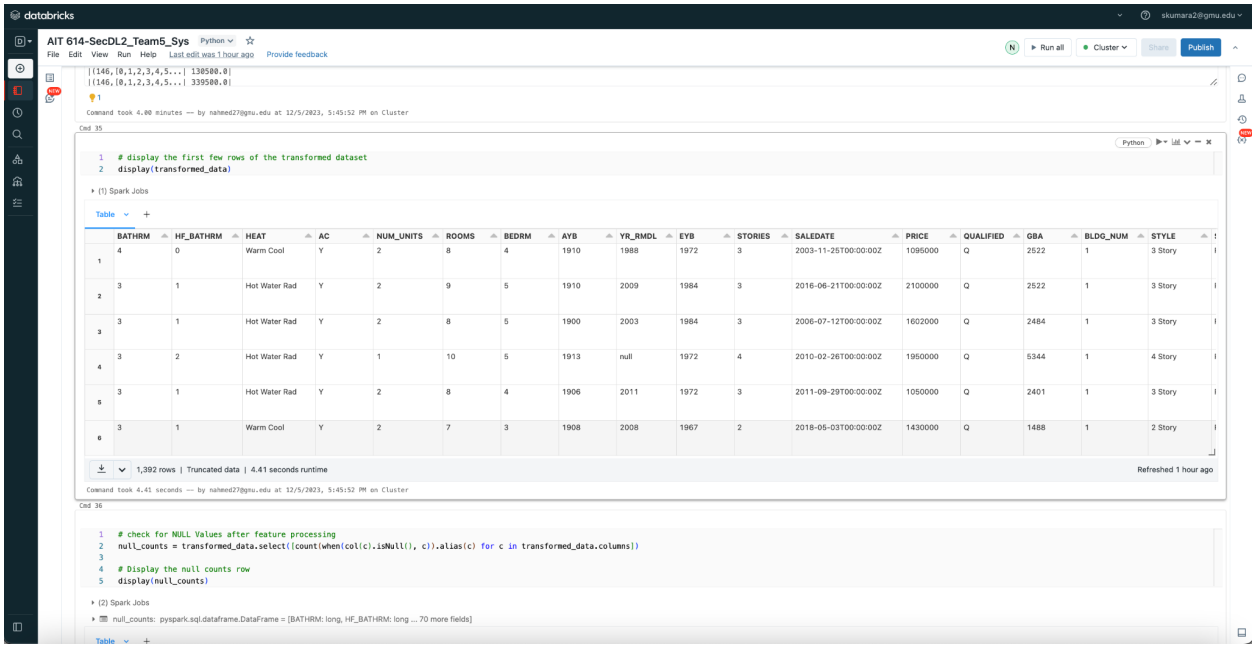
9,064 rows | Truncated data | 2.20 seconds runtime

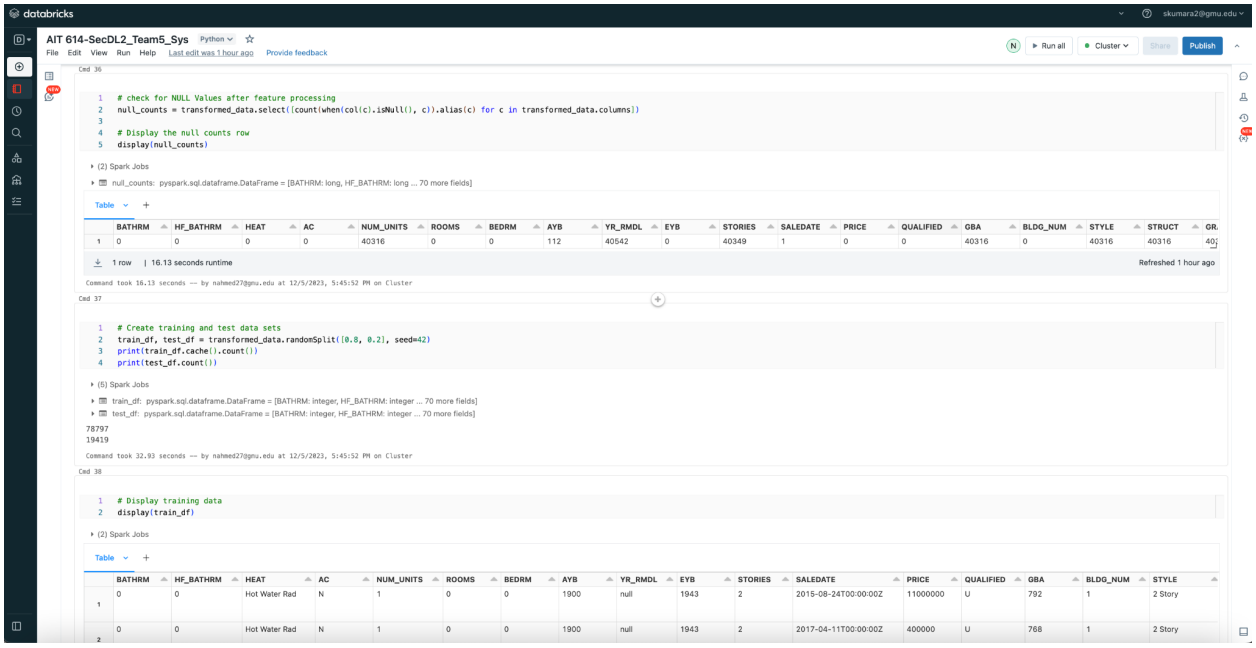
Refreshed 1 hour ago

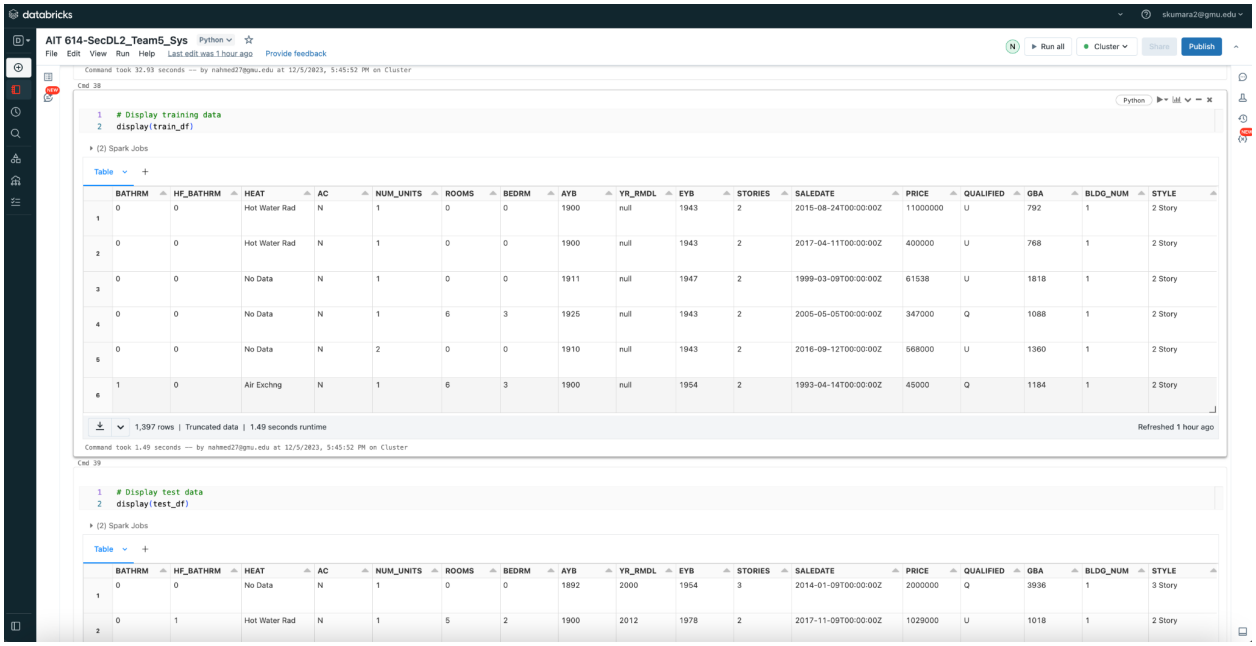
Feature Preprocessing/Training & Test Data Split:

