CIS 4567 Group 10 Preprocessing and Modelling Notebook

April 27, 2022

[1]:
```
%%local
!pip3 install --user matplotlib
```

```
Requirement already satisfied: matplotlib in /home/emr-
notebook/.local/lib/python3.7/site-packages
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.7/site-packages (from matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/emr-
notebook/.local/lib/python3.7/site-packages (from matplotlib)
Requirement already satisfied: pyparsing>=2.2.1 in /home/emr-
notebook/.local/lib/python3.7/site-packages (from matplotlib)
Requirement already satisfied: numpy>=1.16 in /usr/local/lib64/python3.7/site-
packages (from matplotlib)
Requirement already satisfied: pillow>=6.2.0 in /home/emr-
notebook/.local/lib/python3.7/site-packages (from matplotlib)
Requirement already satisfied: cycler>=0.10 in /home/emr-
notebook/.local/lib/python3.7/site-packages (from matplotlib)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-dateutil>=2.7->matplotlib)
```

1 Group 10 Craigslist Used Car Listing Analysis

1.0.1 By: Patrick Garrido and Shayan Agahi

2 Preprocessing

[2]:
```
#Project Objective: Create a Price prediction based off of Odometer value,␣
↪cylinders, fuel, type, transmission and age. Since the cylinders, fuel, type␣
↪and transmission will be categorical values, we will have to create dummies␣
↪for these variables. Additionally, we will be creating a classification␣
↪model to predict likelihood of a car being a specific type (bus, coupe,␣
↪hatchback, mini-van, offroad, pickup, sedan, SUV, truck, van, wagon) based␣
↪on the features that we give it. For our price prediction based of␣
↪regression we will be using "Price" as the dependent variable. As for the␣
↪classification model, which will have to run two different models (logistic␣
↪regression and random forest), we will be using "Type" as the dependent␣
↪variable. The feature columns we will be using are as follows:
```

```
#id: Unique ID for each record
#region: Craigslist region of the car listing
#price: Selling price of the car
#year: Year of manufacturing
#manufacturer: Manufacturing company of the car
#model: Specific car model
#condition: The condition of the vehicle
#cylinders: Number of cylinders in the vehicle
#fuel: Fuel type of the vehicle
#odometer: Numeric value of the miles driven on the vehicle
#type: Categorical type that the vehicle falls into
#state: State in the United States that the car is listed in
```

```
VBox()
```

```
Starting Spark application
```

```
<IPython.core.display.HTML object>
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
SparkSession available as 'spark'.
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[3]:
```
#Install matplotlib


sc.install_pypi_package("matplotlib", "https://pypi.org/simple")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
Collecting matplotlib
  Using cached https://files.pythonhosted.org/packages/ce/63/74c0b6184b6b169b121
bb72458818ee60a7d7c436d7b1907bd5874188c55/matplotlib-3.4.1-cp37-cp37m-manylinux1
_x86_64.whl
Requirement already satisfied: numpy>=1.16 in /usr/local/lib64/python3.7/site-
packages (from matplotlib)
Collecting pyparsing>=2.2.1 (from matplotlib)
  Using cached https://files.pythonhosted.org/packages/8a/bb/488841f56197b13700a
fd5658fc279a2025a39e22449b7cf29864669b15d/pyparsing-2.4.7-py2.py3-none-any.whl
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.7/site-packages (from matplotlib)
```

```
Collecting pillow>=6.2.0 (from matplotlib)
  Using cached https://files.pythonhosted.org/packages/33/34/542152297dcc6c47a9d
cb0685eac6d652d878ed3cea83bf2b23cb988e857/Pillow-8.2.0-cp37-cp37m-manylinux1_x86
_64.whl
Collecting cycler>=0.10 (from matplotlib)
  Using cached https://files.pythonhosted.org/packages/f7/d2/e07d3ebb2bd7af69644
0ce7e754c59dd546ffe1bbe732c8ab68b9c834e61/cycler-0.10.0-py2.py3-none-any.whl
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Using cached https://files.pythonhosted.org/packages/d2/46/231de802ade4225b76b
96cffe419cf3ce52bbe92e3b092cf12db7d11c207/kiwisolver-1.3.1-cp37-cp37m-manylinux1
_x86_64.whl
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-dateutil>=2.7->matplotlib)
Installing collected packages: pyparsing, pillow, cycler, kiwisolver, matplotlib
Successfully installed cycler-0.10.0 kiwisolver-1.3.1 matplotlib-3.4.1
pillow-8.2.0 pyparsing-2.4.7
```

[4]:
```
#Import pyspark, pyspark sql, sql.functions, pandas and matplotlib

import pyspark
import pyspark.sql as sql
import pyspark.sql.functions as fn
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