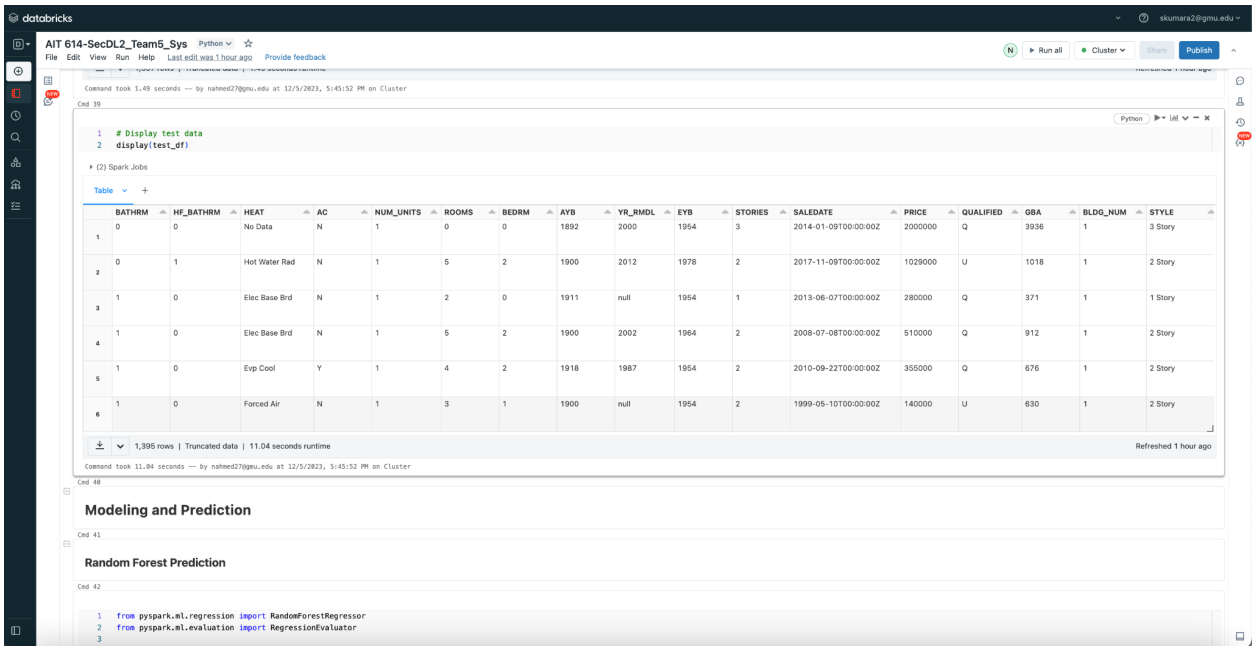





databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
|(146,[0,1,2,3,4,5...| 130500.0|
|(146,[0,1,2,3,4,5...| 339500.0|
Command took 4.00 minutes -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 35
# display the first few rows of the transformed dataset
display(transformed_data)
(1) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 4 0 Warm Cool Y 2 8 4 1910 1988 1972 3 2003-11-25T00:00:00Z 1095000 Q 2522 1 3 Story
2 3 1 Hot Water Rad Y 2 9 5 1910 2009 1984 3 2016-06-21T00:00:00Z 2100000 Q 2522 1 3 Story
3 3 1 Hot Water Rad Y 2 8 5 1900 2003 1984 3 2006-07-12T00:00:00Z 1602000 Q 2484 1 3 Story
4 3 2 Hot Water Rad Y 1 10 5 1913 null 1972 4 2010-02-26T00:00:00Z 1950000 Q 5344 1 4 Story
5 3 1 Hot Water Rad Y 2 8 4 1906 2011 1972 3 2011-09-29T00:00:00Z 1050000 Q 2401 1 3 Story
6 3 1 Warm Cool Y 2 7 3 1908 2008 1967 2 2018-05-03T00:00:00Z 1430000 Q 1488 1 2 Story
1,392 rows | Truncated data | 4.41 seconds runtime
Refreshed 1 hour ago
Command took 4.41 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 36
# check for NULL Values after feature processing
null_counts = transformed_data.select([count(when(col(c).isNull(), c)).alias(c) for c in transformed_data.columns])
# Display the null counts row
display(null_counts)
(2) Spark Jobs
null_counts: pyspark.sql.dataframe.DataFrame = [BATHRM: long, HF_BATHRM: long ... 70 more fields]
Table

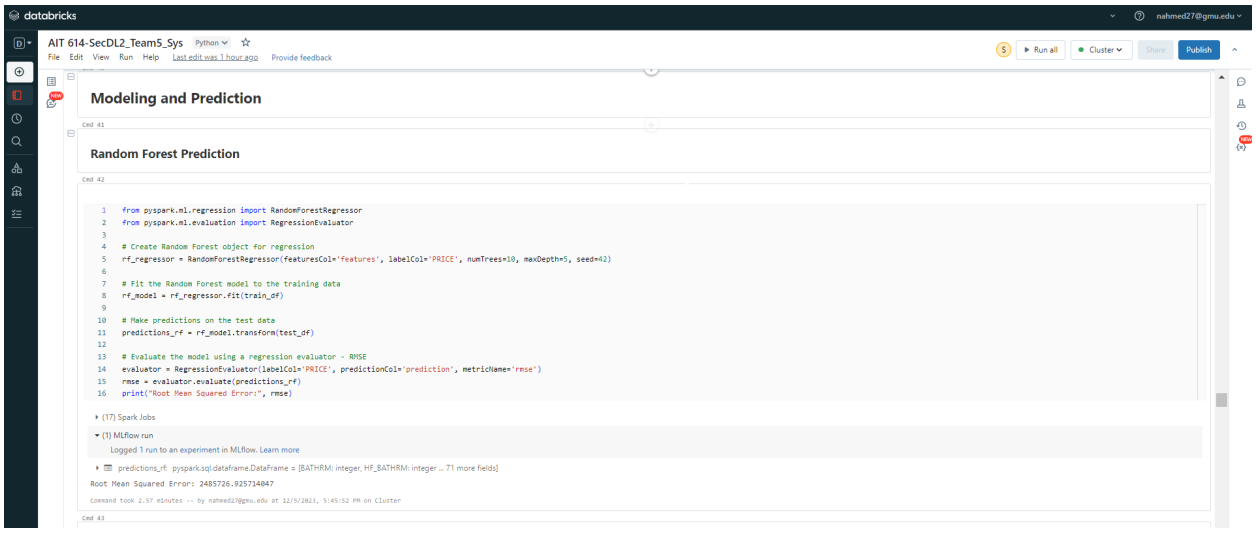

databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 36
# check for NULL Values after feature processing
null_counts = transformed_data.select([count(when(col(c).isNull(), c)).alias(c) for c in transformed_data.columns])
# Display the null counts row
display(null_counts)
(2) Spark Jobs
null_counts: pyspark.sql.dataframe.DataFrame = [BATHRM: long, HF_BATHRM: long ... 70 more fields]
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE STRUCT GR.
1 0 0 0 0 40316 0 0 112 40542 0 40349 1 0 0 40316 0 40316 40316 40
1 row | 16.13 seconds runtime
Refreshed 1 hour ago
Command took 16.13 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 37
Create training and test data sets
train_df, test_df = transformed_data.randomSplit([0.8, 0.2], seed=42)
print(train_df.cache().count())
print(test_df.count())
(5) Spark Jobs
train_df: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 70 more fields]
test_df: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 70 more fields]
78797
19419
Command took 32.93 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 38
# Display training data
display(train_df)
(2) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 Hot Water Rad N 1 0 0 1900 null 1943 2 2015-08-24T00:00:00Z 11000000 U 792 1 2 Story
2 0 0 Hot Water Rad N 1 0 0 1900 null 1943 2 2017-04-11T00:00:00Z 400000 U 768 1 2 Story

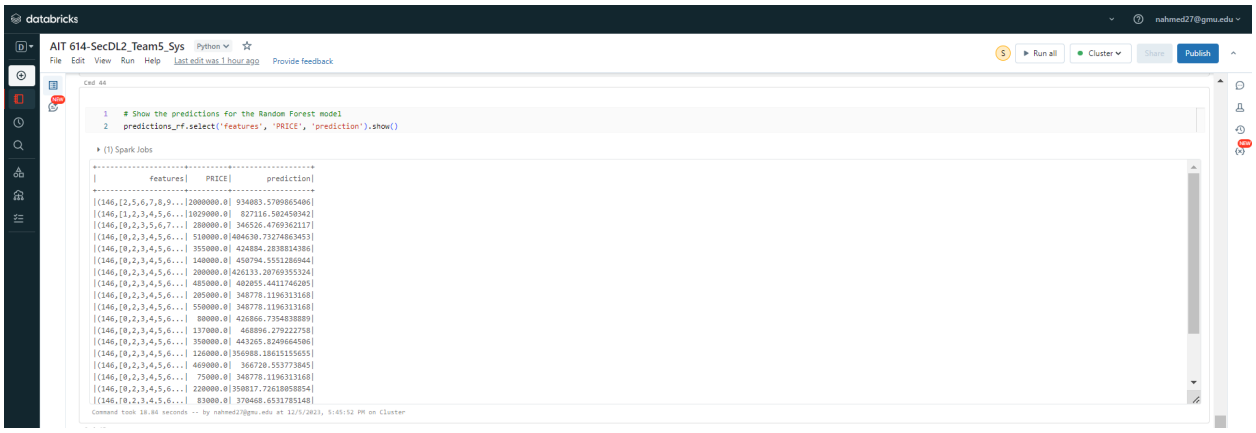

databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Command took 32.93 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 38
# Display training data
display(train_df)
(2) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 Hot Water Rad N 1 0 0 1900 null 1943 2 2015-08-24T00:00:00Z 11000000 U 792 1 2 Story
2 0 0 Hot Water Rad N 1 0 0 1900 null 1943 2 2017-04-11T00:00:00Z 400000 U 768 1 2 Story
3 0 0 No Data N 1 0 0 1911 null 1947 2 1999-03-09T00:00:00Z 61538 U 1818 1 2 Story
4 0 0 No Data N 1 6 3 1925 null 1943 2 2005-05-05T00:00:00Z 347000 Q 1088 1 2 Story
5 0 0 No Data N 2 0 0 1910 null 1943 2 2016-09-12T00:00:00Z 568000 U 1360 1 2 Story
6 1 0 Air Exchng N 1 6 3 1900 null 1954 2 1993-04-14T00:00:00Z 45000 Q 1184 1 2 Story
1,397 rows | Truncated data | 1.49 seconds runtime
Refreshed 1 hour ago
Command took 1.49 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 39
# Display test data
display(test_df)
(2) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 No Data N 1 0 0 1892 2000 1954 3 2014-01-09T00:00:00Z 2000000 Q 3936 1 3 Story
2 0 1 Hot Water Rad N 1 5 2 1900 2012 1978 2 2017-11-09T00:00:00Z 1029000 U 1018 1 2 Story


databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Command took 1.49 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 39
# Display test data
display(test_df)
(2) Spark Jobs
Table
BATHRM HF_BATHRM HEAT AC NUM_UNITS ROOMS BEDRM AYB YR_RMDL EYB STORIES SALEDATE PRICE QUALIFIED GBA BLDG_NUM STYLE
1 0 0 No Data N 1 0 0 1892 2000 1954 3 2014-01-09T00:00:00Z 2000000 Q 3936 1 3 Story
2 0 1 Hot Water Rad N 1 5 2 1900 2012 1978 2 2017-11-09T00:00:00Z 1029000 U 1018 1 2 Story
3 1 0 Elec Base Brd N 1 2 0 1911 null 1954 1 2013-06-07T00:00:00Z 280000 Q 371 1 1 Story
4 1 0 Elec Base Brd N 1 5 2 1900 2002 1964 2 2008-07-08T00:00:00Z 510000 Q 912 1 2 Story
5 1 0 Evp Cool Y 1 4 2 1918 1987 1954 2 2010-09-22T00:00:00Z 355000 Q 676 1 2 Story
6 1 0 Forced Air N 1 3 1 1900 null 1954 2 1999-05-10T00:00:00Z 140000 U 630 1 2 Story
1,395 rows | Truncated data | 11.04 seconds runtime
Refreshed 1 hour ago
Command took 11.04 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 40
Modeling and Prediction
Cmd 41
Random Forest Prediction
Cmd 42
from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.evaluation import RegressionEvaluator

Modeling and Prediction -

Random Forest Prediction:





databricks

nahmed27@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

```
#Determine Feature Importance for the Random Forest Model
importances_rf = rf_model.featureImportances.toArray()

# Get feature names from the vector assembler
feature_names = assembler.getInputCols()

# Create a Pandas DataFrame with feature names and importance scores
importances_df_pd_rf = pd.DataFrame(list(zip(feature_names, importances_rf)), columns=["Attribute_rf", "Importance_rf"])

# Create a Spark DataFrame from the Pandas DataFrame
importances_df = spark.createDataFrame(importances_df_pd_rf)

# Sort the DataFrame by importance in descending order
importances_df = importances_df.sort("Importance_rf", ascending=False)

# Show the result
importances_df.show()
```

▸ (1) Spark Jobs

▸ importances_df: pyspark.sql.dataframe.DataFrame = [Attribute_rf: string, Importance_rf: double]

Attribute_rf	Importance_rf
YR_RMDL_imputed	0.3395122532024931
EYB_imputed	0.09920688659932825
STYLE_encoded	0.09284572634867281
LIVING_GBA_imputed	0.04282543688863448
AYB_imputed	0.03799974093689044
LANDAREA_imputed	0.008224651899755079
BEDRM_imputed	0.007605687193851891
NUM_UNITS_imputed	0.002561003192665...
BATHRM_imputed	7.919145961847923E-4
HEAT_encoded	6.568424393298017E-4
ROOMS_imputed	4.092164217425647...
FIREPLACES_imputed	4.870576676286021E-4
GBA_imputed	3.274670828108505E-4
HF_BATHRM_imputed	3.203500060445433E-5
STORIES_imputed	2.812039250739419...
INTWALL_encoded	0.0
BLDG_NUM_imputed	0.0
SOURCE_encoded	0.0

databricks

nahmed27@gmu.edu

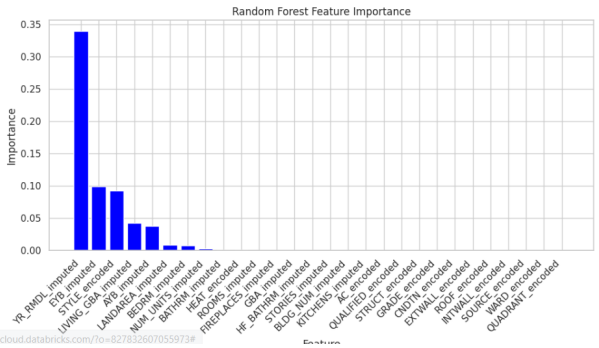
AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