2.1 Importing Dataset and Initial Cleaning

[5]:
```
#import dataset into pyspark dataframe

df = spark.read.option("header",True)\
    .csv('s3://pjgarrido-cis4567-project-bucket/Dataset/vehicles.csv')\

#visualize dataframe output

df.show()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+---+----------+------------------+------+------------------+-----+------+--
```

```
----------+--------------------+---------+-----------+------+--------+----------
--+------------+-----------------+-----+---------+------+-----------+-----------
---------+--------------------+--------------------+-----------------+---------
-----+--------------------+
|_c0|        id|                 url|region|          region_url|price|
year|manufacturer|               model|condition|  cylinders|
fuel|odometer|title_status|transmission|              VIN|drive|     size|
type|paint_color|           image_url|         description|               state|
lat|          long|        posting_date|
+---+----------+--------------------+------+--------------------+-----+------+--
----------+--------------------+---------+-----------+------+--------+----------
--+------------+-----------------+-----+---------+------+-----------+-----------
---------+--------------------+--------------------+-----------------+---------
-----+--------------------+
|  0|7240372487|https://auburn.cr…|auburn|https://auburn.cr…|35990|2010.0|
chevrolet|corvette grand sport|     good|8 cylinders|   gas| 32742.0|
clean|       other|1G1YU3DW1A5106980|  rwd|     null| other|
null|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-12-02T08:11:…|
|  1|7240309422|https://auburn.cr…|auburn|https://auburn.cr…| 7500|2014.0|
hyundai|              sonata|excellent|4 cylinders|   gas| 93600.0|       clean|
automatic|5NPEC4AB0EH813529|  fwd|     null| sedan|
null|https://images.cr…|I'll move to anot…|                  al|
32.5475|      -85.4682|2020-12-02T02:11:…|
|  2|7240224296|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2006.0|
bmw|             x3 3.0i|     good|6 cylinders|   gas| 87046.0|       clean|
automatic|             null| null|     null|   SUV|
blue|https://images.cr…|Clean 2006 BMW X3…|                  al|
32.616807|    -85.464149|2020-12-01T19:50:…|
|  3|7240103965|https://auburn.cr…|auburn|https://auburn.cr…| 2000|1974.0|
chevrolet|                c-10|     good|4 cylinders|   gas|190000.0|
clean|   automatic|             null|  rwd|full-size|pickup|
blue|https://images.cr…|1974 chev. truck …|                  al|
32.8616|      -85.2161|2020-12-01T15:54:…|
|  4|7239983776|https://auburn.cr…|auburn|https://auburn.cr…|19500|2005.0|
ford|         f350 lariat|excellent|8 cylinders|diesel|116000.0|        lien|
automatic|             null|  4wd|full-size|pickup|
blue|https://images.cr…|2005 Ford F350 La…|                  al|
32.5475|      -85.4682|2020-12-01T12:53:…|
|  5|7239776805|https://auburn.cr…|auburn|https://auburn.cr…|29590|2016.0|
toyota|tacoma double cab…|     good|6 cylinders|   gas| 33290.0|       clean|
other|3TMAZ5CN6GM020355| null|     null|pickup|
red|https://images.cr…|Carvana is the sa…|                 al|
32.59|        -85.48|2020-12-01T07:27:…|
|  6|7239425036|https://auburn.cr…|auburn|https://auburn.cr…|39990|2012.0|
ford|mustang shelby gt…|     good|8 cylinders|   gas|  9692.0|       clean|
other|1ZVBP8JS8C5240016|  rwd|     null| coupe|
blue|https://images.cr…|Carvana is the sa…|                  al|
```

```
32.59|        -85.48|2020-11-30T13:34:…|
|  7|7238667661|https://auburn.cr…|auburn|https://auburn.cr…|41990|2012.0|
chevrolet| camaro zl1 coupe 2d|     good|8 cylinders|   gas|  2778.0|
clean|       other|2G1FS1EP4C9800609|  rwd|     null| coupe|
red|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-29T07:39:…|
|  8|7238127696|https://auburn.cr…|auburn|https://auburn.cr…|31990|2017.0|
jeep|wrangler unlimite…|     good|6 cylinders|   gas| 29614.0|       clean|
other|1C4BJWDG9HL725235|  4wd|     null| other|
null|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-28T07:21:…|
|  9|7237779886|https://auburn.cr…|auburn|https://auburn.cr…|  490|2019.0|
ford|transit connect w…|excellent|4 cylinders|   gas|  4775.0|       clean|
automatic|NM0GE9F22K1398142| null|     null|   van|
null|https://images.cr…|2019 Ford Transit…|                  al|
36.967357|   -122.024254|2020-11-27T12:42:…|
| 10|7237759157|https://auburn.cr…|auburn|https://auburn.cr…|27500|2012.0|
ford|               f-250|excellent|       null|diesel|189000.0|       clean|
automatic|             null|  4wd|     null|pickup|
silver|https://images.cr…|189k miles Leathe…|                  al|
32.639|      -85.3803|2020-11-27T12:12:…|
| 11|7237595428|https://auburn.cr…|auburn|https://auburn.cr…|36990|2013.0|
bmw|         m3 coupe 2d|     good|8 cylinders|   gas| 50956.0|       clean|
other|WBSKG9C51DE799269|  rwd|     null| coupe|
black|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-27T07:21:…|
| 12|7237366792|https://auburn.cr…|auburn|https://auburn.cr…|24990|2016.0|
ram|1500 crew cab slt…|     good|6 cylinders| other| 57926.0|       clean|
other|1C6RR6LG9GS331867| null|     null|pickup|
null|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-26T12:50:…|
| 13|7237318515|https://auburn.cr…|auburn|https://auburn.cr…| 5995|2010.0|
hyundai|              tucson|     null|       null|   gas|126000.0|       clean|
automatic|             null| null|     null|  null|
null|https://images.cr…|2010 Tucson AWD w…|
al|32.623298999999996|    -85.481787|2020-11-26T10:57:…|
| 14|7237009212|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2003.0|
ford|          expedition|     good|8 cylinders|   gas|177000.0|       clean|
automatic|             null|  rwd|full-size|   SUV|
blue|https://images.cr…|2003 Ford Expedit…|                  al|
33.1512|      -85.3722|2020-11-25T14:42:…|
| 15|7236904120|https://auburn.cr…|auburn|https://auburn.cr…|38500|  null|
null|                 500|     null|8 cylinders|   gas| 28246.0|       clean|
automatic|1C6RREMT7KN655834|  rwd|     null|pickup|
white|https://images.cr…|"2019 *Ram* *1500…|500Call Us Today!…|
one owner| Florida truck|      Big horn Sport|
| 16|7236744893|https://auburn.cr…|auburn|https://auburn.cr…|33990|2012.0|
chevrolet|corvette grand sport|     good|8 cylinders|   gas| 49245.0|
```

```
clean|   automatic|1G1YW3DWXC5106649|  rwd|     null| other|
white|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-25T07:08:…|
| 17|7236413365|https://auburn.cr…|auburn|https://auburn.cr…| 2650|1996.0|
toyota|            t100 4x4|     good|6 cylinders|   gas|414625.0|       clean|
automatic|             null|  4wd|     null|pickup|
blue|https://images.cr…|1996 Toyota T100 …|                  al|
32.7632|      -85.5144|2020-11-24T12:58:…|
| 18|7236210088|https://auburn.cr…|auburn|https://auburn.cr…|32990|2019.0|
ford|f150 supercrew ca…|     good|6 cylinders|   gas|  6910.0|       clean|
other|1FTEW1CP9KFB57643| null|     null|pickup|
silver|https://images.cr…|Carvana is the sa…|                  al|
32.59|        -85.48|2020-11-24T07:23:…|
| 19|7235942858|https://auburn.cr…|auburn|https://auburn.cr…|47000|2020.0|
jeep|           gladiator| like new|6 cylinders|   gas| 10500.0|       clean|
automatic|1C6JJTEG0LL206955|  4wd|full-size|pickup|
grey|https://images.cr…|I'm putting up fo…|                  al|
32.611442|    -85.481615|2020-11-23T15:02:…|
+---+----------+--------------------+------+--------------------+-----+------+--
----------+--------------------+---------+-----------+------+--------+----------
--+------------+-----------------+-----+---------+------+-----------+-----------
---------+--------------------+--------------------+-----------------+---------
-----+--------------------+
only showing top 20 rows
```

[6]:
```
#Convert str datatypes to int for numeric variables for filtering

df = df.withColumn("year", df['year'].cast('int'))
df = df.withColumn("price",df['price'].cast('int'))
df = df.withColumn("odometer", df['odometer'].cast('int'))
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[7]:
```
#filter out values above thresholds to remove non-numeric values

df_filtered=df.filter((fn.col('year') <= 2021) & (fn.col('price') <=␣
↪3615215112) & (fn.col('odometer') <= 3615215112))
df_filtered.show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+---+----------+--------------------+------+--------------------+-----+----+----
```

```
--------+-------------------+--------+----------+------+--------+-----------
+-----------+----------------+-----+--------+------+----------+-------------
-------+------------------+-----+----------------+----------+-------------
------+
|_c0|        id|                 url|region|
region_url|price|year|manufacturer|               model|condition|  cylinders|
fuel|odometer|title_status|transmission|             VIN|drive|     size|
type|paint_color|           image_url|         description|state|
lat|       long|        posting_date|
+---+----------+--------------------+------+--------------------+-----+----+----
--------+------------------+--------+----------+------+--------+-----------
+-----------+----------------+-----+--------+------+----------+-------------
-------+------------------+-----+----------------+----------+-------------
------+
|  0|7240372487|https://auburn.cr…|auburn|https://auburn.cr…|35990|2010|
chevrolet|corvette grand sport|     good|8 cylinders|   gas|   32742|
clean|       other|1G1YU3DW1A5106980|  rwd|     null| other|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-12-02T08:11:…|
|  1|7240309422|https://auburn.cr…|auburn|https://auburn.cr…| 7500|2014|
hyundai|              sonata|excellent|4 cylinders|   gas|   93600|       clean|
automatic|5NPEC4AB0EH813529|  fwd|     null| sedan|
null|https://images.cr…|I'll move to anot…|   al|           32.5475|
-85.4682|2020-12-02T02:11:…|
|  2|7240224296|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2006|
bmw|             x3 3.0i|     good|6 cylinders|   gas|   87046|       clean|
automatic|             null| null|     null|   SUV|
blue|https://images.cr…|Clean 2006 BMW X3…|   al|         32.616807|
-85.464149|2020-12-01T19:50:…|
|  3|7240103965|https://auburn.cr…|auburn|https://auburn.cr…| 2000|1974|
chevrolet|                c-10|     good|4 cylinders|   gas|  190000|
clean|   automatic|             null|  rwd|full-size|pickup|
blue|https://images.cr…|1974 chev. truck …|   al|           32.8616|
-85.2161|2020-12-01T15:54:…|
|  4|7239983776|https://auburn.cr…|auburn|https://auburn.cr…|19500|2005|
ford|         f350 lariat|excellent|8 cylinders|diesel|  116000|        lien|
automatic|             null|  4wd|full-size|pickup|
blue|https://images.cr…|2005 Ford F350 La…|   al|           32.5475|
-85.4682|2020-12-01T12:53:…|
|  5|7239776805|https://auburn.cr…|auburn|https://auburn.cr…|29590|2016|
toyota|tacoma double cab…|     good|6 cylinders|   gas|   33290|       clean|
other|3TMAZ5CN6GM020355| null|     null|pickup|
red|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-12-01T07:27:…|
|  6|7239425036|https://auburn.cr…|auburn|https://auburn.cr…|39990|2012|
ford|mustang shelby gt…|     good|8 cylinders|   gas|    9692|       clean|
other|1ZVBP8JS8C5240016|  rwd|     null| coupe|
blue|https://images.cr…|Carvana is the sa…|   al|             32.59|
```

```
-85.48|2020-11-30T13:34:…|
|  7|7238667661|https://auburn.cr…|auburn|https://auburn.cr…|41990|2012|
chevrolet| camaro zl1 coupe 2d|     good|8 cylinders|   gas|    2778|
clean|       other|2G1FS1EP4C9800609|  rwd|     null| coupe|
red|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-29T07:39:…|
|  8|7238127696|https://auburn.cr…|auburn|https://auburn.cr…|31990|2017|
jeep|wrangler unlimite…|     good|6 cylinders|   gas|   29614|       clean|
other|1C4BJWDG9HL725235|  4wd|     null| other|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-28T07:21:…|
|  9|7237779886|https://auburn.cr…|auburn|https://auburn.cr…|  490|2019|
ford|transit connect w…|excellent|4 cylinders|   gas|    4775|       clean|
automatic|NM0GE9F22K1398142| null|     null|   van|
null|https://images.cr…|2019 Ford Transit…|   al|
36.967357|-122.024254|2020-11-27T12:42:…|
| 10|7237759157|https://auburn.cr…|auburn|https://auburn.cr…|27500|2012|
ford|               f-250|excellent|       null|diesel|  189000|       clean|
automatic|             null|  4wd|     null|pickup|
silver|https://images.cr…|189k miles Leathe…|   al|            32.639|
-85.3803|2020-11-27T12:12:…|
| 11|7237595428|https://auburn.cr…|auburn|https://auburn.cr…|36990|2013|
bmw|         m3 coupe 2d|     good|8 cylinders|   gas|   50956|       clean|
other|WBSKG9C51DE799269|  rwd|     null| coupe|
black|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-27T07:21:…|
| 12|7237366792|https://auburn.cr…|auburn|https://auburn.cr…|24990|2016|
ram|1500 crew cab slt…|     good|6 cylinders| other|   57926|       clean|
other|1C6RR6LG9GS331867| null|     null|pickup|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-26T12:50:…|
| 13|7237318515|https://auburn.cr…|auburn|https://auburn.cr…| 5995|2010|
hyundai|              tucson|     null|       null|   gas|  126000|       clean|
automatic|             null| null|     null|  null|
null|https://images.cr…|2010 Tucson AWD w…|   al|32.623298999999996|
-85.481787|2020-11-26T10:57:…|
| 14|7237009212|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2003|
ford|          expedition|     good|8 cylinders|   gas|  177000|       clean|
automatic|             null|  rwd|full-size|   SUV|
blue|https://images.cr…|2003 Ford Expedit…|   al|           33.1512|
-85.3722|2020-11-25T14:42:…|
| 16|7236744893|https://auburn.cr…|auburn|https://auburn.cr…|33990|2012|
chevrolet|corvette grand sport|     good|8 cylinders|   gas|   49245|
clean|   automatic|1G1YW3DWXC5106649|  rwd|     null| other|
white|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-25T07:08:…|
| 17|7236413365|https://auburn.cr…|auburn|https://auburn.cr…| 2650|1996|
toyota|            t100 4x4|     good|6 cylinders|   gas|  414625|       clean|
```

```
automatic|             null|  4wd|     null|pickup|
blue|https://images.cr…|1996 Toyota T100 …|   al|           32.7632|
-85.5144|2020-11-24T12:58:…|
| 18|7236210088|https://auburn.cr…|auburn|https://auburn.cr…|32990|2019|
ford|f150 supercrew ca…|     good|6 cylinders|   gas|    6910|       clean|
other|1FTEW1CP9KFB57643| null|     null|pickup|
silver|https://images.cr…|Carvana is the sa…|   al|            32.59|
-85.48|2020-11-24T07:23:…|
| 19|7235942858|https://auburn.cr…|auburn|https://auburn.cr…|47000|2020|
jeep|           gladiator| like new|6 cylinders|   gas|   10500|       clean|
automatic|1C6JJTEG0LL206955|  4wd|full-size|pickup|
grey|https://images.cr…|I'm putting up fo…|   al|         32.611442|
-85.481615|2020-11-23T15:02:…|
| 21|7235872843|https://auburn.cr…|auburn|https://auburn.cr…| 6500|2010|
null|            bmw328xi|excellent|6 cylinders|   gas|  149786|       clean|
automatic|WBAPK5C59AA647356|  fwd|     null| sedan|
black|https://images.cr…|Clean vehicle, no…|   al|         32.951775|
-85.94718|2020-11-23T13:20:…|
+---+----------+--------------------+------+--------------------+-----+----+----
--------+--------------------+---------+-----------+------+--------+-----------
+-----------+-----------------+-----+---------+------+-----------+-------------
-------+--------------------+-----+-----------------+-----------+-------------
------+
only showing top 20 rows
```

[8]:
```
#filter out 0 values

df_zeroes=df_filtered.filter((fn.col('price') > 0) & (fn.col('odometer') > 0))
df_zeroes.show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