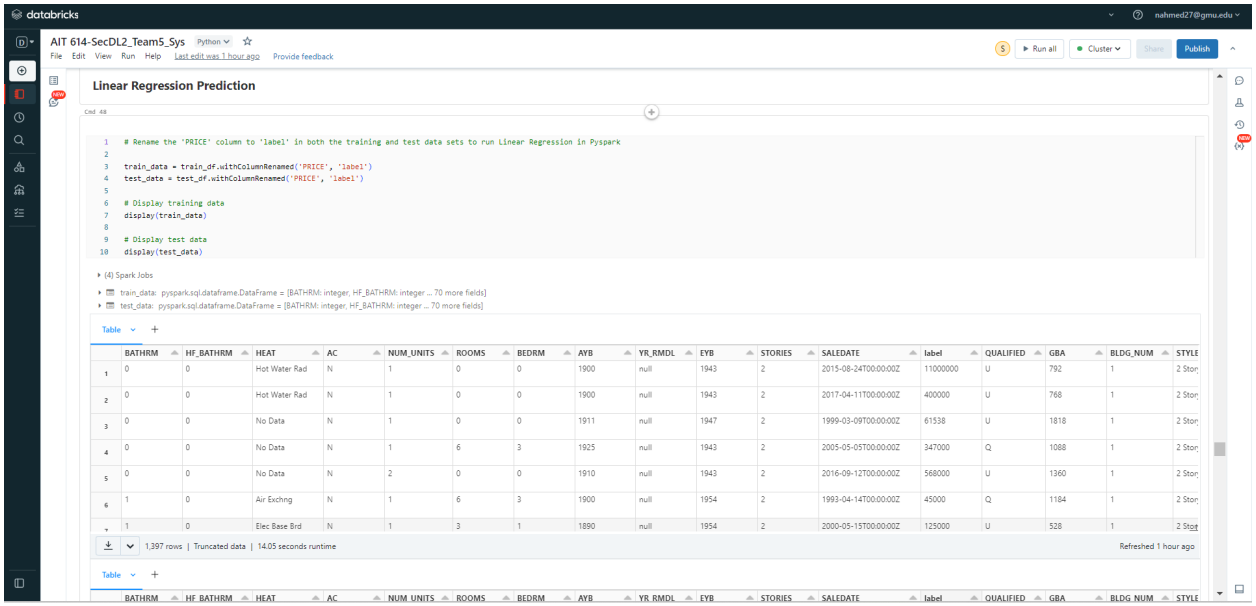
```
# Visalizing the feature importance for Random Forest using a bar plot.
importances_df_pd = importances_df.toPandas()

plt.figure(figsize=(10, 6))
plt.bar(importances_df_pd['Attribute_rf'], importances_df_pd['Importance_rf'], color='blue')
plt.xlabel('Feature')
plt.ylabel('Importance')
plt.title('Random Forest Feature Importance')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

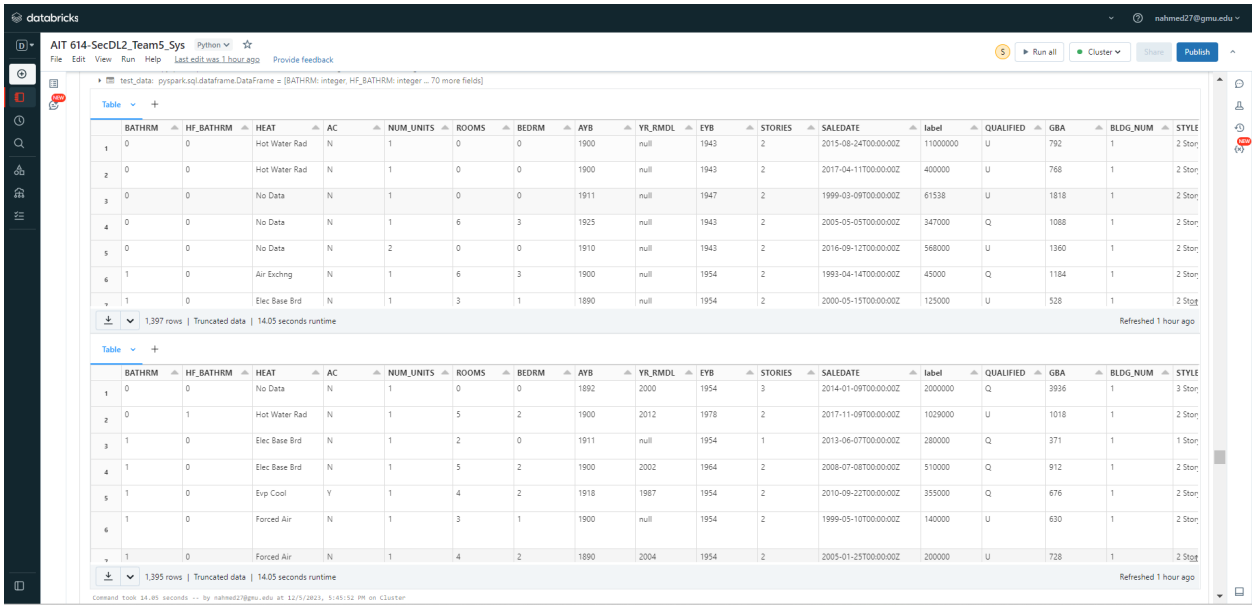
▸ (3) Spark Jobs

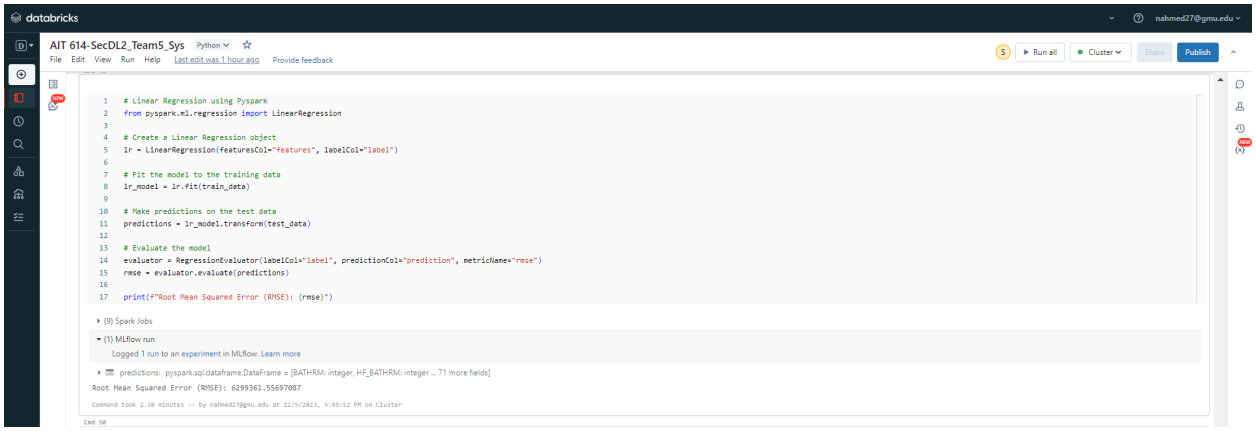


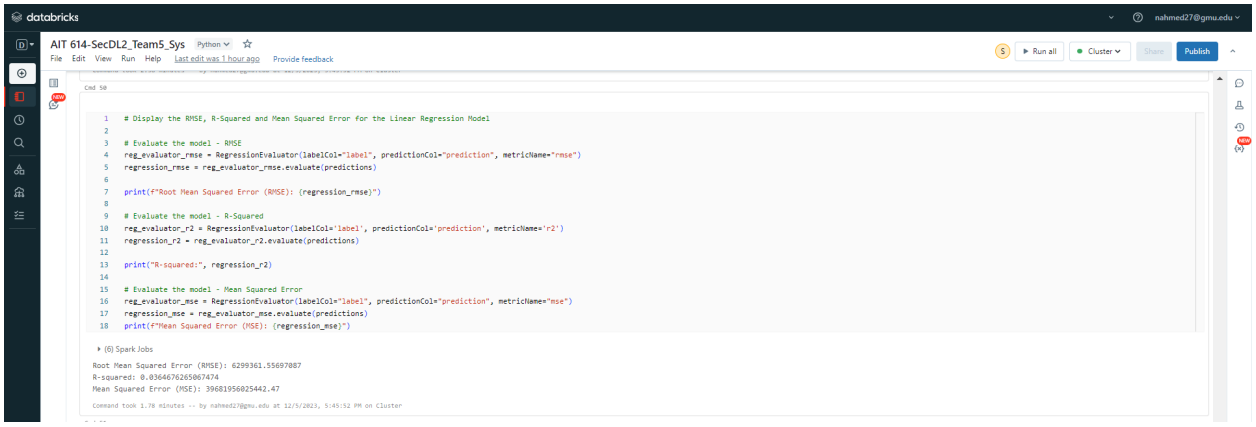
https://community.cloud.databricks.com/?o=8278326070559738#

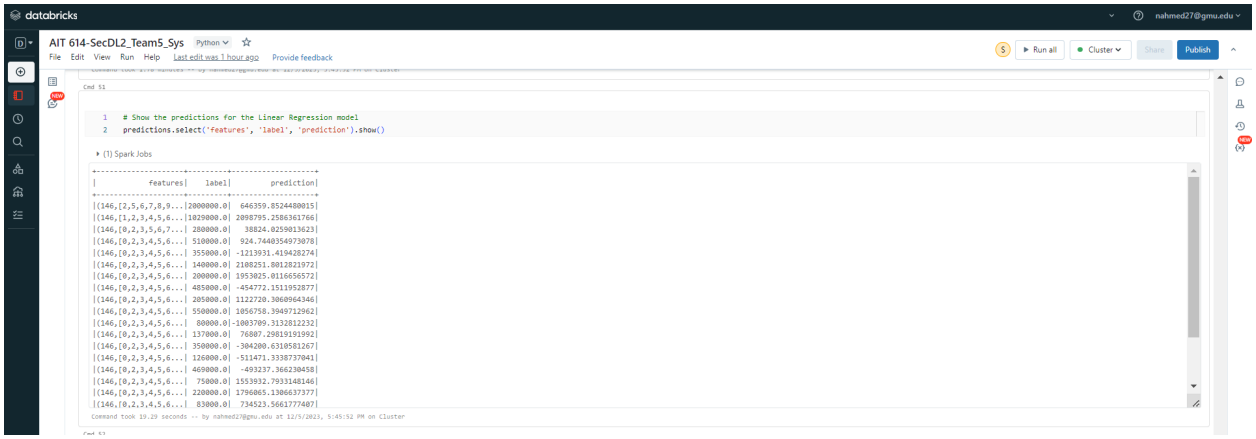
Linear Regression Prediction:

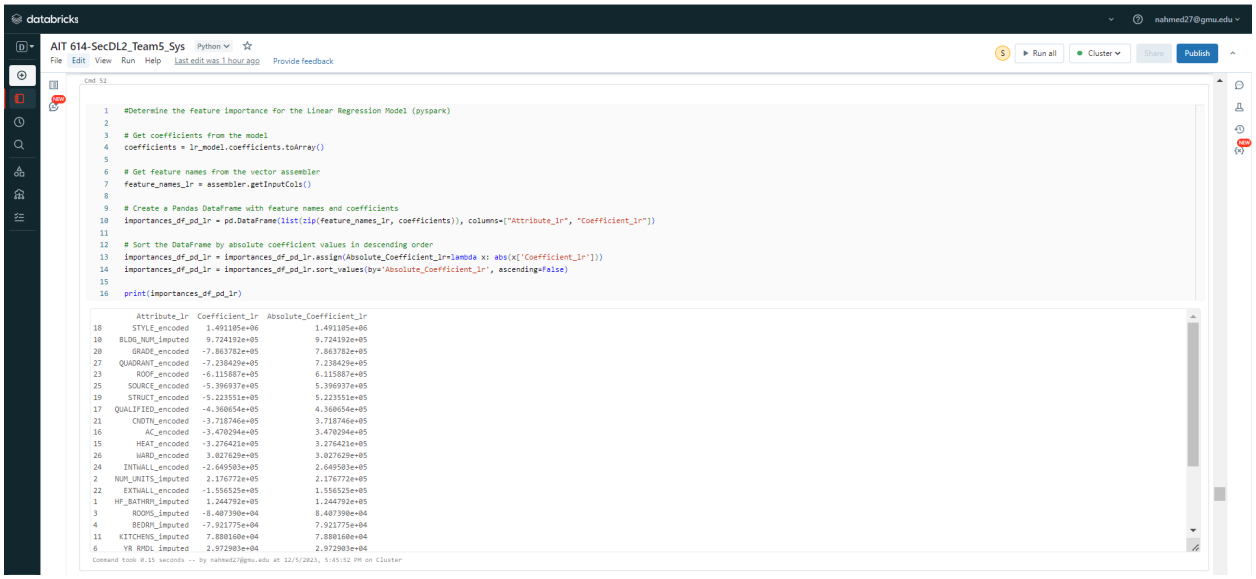







databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 50
# Display the RMSE, R-Squared and Mean Squared Error for the Linear Regression Model
# Evaluate the model - RMSE
reg_evaluator_rmse = RegressionEvaluator(labelCol="label", predictionCol="prediction", metricName="rmse")
regression_rmse = reg_evaluator_rmse.evaluate(predictions)
print(f"Root Mean Squared Error (RMSE): {regression_rmse}")
# Evaluate the model - R-Squared
reg_evaluator_r2 = RegressionEvaluator(labelCol='label', predictionCol='prediction', metricName='r2')
regression_r2 = reg_evaluator_r2.evaluate(predictions)
print("R-squared:", regression_r2)
# Evaluate the model - Mean Squared Error
reg_evaluator_mse = RegressionEvaluator(labelCol="label", predictionCol="prediction", metricName="mse")
regression_mse = reg_evaluator_mse.evaluate(predictions)
print(f"Mean Squared Error (MSE): {regression_mse}")
(6) Spark Jobs
Root Mean Squared Error (RMSE): 6299361.55697087
R-squared: 0.0364676265067474
Mean Squared Error (MSE): 39681956025442.47
Command took 1.78 minutes -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 51
# Show the predictions for the Linear Regression model
predictions.select('features', 'label', 'prediction').show()
(1) Spark Jobs
+--------------------+---------+------------------+
| features| label| prediction|
+--------------------+---------+------------------+
|(146,[2,5,6,7,8,9...|2000000.0| 646359.8524480015|
|(146,[1,2,3,4,5,6...|1029000.0| 2098795.2586361766|
|(146,[0,2,3,5,6,7...| 280000.0| 38824.0259013623|
|(146,[0,2,3,4,5,6...| 510000.0| 924.7440354973078|
|(146,[0,2,3,4,5,6...| 355000.0| -1213931.4194828274|
|(146,[0,2,3,4,5,6...| 140000.0| 2108251.8012821972|
|(146,[0,2,3,4,5,6...| 200000.0| 1953025.0116656572|
|(146,[0,2,3,4,5,6...| 485000.0| -454772.1511952877|