+---+----------+--------------------+------+--------------------+-----+----+----
--------+--------------------+---------+-----------+------+--------+-----------
+-----------+-----------------+-----+---------+------+-----------+-------------
-------+--------------------+-----+-----------------+-----------+-------------
------+
|_c0|        id|                 url|region|
region_url|price|year|manufacturer|              model|condition|  cylinders|
fuel|odometer|title_status|transmission|             VIN|drive|     size|
type|paint_color|           image_url|         description|state|
lat|       long|        posting_date|
+---+----------+--------------------+------+--------------------+-----+----+----
--------+--------------------+---------+-----------+------+--------+-----------
+-----------+-----------------+-----+---------+------+-----------+-------------
```

```
-------+------------------+-----+-----------------+----------+-------------
------+
|  0|7240372487|https://auburn.cr…|auburn|https://auburn.cr…|35990|2010|
chevrolet|corvette grand sport|     good|8 cylinders|   gas|   32742|
clean|       other|1G1YU3DW1A5106980|  rwd|     null| other|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-12-02T08:11:…|
|  1|7240309422|https://auburn.cr…|auburn|https://auburn.cr…| 7500|2014|
hyundai|              sonata|excellent|4 cylinders|   gas|   93600|       clean|
automatic|5NPEC4AB0EH813529|  fwd|     null| sedan|
null|https://images.cr…|I'll move to anot…|   al|           32.5475|
-85.4682|2020-12-02T02:11:…|
|  2|7240224296|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2006|
bmw|             x3 3.0i|     good|6 cylinders|   gas|   87046|       clean|
automatic|             null| null|     null|   SUV|
blue|https://images.cr…|Clean 2006 BMW X3…|   al|         32.616807|
-85.464149|2020-12-01T19:50:…|
|  3|7240103965|https://auburn.cr…|auburn|https://auburn.cr…| 2000|1974|
chevrolet|                c-10|     good|4 cylinders|   gas|  190000|
clean|   automatic|             null|  rwd|full-size|pickup|
blue|https://images.cr…|1974 chev. truck …|   al|           32.8616|
-85.2161|2020-12-01T15:54:…|
|  4|7239983776|https://auburn.cr…|auburn|https://auburn.cr…|19500|2005|
ford|         f350 lariat|excellent|8 cylinders|diesel|  116000|        lien|
automatic|             null|  4wd|full-size|pickup|
blue|https://images.cr…|2005 Ford F350 La…|   al|           32.5475|
-85.4682|2020-12-01T12:53:…|
|  5|7239776805|https://auburn.cr…|auburn|https://auburn.cr…|29590|2016|
toyota|tacoma double cab…|     good|6 cylinders|   gas|   33290|       clean|
other|3TMAZ5CN6GM020355| null|     null|pickup|
red|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-12-01T07:27:…|
|  6|7239425036|https://auburn.cr…|auburn|https://auburn.cr…|39990|2012|
ford|mustang shelby gt…|     good|8 cylinders|   gas|    9692|       clean|
other|1ZVBP8JS8C5240016|  rwd|     null| coupe|
blue|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-30T13:34:…|
|  7|7238667661|https://auburn.cr…|auburn|https://auburn.cr…|41990|2012|
chevrolet| camaro zl1 coupe 2d|     good|8 cylinders|   gas|    2778|
clean|       other|2G1FS1EP4C9800609|  rwd|     null| coupe|
red|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-29T07:39:…|
|  8|7238127696|https://auburn.cr…|auburn|https://auburn.cr…|31990|2017|
jeep|wrangler unlimite…|     good|6 cylinders|   gas|   29614|       clean|
other|1C4BJWDG9HL725235|  4wd|     null| other|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-28T07:21:…|
|  9|7237779886|https://auburn.cr…|auburn|https://auburn.cr…|  490|2019|
```

```
ford|transit connect w…|excellent|4 cylinders|   gas|    4775|       clean|
automatic|NM0GE9F22K1398142| null|     null|   van|
null|https://images.cr…|2019 Ford Transit…|   al|
36.967357|-122.024254|2020-11-27T12:42:…|
| 10|7237759157|https://auburn.cr…|auburn|https://auburn.cr…|27500|2012|
ford|               f-250|excellent|       null|diesel|  189000|       clean|
automatic|             null|  4wd|     null|pickup|
silver|https://images.cr…|189k miles Leathe…|   al|            32.639|
-85.3803|2020-11-27T12:12:…|
| 11|7237595428|https://auburn.cr…|auburn|https://auburn.cr…|36990|2013|
bmw|         m3 coupe 2d|     good|8 cylinders|   gas|   50956|       clean|
other|WBSKG9C51DE799269|  rwd|     null| coupe|
black|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-27T07:21:…|
| 12|7237366792|https://auburn.cr…|auburn|https://auburn.cr…|24990|2016|
ram|1500 crew cab slt…|     good|6 cylinders| other|   57926|       clean|
other|1C6RR6LG9GS331867| null|     null|pickup|
null|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-26T12:50:…|
| 13|7237318515|https://auburn.cr…|auburn|https://auburn.cr…| 5995|2010|
hyundai|              tucson|     null|       null|   gas|  126000|       clean|
automatic|             null| null|     null|  null|
null|https://images.cr…|2010 Tucson AWD w…|   al|32.623298999999996|
-85.481787|2020-11-26T10:57:…|
| 14|7237009212|https://auburn.cr…|auburn|https://auburn.cr…| 4900|2003|
ford|          expedition|     good|8 cylinders|   gas|  177000|       clean|
automatic|             null|  rwd|full-size|   SUV|
blue|https://images.cr…|2003 Ford Expedit…|   al|           33.1512|
-85.3722|2020-11-25T14:42:…|
| 16|7236744893|https://auburn.cr…|auburn|https://auburn.cr…|33990|2012|
chevrolet|corvette grand sport|     good|8 cylinders|   gas|   49245|
clean|   automatic|1G1YW3DWXC5106649|  rwd|     null| other|
white|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-25T07:08:…|
| 17|7236413365|https://auburn.cr…|auburn|https://auburn.cr…| 2650|1996|
toyota|            t100 4x4|     good|6 cylinders|   gas|  414625|       clean|
automatic|             null|  4wd|     null|pickup|
blue|https://images.cr…|1996 Toyota T100 …|   al|           32.7632|
-85.5144|2020-11-24T12:58:…|
| 18|7236210088|https://auburn.cr…|auburn|https://auburn.cr…|32990|2019|
ford|f150 supercrew ca…|     good|6 cylinders|   gas|    6910|       clean|
other|1FTEW1CP9KFB57643| null|     null|pickup|
silver|https://images.cr…|Carvana is the sa…|   al|             32.59|
-85.48|2020-11-24T07:23:…|
| 19|7235942858|https://auburn.cr…|auburn|https://auburn.cr…|47000|2020|
jeep|           gladiator| like new|6 cylinders|   gas|   10500|       clean|
automatic|1C6JJTEG0LL206955|  4wd|full-size|pickup|
grey|https://images.cr…|I'm putting up fo…|   al|         32.611442|
```

```
-85.481615|2020-11-23T15:02:…|
| 21|7235872843|https://auburn.cr…|auburn|https://auburn.cr…| 6500|2010|
null|            bmw328xi|excellent|6 cylinders|   gas|  149786|       clean|
automatic|WBAPK5C59AA647356|  fwd|     null| sedan|
black|https://images.cr…|Clean vehicle, no…|   al|         32.951775|
-85.94718|2020-11-23T13:20:…|
+---+----------+--------------------+------+--------------------+-----+----+----
--------+--------------------+---------+----------+------+--------+------------
+------------+-----------------+-----+---------+------+----------+-------------
-------+--------------------+-----+-----------------+----------+--------------
------+
only showing top 20 rows
```

[9]:
```
#Create new dataframe which only contains relevant features for modelling

new_df = df_zeroes.
 ↪select('id','region','price','year','manufacturer','model','condition','cylinders','fuel','

#visualize dataframe output

new_df.show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+----------+------+-----+----+------------+--------------------+---------+------
-----+------+--------+------+-----+
|        id|region|price|year|manufacturer|               model|condition|
cylinders|  fuel|odometer|  type|state|
+----------+------+-----+----+------------+--------------------+---------+------
-----+------+--------+------+-----+
|7240372487|auburn|35990|2010|   chevrolet|corvette grand sport|     good|8
cylinders|   gas|   32742| other|   al|
|7240309422|auburn| 7500|2014|     hyundai|              sonata|excellent|4
cylinders|   gas|   93600| sedan|   al|
|7240224296|auburn| 4900|2006|         bmw|             x3 3.0i|     good|6
cylinders|   gas|   87046|   SUV|   al|
|7240103965|auburn| 2000|1974|   chevrolet|                c-10|     good|4
cylinders|   gas|  190000|pickup|   al|
|7239983776|auburn|19500|2005|        ford|         f350 lariat|excellent|8
cylinders|diesel|  116000|pickup|   al|
|7239776805|auburn|29590|2016|      toyota|tacoma double cab…|     good|6
cylinders|   gas|   33290|pickup|   al|
|7239425036|auburn|39990|2012|        ford|mustang shelby gt…|     good|8
cylinders|   gas|    9692| coupe|   al|
|7238667661|auburn|41990|2012|   chevrolet| camaro zl1 coupe 2d|     good|8
```

```
cylinders|   gas|    2778| coupe|   al|
|7238127696|auburn|31990|2017|        jeep|wrangler unlimite…|     good|6
cylinders|   gas|   29614| other|   al|
|7237779886|auburn|  490|2019|        ford|transit connect w…|excellent|4
cylinders|   gas|    4775|   van|   al|
|7237759157|auburn|27500|2012|        ford|               f-250|excellent|
null|diesel|  189000|pickup|   al|
|7237595428|auburn|36990|2013|         bmw|         m3 coupe 2d|     good|8
cylinders|   gas|   50956| coupe|   al|
|7237366792|auburn|24990|2016|         ram|1500 crew cab slt…|     good|6
cylinders| other|   57926|pickup|   al|
|7237318515|auburn| 5995|2010|     hyundai|              tucson|     null|
null|   gas|  126000|  null|   al|
|7237009212|auburn| 4900|2003|        ford|          expedition|     good|8
cylinders|   gas|  177000|   SUV|   al|
|7236744893|auburn|33990|2012|   chevrolet|corvette grand sport|     good|8
cylinders|   gas|   49245| other|   al|
|7236413365|auburn| 2650|1996|      toyota|            t100 4x4|     good|6
cylinders|   gas|  414625|pickup|   al|
|7236210088|auburn|32990|2019|        ford|f150 supercrew ca…|     good|6
cylinders|   gas|    6910|pickup|   al|
|7235942858|auburn|47000|2020|        jeep|           gladiator| like new|6
cylinders|   gas|   10500|pickup|   al|
|7235872843|auburn| 6500|2010|        null|            bmw328xi|excellent|6
cylinders|   gas|  149786| sedan|   al|
+----------+------+-----+----+------------+--------------------+---------+------
------+------+--------+------+-----+
only showing top 20 rows
```

[10]:
```
#Convert str datatypes to int for numeric variables for new dataframe

new_df = new_df.withColumn("year", df['year'].cast('int'))
new_df = new_df.withColumn("price",df['price'].cast('int'))
new_df = new_df.withColumn("odometer", df['odometer'].cast('int'))
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[11]:
```
#Confirm datatype changes

new_df.printSchema()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
root
 |-- id: string (nullable = true)
 |-- region: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- manufacturer: string (nullable = true)
 |-- model: string (nullable = true)
 |-- condition: string (nullable = true)
 |-- cylinders: string (nullable = true)
 |-- fuel: string (nullable = true)
 |-- odometer: integer (nullable = true)
 |-- type: string (nullable = true)
 |-- state: string (nullable = true)
```

2.2 Filtering Duplicate Values

[12]:
```
#Take a count of values and compare it to the count of distinct values to find␣
 ↪potential duplicates

new_df.count(), new_df.distinct().count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
(372998, 372998)
```

[13]:
```
#According to our count, we have no duplicate values. Just to make sure, we␣
 ↪will search for potential records
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[14]:
```
#Find duplicated rows

(
    new_df
    .groupby(new_df.columns)
    .count()
    .filter('count > 1')
    .show()
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+---+------+-----+----+------------+-----+---------+---------+----+--------+----
+-----+-----+
| id|region|price|year|manufacturer|model|condition|cylinders|fuel|odometer|type
|state|count|
+---+------+-----+----+------------+-----+---------+---------+----+--------+----
+-----+-----+
+---+------+-----+----+------------+-----+---------+---------+----+--------+----
+-----+-----+
```

[15]:
```
#Remove the duplicated rows

distinctDf = new_df.dropDuplicates()
distinctDf.count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
372998
```

[16]:
```
#Display count of rows with no id

no_ids = (
    distinctDf
    .select([col for col in distinctDf.columns if col != 'id'])
)

no_ids.count(), no_ids.distinct().count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
(372998, 330749)
```

[17]:
```
#Show which row is duplicated

(
    distinctDf
    .groupby([col for col in distinctDf.columns if col != 'id'])
    .count()
    .filter('count > 1')
    .show()
)
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