|(146,[0,2,3,4,5,6...| 205000.0| 1122720.3060964346|
|(146,[0,2,3,4,5,6...| 550000.0| 1056758.3949712962|
|(146,[0,2,3,4,5,6...| 80000.0|-1003709.3132812232|
|(146,[0,2,3,4,5,6...| 137000.0| 76807.29819191992|
|(146,[0,2,3,4,5,6...| 350000.0| -304200.6310581267|
|(146,[0,2,3,4,5,6...| 126000.0| -511471.3338737041|
|(146,[0,2,3,4,5,6...| 469000.0| -493237.366230458|
|(146,[0,2,3,4,5,6...| 75000.0| 1553932.7933148146|
|(146,[0,2,3,4,5,6...| 220000.0| 1796065.1306637377|
|(146,[0,2,3,4,5,6...| 83000.0| 734523.5661777407|
Command took 19.29 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster


databricks
nahmed27@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Cmd 52
#Determine the feature importance for the Linear Regression Model (pyspark)
# Get coefficients from the model
coefficients = lr_model.coefficients.toArray()
# Get feature names from the vector assembler
feature_names_lr = assembler.getInputCols()
# Create a Pandas DataFrame with feature names and coefficients
importances_df_pd_lr = pd.DataFrame(list(zip(feature_names_lr, coefficients)), columns=["Attribute_lr", "Coefficient_lr"])
# Sort the DataFrame by absolute coefficient values in descending order
importances_df_pd_lr = importances_df_pd_lr.assign(Absolute_Coefficient_lr=lambda x: abs(x['Coefficient_lr']))
importances_df_pd_lr = importances_df_pd_lr.sort_values(by='Absolute_Coefficient_lr', ascending=False)
print(importances_df_pd_lr)
Attribute_lr Coefficient_lr Absolute_Coefficient_lr
18 STYLE_encoded 1.491105e+06 1.491105e+06
10 BLDG_NUM_imputed 9.724192e+05 9.724192e+05
20 GRADE_encoded -7.863782e+05 7.863782e+05
27 QUADRANT_encoded -7.238429e+05 7.238429e+05
23 ROOF_encoded -6.115887e+05 6.115887e+05
25 SOURCE_encoded -5.396937e+05 5.396937e+05
19 STRUCT_encoded -5.223551e+05 5.223551e+05
17 QUALIFIED_encoded -4.360654e+05 4.360654e+05
21 CNDTN_encoded -3.718746e+05 3.718746e+05
16 AC_encoded -3.470294e+05 3.470294e+05
15 HEAT_encoded -3.276421e+05 3.276421e+05
26 WARD_encoded 3.027629e+05 3.027629e+05
24 INTWALL_encoded -2.649503e+05 2.649503e+05
2 NUM_UNITS_imputed 2.176772e+05 2.176772e+05
22 EXTWALL_encoded -1.556525e+05 1.556525e+05
1 HF_BATHRM_imputed 1.244792e+05 1.244792e+05
3 ROOMS_imputed -8.407390e+04 8.407390e+04
4 BEDRM_imputed -7.921775e+04 7.921775e+04
11 KITCHENS_imputed 7.880160e+04 7.880160e+04
6 YR RMDL_imputed 2.972903e+04 2.972903e+04
Command took 0.15 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

databricks

nahmed27@gmu.edu

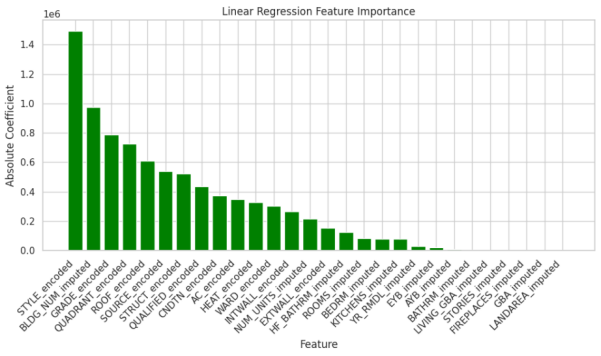
AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

Cmd 53