+---------------+------+----+------------+-------------------+---------+------
-----+------+--------+------+-------------------+-----+
|          region| price|year|manufacturer|              model|condition|
cylinders|  fuel|odometer|  type|              state|count|
+---------------+------+----+------------+-------------------+---------+------
-----+------+--------+------+-------------------+-----+
|         phoenix|   500|2016|      nissan|              rouge| like new|4
cylinders|   gas|   77000|   SUV|                 az|    2|
|          eugene|  9995|2000|      toyota|      tacoma xtracab|     null|
null|   gas|  202395| truck| depending on spe…|    2|
|         phoenix|   500|2012|        fiat|                500|     null|4
cylinders|   gas|  100000| other|                 az|    6|
|        syracuse|  7495|2009|   chevrolet|            express|     good|8
cylinders|   gas|  183490|   van|                 ny|    3|
|          boston| 14500|2016|       honda|         civic sedan|     null|4
cylinders|   gas|   54723| sedan|                 ma|    2|
|   santa barbara| 11900|2013|      toyota|             corolla|     good|
null|   gas|   38584|  null|                 ca|    2|
|          eugene|105900|2020|        jeep|gladiator hellcat…|     null|6
cylinders|   gas|    4772|pickup|772MPG: 16 City /…|    3|
|       wenatchee| 16999|2016|      nissan|rogue s awd gas s…|     null|
null|   gas|   45048|   SUV| call 425-358-399…|    2|
|    indianapolis|  1500|2004|    chrysler|             sebring|     good|6
cylinders|   gas|  133000|  null|                 in|    2|
| medford-ashland|  7495|2015|       honda|             odyssey|     null|6
cylinders|   gas|  185209| other|                 or|    2|
|      charleston|  3500|2001|       acura|                 tl|     null|6
cylinders|   gas|  139303| sedan|                 wv|    3|
|      greensboro| 13499|2016|       dodge|   grand caravan sxt|     null|
null|   gas|   72850|  null|                 nc|    3|
|        richmond|  3000|2003|      toyota|              prius|     null|4
cylinders|hybrid|  122450| sedan|                 va|    4|
|     little rock| 66682|2019|         gmc|       sierra 2500hd|     null|
null|diesel|   31012|pickup|                 ar|    2|
|       charlotte|  7700|2011|  volkswagen|                 cc|excellent|4
cylinders|   gas|  117198| sedan|                 nc|    2|
|          maine| 13990|2017|   chevrolet|      equinox ls awd|excellent|4
cylinders|   gas|   74000|  null|                 me|    3|
|corvallis/albany|   578|2016|   chevrolet|      silverado 1500|     null|8
cylinders|   gas|   36848| truck|                 or|    2|
|          austin|  4999|2007|      nissan|             murano|     null|
null|   gas|  143000|  null|                 tx|    2|
|    monterey bay|  4995|2005|      toyota|             corolla|     good|4
```

```
cylinders|   gas|  157569| sedan|                  ca|    3|
|         houston| 20900|2014|       dodge|             durango|excellent|
null|   gas|   61121|   SUV|                  tx|    2|
+---------------+------+----+-----------+--------------------+---------+------
-----+------+--------+------+--------------------+-----+
only showing top 20 rows
```

[18]:
```
#Remove the duplicated record

id_removed = distinctDf.dropDuplicates(
    subset = [col for col in distinctDf.columns if col != 'id']
)
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[19]:
```
#count of removed values

id_removed.count()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


330749
```

[20]:
```
#Determine if there any duplicated IDs

id_removed.agg(
      fn.count('id').alias('CountOfIDs')
    , fn.countDistinct('id').alias('CountOfDistinctIDs')
).show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+----------+------------------+
|CountOfIDs|CountOfDistinctIDs|
+----------+------------------+
|    330749|            330749|
+----------+------------------+
```

[21]:
```
#Find which IDs are duplicated

(
    id_removed
    .groupby('id')
    .count()
    .filter('count > 1')
    .show()
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+---+-----+
| id|count|
+---+-----+
+---+-----+
```

[22]:
```
#Generate new id to solve the problem of duplicate ids

new_id = (
    id_removed
    .select(
        [fn.monotonically_increasing_id().alias('New_Id')] +
        [col for col in id_removed.columns if col != 'Id'])
)

new_id.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+------+----------+------------------+-----+----+------------+-----+---------+-
----------+------+--------+-----------+--------------------+
|New_Id|        id|            region|price|year|manufacturer|model|condition|
cylinders|  fuel|odometer|       type|               state|
+------+----------+------------------+-----+----+------------+-----+---------+-
----------+------+--------+-----------+--------------------+
|     0|7237761307|            eugene|28990|2003|         ram| null|     null|
null|diesel|   91567|     pickup|                null|
|     1|7237366028|  colorado springs|19988|2009|         ram| null|     null|
null|   gas|   93562|     pickup|                null|
|     2|7239170981|    fredericksburg| 3995|2009|       honda| null|     null|4
cylinders|   gas|  238000|       null|  C.R.Garland Aut…|
```

```
|     3|7235356070| mcallen / edinburg| 4500|2004|      nissan| null|excellent|6
cylinders|   gas|  137451|        SUV|  cómodo no dejan…|
|     4|7240709284|          las vegas| 5900|2004|       ford| null|excellent|
null|diesel|  250000|       null|           Automatic|
|     5|7226776086|             wyoming|24999|2006|         ram| null|     null|
null|diesel|  160186|     pickup| Inc    Year: 200…|
|     6|7237477082|             wyoming|36500|2012|         ram| null|     null|
null|diesel|  126941|     pickup| Inc    Year: 201…|
|     7|7234532298|     salt lake city|18999|2003|         ram| null|     null|
null|diesel|  188745|     pickup| Inc - (970) 456-…|
|     8|7236102344|     salt lake city| 9000|2003|         ram| null|     null|
null|diesel|  215370|     pickup| Inc - (970) 456-…|
|     9|7226975171|         greensboro| 6490|2011|       honda| null|     null|
null|   gas|  125081|       null| MAKE SURE TO CLI…|
|    10|7238072478|             eugene|34990|2012|         ram| null|     null|
null|diesel|  153860|     pickup| has a Clean titl…|
|    11|7230815357|        new orleans|45991|2014|       rover| null|     null|8
cylinders|   gas|   48100|       null| look no further!…|
|    12|7230276408|            boulder|22000|2007|   chevrolet| null|     good|8
cylinders|diesel|  268000|       null| low/high flow vo…|
|    13|7240535821|            phoenix|42900|1933|        ford| null|     null|
null|   gas|    5289|       null|289 MILES  * BELI…|
|    14|7229038332|          cleveland|19500|1969|     mercury| null|excellent|8
cylinders|   gas|   23233|convertible|500.00   email or…|
|    15|7239724042|dallas / fort worth| 4500|2000|         gmc| null|     good|8
cylinders|   gas|  190625|      truck|625 miles. Call m…|
|    16|7238449510|         montgomery| 1999|1989|   chevrolet| null|     good|6
cylinders|   gas|  200739|     pickup|                  al|
|    17|7240405045|        little rock| 8500|2002|       lexus| null|     null|
null|   gas|  155047|       null|                  ar|
|    18|7239986191|           prescott|47000|1955|   chevrolet| null|excellent|8
cylinders|   gas|    7539|      sedan|                  az|
|    19|7240265288|            phoenix|30000|2012|   chevrolet| null|     good|8
cylinders|   gas|   40000|       null|                  az|
+------+----------+-------------------+-----+----+------------+-----+---------+-
----------+------+--------+-----------+-------------------+
only showing top 20 rows
```

[23]:
```
#While going through our records, we found no duplicate values
#This could be attributed to the nature of Craigslist listings. Posters often␣
 ↪have wildly different listing styles
#While there are certainly minimum features required to get a sales post␣
 ↪listed, the individual nature of each user's post
#is a factor to consider as to why no two records share a similar id.
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

2.3 Missing Values

[24]:
```
#Examine if there are any rows with missing values

new_id.rdd.map(
           lambda row: (
                 row['id']
               , sum([c == None for c in row])
           )
       )\
       .filter(lambda el: el[1] >= 1)\
       .count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
184127
```

[25]:
```
#Count the number of missing values in each row

(
    spark.createDataFrame(
        new_id.rdd.map(
           lambda row: (
                 row['id']
               , sum([c == None for c in row])
           )
        )
        .filter(lambda el: el[1] >= 1)
        .collect()
        ,['id', 'CountMissing']
    )
    .orderBy('CountMissing', ascending=False)
    .show()
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+----------+------------+
|        id|CountMissing|
```

```
+----------+------------+
|7233516934|           5|
|7230183790|           5|
|7237642106|           5|
|7237642871|           5|
|7230184660|           5|
|7240936442|           5|
|7240936609|           5|
|7230183608|           5|
|7230994709|           5|
|7229704142|           5|
|7230194458|           5|
|7230217017|           5|
|7237615750|           5|
|7238087150|           5|
|7237621917|           5|
|7226333872|           5|
|7226370935|           5|
|7226370288|           5|
|7226515316|           5|
|7226369785|           5|
+----------+------------+
only showing top 20 rows
```

[26]:
```
#View a row with the most missing values

(
    new_id
    .where('id == 7230994709')
    .show()
)
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+------------+----------+---------+-----+----+------------+------------------+-
--------+---------+----+--------+----+-----+
|      New_Id|        id|   region|price|year|manufacturer|
model|condition|cylinders|fuel|odometer|type|state|
+------------+----------+---------+-----+----+------------+------------------+-
--------+---------+----+--------+----+-----+
|128849020002|7230994709|rochester| 5995|2011|        null|BLUE BIRD Blue Bird|
null|     null|null|   93000|null|   ny|
+------------+----------+---------+-----+----+------------+------------------+-
--------+---------+----+--------+----+-----+
```

[27]:
```
#Drop rows that have less than thresh NON-NULL values.

merc_out = new_id.dropna(thresh=4)
new_id.count(), merc_out.count()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

(330749, 330749)

[28]:
```
#Count proportion of missing values in each column

for k, v in sorted(
    merc_out.agg(*[
                (1 - (fn.count(c) / fn.count('*')))
                     .alias(c + '_miss')
                for c in merc_out.columns
           ])
        .collect()[0]
        .asDict()
        .items()
    , key=lambda el: el[1]
    , reverse=True
):
    print(k, v)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
condition_miss 0.3541537540551899
cylinders_miss 0.3224680951416331
type_miss 0.19414722342320012
manufacturer_miss 0.031347033551121894
state_miss 0.01169164532621414
model_miss 0.007498133025345521
fuel_miss 0.0055570840728165205
New_Id_miss 0.0
id_miss 0.0
region_miss 0.0
price_miss 0.0
year_miss 0.0
odometer_miss 0.0
```

[29]:
```
#Cylinder and Condition have a high proportion missing, at least 33%. While␣
 ↪useful information, we decided to remove them
#rather than try to impute values.

#Condition was another categorical variable which, while potentially useful,␣
 ↪was missing 35% of values. We decided to remove it
#as well

#Finally, we are interested in type, but are missing around 20% of values. We␣
 ↪may have to try to impute these if our modelling
#shows issues
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[30]:
```
#Drop condition and cylinder columns due to high proportion of missing values

dropped_df = merc_out.
 ↪select('id','year','region','price','year','manufacturer','model','fuel','odometer','type',
dropped_df.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+----------+----+------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
|        id|year|            region|price|year|manufacturer|model|
fuel|odometer|       type|               state|
+----------+----+------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
|7237761307|2003|            eugene|28990|2003|         ram| null|diesel|
91567|     pickup|                null|
|7237366028|2009|   colorado springs|19988|2009|         ram| null|   gas|
93562|     pickup|                null|
|7239170981|2009|     fredericksburg| 3995|2009|       honda| null|   gas|
238000|       null|  C.R.Garland Aut…|
|7235356070|2004| mcallen / edinburg| 4500|2004|      nissan| null|   gas|
137451|        SUV|  cómodo no dejan…|
|7240709284|2004|          las vegas| 5900|2004|        ford| null|diesel|
250000|       null|           Automatic|
|7226776086|2006|           wyoming|24999|2006|         ram| null|diesel|
160186|     pickup| Inc    Year: 200…|
|7237477082|2012|           wyoming|36500|2012|         ram| null|diesel|
```

```
126941|     pickup| Inc    Year: 201…|
|7234532298|2003|     salt lake city|18999|2003|         ram| null|diesel|
188745|     pickup| Inc - (970) 456-…|
|7236102344|2003|     salt lake city| 9000|2003|         ram| null|diesel|
215370|     pickup| Inc - (970) 456-…|
|7226975171|2011|         greensboro| 6490|2011|       honda| null|   gas|
125081|       null| MAKE SURE TO CLI…|
|7238072478|2012|             eugene|34990|2012|         ram| null|diesel|
153860|     pickup| has a Clean titl…|
|7230815357|2014|        new orleans|45991|2014|       rover| null|   gas|
48100|       null| look no further!…|
|7230276408|2007|            boulder|22000|2007|   chevrolet| null|diesel|
268000|       null| low/high flow vo…|
|7240535821|1933|            phoenix|42900|1933|        ford| null|   gas|
5289|       null|289 MILES  * BELI…|
|7229038332|1969|          cleveland|19500|1969|     mercury| null|   gas|
23233|convertible|500.00   email or…|
|7239724042|2000|dallas / fort worth| 4500|2000|         gmc| null|   gas|
190625|      truck|625 miles. Call m…|
|7238449510|1989|         montgomery| 1999|1989|   chevrolet| null|   gas|
200739|     pickup|                 al|
|7240405045|2002|        little rock| 8500|2002|       lexus| null|   gas|
155047|       null|                 ar|
|7239986191|1955|           prescott|47000|1955|   chevrolet| null|   gas|
7539|      sedan|                az|
|7240265288|2012|            phoenix|30000|2012|   chevrolet| null|   gas|
40000|       null|                az|
+----------+----+-------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
only showing top 20 rows
```

[31]:
```
#Create new dataframe to include imputed mean value for odometer

means = (
    dropped_df
    .agg(
          fn.mean(
              fn.col('odometer')
          ).alias('odometer')
    )
).toPandas().to_dict('records')[0]

means
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
{'odometer': 104789.3900057143}
```

[32]:
```
#Create new dataframe with imputed odometer values, using fillna() method

imputed = (
    dropped_df
    .fillna(means)
)