```python
# # Visalizing the feature importance for Linear Regression using a bar plot.
importances_df_pd_lr = importances_df_pd_lr.sort_values(by='Absolute_Coefficient_lr', ascending=False)

plt.figure(figsize=(10, 6))
plt.bar(importances_df_pd_lr['Attribute_lr'], importances_df_pd_lr['Absolute_Coefficient_lr'], color='green')
plt.xlabel('Feature')
plt.ylabel('Absolute Coefficient')
plt.title('Linear Regression Feature Importance')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



databricks

nahmed27@gmu.edu

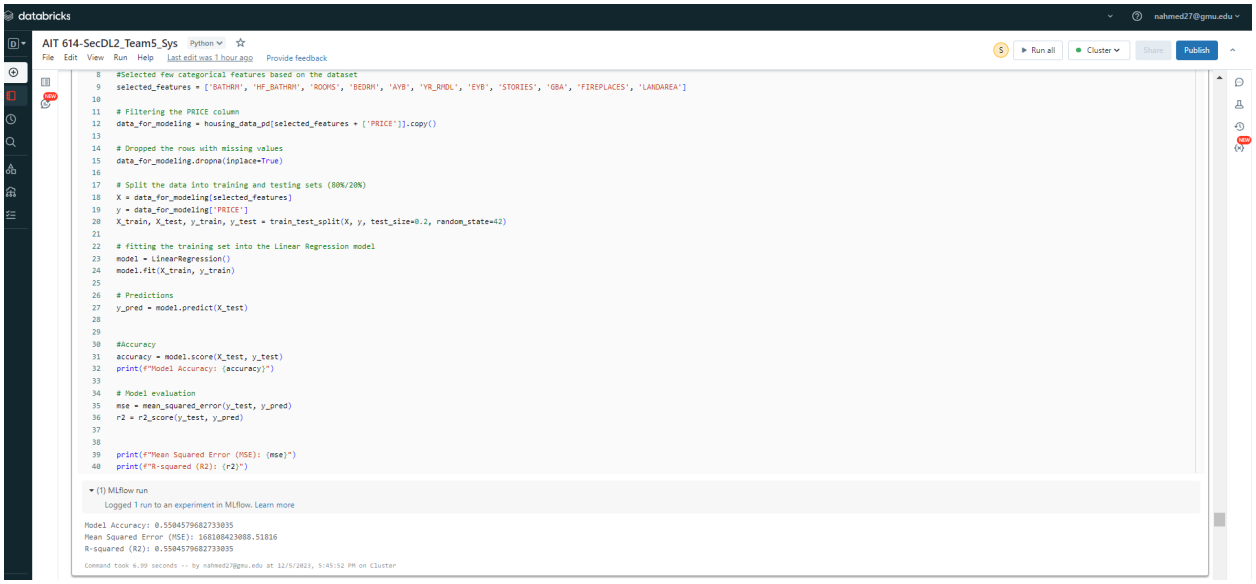
AIT 614-SecDL2_Team5_Sys Python

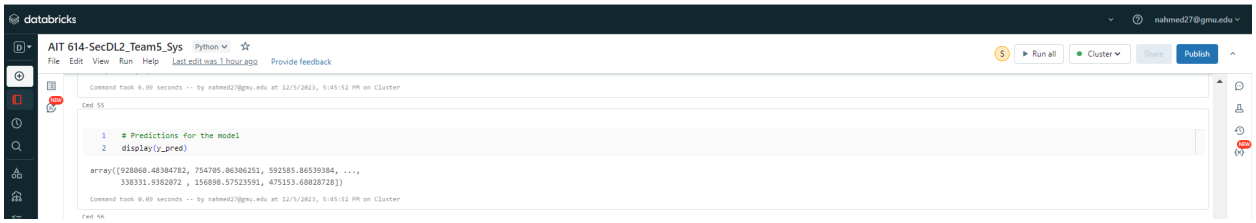
File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

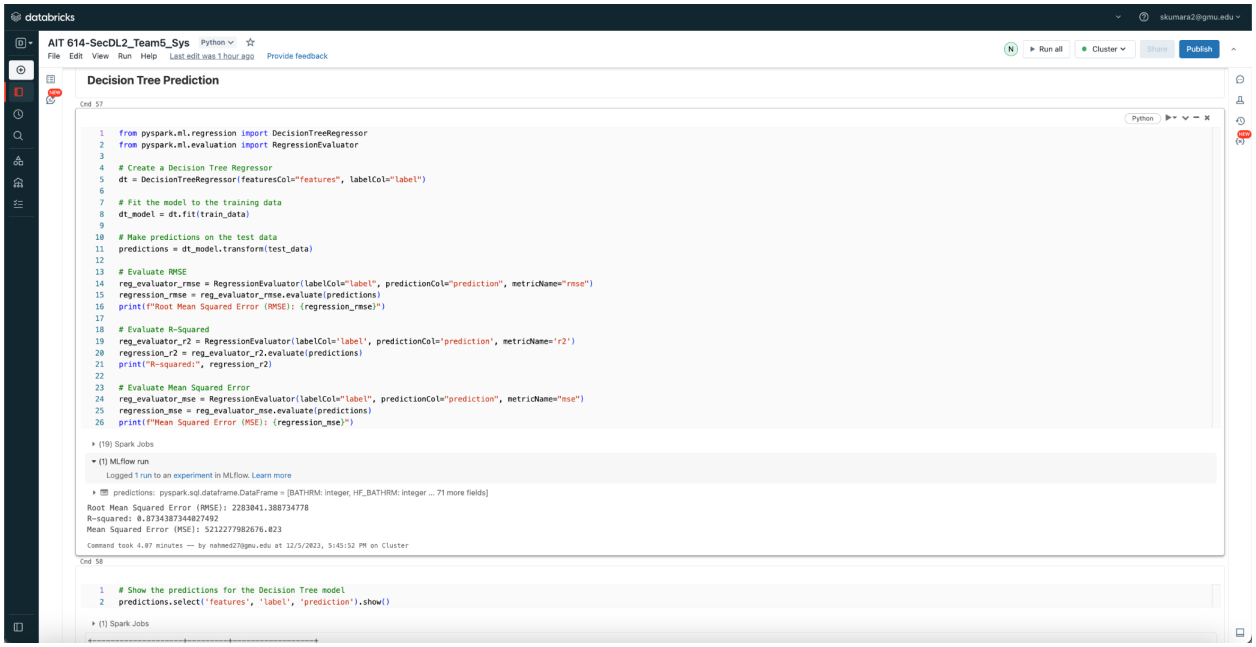
Cmd 54

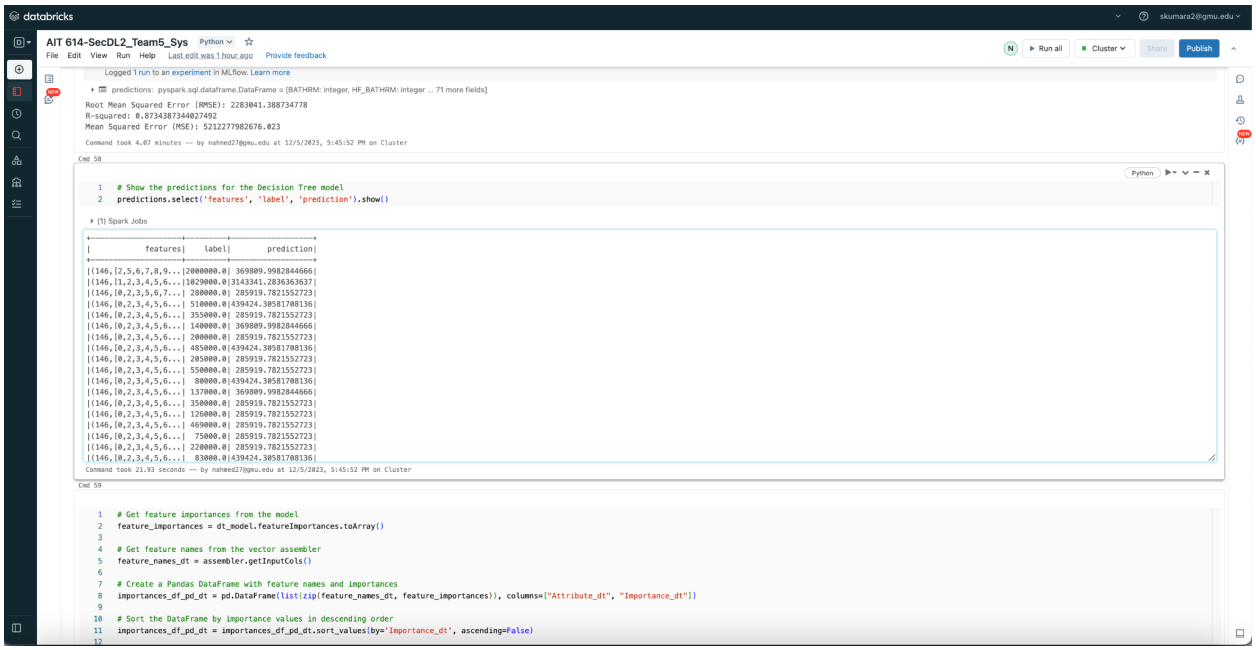
```python
#Here is a simple linear regression model using sklearn that prints the accuracy and r^2 score in comparison to the other one. It looks like that the linear regression algorithm is not a good fit to this dataset.

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# 'PRICE' is the target variable
#Selected few categorical features based on the dataset
selected_features = ['BATHRM', 'HF_BATHRM', 'ROOMS', 'BEDRM', 'AYB', 'YR_RMDL', 'EYB', 'STORIES', 'GBA', 'FIREPLACES', 'LANDAREA']

# Filtering the PRICE column
data_for_modeling = housing_data_pd[selected_features + ['PRICE']].copy()

# Dropped the rows with missing values
data_for_modeling.dropna(inplace=True)

# Split the data into training and testing sets (80%/20%)
X = data_for_modeling[selected_features]
y = data_for_modeling['PRICE']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# fitting the training set into the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)


#Accuracy
accuracy = model.score(X_test, y_test)
print(f"Model Accuracy: {accuracy}")

# Model evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)


print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R2): {r2}")
```

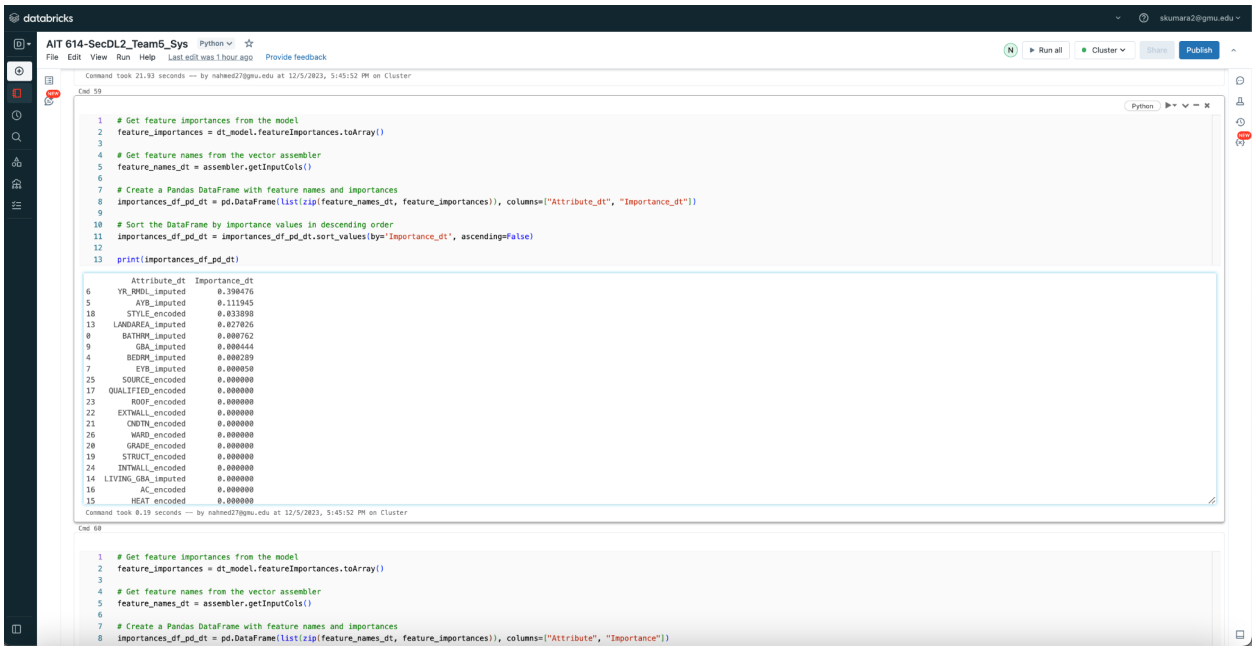




Decision Tree Prediction:




databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Logged 1 run to an experiment in MLflow. Learn more
predictions: pyspark.sql.dataframe.DataFrame = [BATHRM: integer, HF_BATHRM: integer ... 71 more fields]
Root Mean Squared Error (RMSE): 2283041.388734778
R-squared: 0.8734387344027492
Mean Squared Error (MSE): 5212279982676.023
Command took 4.07 minutes -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 58
# Show the predictions for the Decision Tree model
predictions.select('features', 'label', 'prediction').show()
(1) Spark Jobs
+--------------------+---------+------------------+
| features| label| prediction|
+--------------------+---------+------------------+
|(146,[2,5,6,7,8,9...|2000000.0| 369809.9982844666|
|(146,[1,2,3,4,5,6...|1029000.0|3143341.2836363637|
|(146,[0,2,3,5,6,7...| 280000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 510000.0|439424.30581708136|
|(146,[0,2,3,4,5,6...| 355000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 140000.0| 369809.9982844666|
|(146,[0,2,3,4,5,6...| 200000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 485000.0|439424.30581708136|
|(146,[0,2,3,4,5,6...| 205000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 550000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 80000.0|439424.30581708136|
|(146,[0,2,3,4,5,6...| 137000.0| 369809.9982844666|
|(146,[0,2,3,4,5,6...| 350000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 126000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 469000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 75000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 220000.0| 285919.7821552723|
|(146,[0,2,3,4,5,6...| 83000.0|439424.30581708136|
Command took 21.93 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 59
# Get feature importances from the model
feature_importances = dt_model.featureImportances.toArray()
# Get feature names from the vector assembler
feature_names_dt = assembler.getInputCols()
# Create a Pandas DataFrame with feature names and importances
importances_df_pd_dt = pd.DataFrame(list(zip(feature_names_dt, feature_importances)), columns=["Attribute_dt", "Importance_dt"])
# Sort the DataFrame by importance values in descending order
importances_df_pd_dt = importances_df_pd_dt.sort_values(by='Importance_dt', ascending=False)


databricks
skumara2@gmu.edu
AIT 614-SecDL2_Team5_Sys
Python
File Edit View Run Help Last edit was 1 hour ago Provide feedback
Run all
Cluster
Share
Publish
Command took 21.93 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 59
# Get feature importances from the model
feature_importances = dt_model.featureImportances.toArray()
# Get feature names from the vector assembler
feature_names_dt = assembler.getInputCols()
# Create a Pandas DataFrame with feature names and importances
importances_df_pd_dt = pd.DataFrame(list(zip(feature_names_dt, feature_importances)), columns=["Attribute_dt", "Importance_dt"])
# Sort the DataFrame by importance values in descending order
importances_df_pd_dt = importances_df_pd_dt.sort_values(by='Importance_dt', ascending=False)
print(importances_df_pd_dt)
Attribute_dt Importance_dt
6 YR_RMDL_imputed 0.390476
5 AYB_imputed 0.111945
18 STYLE_encoded 0.033898
13 LANDAREA_imputed 0.027026
0 BATHRM_imputed 0.000762
9 GBA_imputed 0.000444
4 BEDRM_imputed 0.000289
7 EYB_imputed 0.000050
25 SOURCE_encoded 0.000000
17 QUALIFIED_encoded 0.000000
23 ROOF_encoded 0.000000
22 EXTWALL_encoded 0.000000
21 CNDTN_encoded 0.000000
26 WARD_encoded 0.000000
20 GRADE_encoded 0.000000
19 STRUCT_encoded 0.000000
24 INTWALL_encoded 0.000000
14 LIVING_GBA_imputed 0.000000
16 AC_encoded 0.000000
15 HEAT_encoded 0.000000
Command took 0.19 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster
Cmd 60
# Get feature importances from the model
feature_importances = dt_model.featureImportances.toArray()
# Get feature names from the vector assembler
feature_names_dt = assembler.getInputCols()
# Create a Pandas DataFrame with feature names and importances
importances_df_pd_dt = pd.DataFrame(list(zip(feature_names_dt, feature_importances)), columns=["Attribute", "Importance"])

databricks

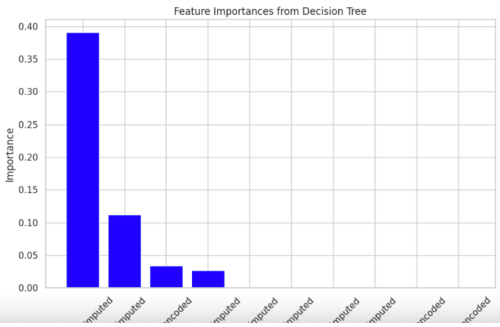
skumara2@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

```python
# Get feature importances from the model
feature_importances = dt_model.featureImportances.toArray()

# Get feature names from the vector assembler
feature_names_dt = assembler.getInputCols()

# Create a Pandas DataFrame with feature names and importances
importances_df_pd_dt = pd.DataFrame(list(zip(feature_names_dt, feature_importances)), columns=["Attribute", "Importance"])

# Sort the DataFrame by importance values in descending order
top_importances_df_pd_dt = importances_df_pd_dt.sort_values(by='Importance', ascending=False).head(10)

# Plotting
plt.figure(figsize=(10, 6))
plt.bar(top_importances_df_pd_dt['Attribute'], top_importances_df_pd_dt['Importance'], color='blue')
plt.xlabel('Features')
plt.ylabel('Importance')
plt.title('Feature Importances from Decision Tree')
plt.xticks(rotation=45)
plt.show()
```



databricks

skumara2@gmu.edu

AIT 614-SecDL2_Team5_Sys Python

File Edit View Run Help Last edit was 1 hour ago Provide feedback

Run all Cluster Share Publish

Command took 0.60 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

Cmd 61

```python
# Get the decision tree model
tree_model = dt_model

# Print out the decision tree
print(tree_model.toDebugString)
```

```
DecisionTreeRegressionModel: uid=DecisionTreeRegressor_fa93f58a01a1, depth=5, numNodes=57, numFeatures=146
  If (feature 136 in {0.0})
   If (feature 135 in {0.0})
    If (feature 72 in {0.0})
     If (feature 0 <= 2.5)
      If (feature 74 in {1.0})
       Predict: 285919.7821552723
      Else (feature 74 not in {1.0})
       Predict: 439424.30581708136
     Else (feature 0 > 2.5)
      If (feature 7 <= 1968.5)
       Predict: 522396.62291853176
      Else (feature 7 > 1968.5)
       Predict: 763864.6055111111
    Else (feature 72 not in {0.0})
     Predict: 2.5E7
   Else (feature 135 not in {0.0})
    If (feature 6 <= 2005.5)
     If (feature 0 <= 2.5)
      If (feature 0 <= 1.5)
       Predict: 369809.9982844666
```

Command took 0.09 seconds -- by nahmed27@gmu.edu at 12/5/2023, 5:45:52 PM on Cluster

Shift+Enter to run
Shift+Ctrl+Enter to run selected text