imputed.show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+----------+----+------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
|        id|year|            region|price|year|manufacturer|model|
fuel|odometer|       type|              state|
+----------+----+------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
|7237761307|2003|            eugene|28990|2003|         ram| null|diesel|
91567|     pickup|               null|
|7237366028|2009|  colorado springs|19988|2009|         ram| null|   gas|
93562|     pickup|               null|
|7239170981|2009|    fredericksburg| 3995|2009|       honda| null|   gas|
238000|       null|  C.R.Garland Aut…|
|7235356070|2004| mcallen / edinburg| 4500|2004|      nissan| null|   gas|
137451|        SUV|  cómodo no dejan…|
|7240709284|2004|         las vegas| 5900|2004|        ford| null|diesel|
250000|       null|           Automatic|
|7226776086|2006|           wyoming|24999|2006|         ram| null|diesel|
160186|     pickup| Inc    Year: 200…|
|7237477082|2012|           wyoming|36500|2012|         ram| null|diesel|
126941|     pickup| Inc    Year: 201…|
|7234532298|2003|    salt lake city|18999|2003|         ram| null|diesel|
188745|     pickup| Inc - (970) 456-…|
|7236102344|2003|    salt lake city| 9000|2003|         ram| null|diesel|
215370|     pickup| Inc - (970) 456-…|
|7226975171|2011|        greensboro| 6490|2011|       honda| null|   gas|
125081|       null| MAKE SURE TO CLI…|
|7238072478|2012|            eugene|34990|2012|         ram| null|diesel|
153860|     pickup| has a Clean titl…|
|7230815357|2014|       new orleans|45991|2014|       rover| null|   gas|
48100|       null| look no further!…|
|7230276408|2007|           boulder|22000|2007|   chevrolet| null|diesel|
268000|       null| low/high flow vo…|
```

```
|7240535821|1933|            phoenix|42900|1933|        ford| null|   gas|
5289|       null|289 MILES  * BELI…|
|7229038332|1969|          cleveland|19500|1969|     mercury| null|   gas|
23233|convertible|500.00   email or…|
|7239724042|2000|dallas / fort worth| 4500|2000|         gmc| null|   gas|
190625|      truck|625 miles. Call m…|
|7238449510|1989|         montgomery| 1999|1989|   chevrolet| null|   gas|
200739|     pickup|                 al|
|7240405045|2002|        little rock| 8500|2002|       lexus| null|   gas|
155047|       null|                 ar|
|7239986191|1955|           prescott|47000|1955|   chevrolet| null|   gas|
7539|      sedan|                az|
|7240265288|2012|            phoenix|30000|2012|   chevrolet| null|   gas|
40000|       null|                az|
+----------+----+------------------+-----+----+------------+-----+------+------
--+-----------+--------------------+
only showing top 20 rows
```

2.4 Descriptive Statistics

[33]:
```
#create list of features for numerical data

features = ['year','price','odometer']

#describe numeric features

descriptive_stats = imputed.describe(features)
descriptive_stats.show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+-------+-----------------+-----------------+------------------+
|summary|             year|            price|          odometer|
+-------+-----------------+-----------------+------------------+
|  count|           330749|           330749|            330749|
|   mean|2011.0366561954836|17763.716797329696| 104789.3900057143|
| stddev| 7.974047994873543| 329392.1782005149|3563048.0809303066|
|    min|             1900|                1|                 1|
|    max|             2021|        123456789|        2043755555|
+-------+-----------------+-----------------+------------------+
```

[34]:
```
#describe all columns

descriptive_stats_all = imputed.describe()
descriptive_stats_all.show()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+-------+------------------+-----------------+--------------------+-----------
-------+------------------+-----------+--------------------+------+-----------
-----+------+------------------+
|summary|                id|             year|              region|
price|              year|manufacturer|               model|  fuel|
odometer|  type|             state|
+-------+------------------+-----------------+--------------------+-----------
-------+------------------+-----------+--------------------+------+-----------
-----+------+------------------+
|  count|            330749|           330749|              330749|
330749|            330749|      320381|              328269|328911|
330749|266535|            326882|
|   mean|7.235564310095632E9|2011.0366561954836|
null|17763.716797329696|2011.0366561954836|        null|  1834.9227294512343|
null|104789.3900057143|  null|  2308.00583090379|
| stddev|  4590897.221947935| 7.974047994873545|                null|
329392.1782005149| 7.974047994873545|        null|   4613.893504914134|
null|3563048.080930306|  null|1178.7649126370138|
|    min|         7218796645|             1900|         SF bay area|
1|              1900|       acura|             """6"""|diesel|               1|
SUV|                  |
|    max|         7241019367|             2021|zanesville / camb…|
123456789|              2021|       volvo| 2018 IMPALA  | other|
2043755555| wagon| rear deck spoiler|
+-------+------------------+-----------------+--------------------+-----------
-------+------------------+-----------+--------------------+------+-----------
-----+------+------------------+
```

[35]:
```
#statistics grouped by fuel type
#create list of features for grouping by fuel type

group_features = ['fuel','year','price','odometer']

(
    imputed
    .select(group_features)
    .groupBy('fuel')
    .agg(*[
          fn.count('*').alias('Count')
        , fn.mean('year').alias('year_avg')
        , fn.mean('price').alias('price_avg')
        , fn.mean('odometer').alias('odometer_avg')
```

```
        , fn.stddev('year').alias('year_stdev')
        , fn.stddev('price').alias('price_stdev')
        , fn.stddev('odometer').alias('odometer_stdev')
    ])
    .orderBy('fuel')
).show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+--------+------+-----------------+-----------------+----------------+-------
-----------+------------------+--------------------+
|    fuel| Count|          year_avg|        price_avg|     odometer_avg|
year_stdev|       price_stdev|      odometer_stdev|
+--------+------+-----------------+-----------------+----------------+-------
-----------+------------------+--------------------+
|    null|  1838|2013.8890097932535|19693.043525571273|81746.14254624592|
6.026766552451515|15112.613584616338|  55135.893649504585|
|  diesel| 21189| 2010.575062532446| 35508.48048515739|234066.9690877342|
7.101750652460408| 848128.1973188096|1.4040413655844554E7|
|electric|  2008|2015.6583665338646|24876.862549800797|36342.88894422311|2.75739
16100901767|16956.002233396455|   28961.87371563167|
|     gas|290590|2010.8803124677381|16232.820565057296| 97287.2702054441|
8.132669033748376|265484.35719200474|   272849.6631905246|
|  hybrid|  4381| 2012.753709198813|15402.460168911208|96751.17347637526|
4.04645547089951|186605.52565019712|    71082.7075684718|
|   other| 10743|2014.1239877129294| 23477.64255794471|72748.79232988923|
6.132807232799724| 14580.41770413505|   57342.64865534258|
+--------+------+-----------------+-----------------+----------------+-------
-----------+------------------+--------------------+
```

[36]:
```
#Run correlation analysis between odometer and price

(
    imputed
    .corr('odometer', 'price')
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
-0.0003709250289899684
```

[37]:
```
#Surprisingly low correlation. As we later found, much of this was likely due␣
↪to highly skewed values.
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[38]:
```
#Run correlation analysis between year and price

(
    imputed
    .corr('year', 'price')
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

0.01007360807970214

[39]:
```
#slightly higher, but ulimtately still low correlation. Further outlier␣
↪analysis would explain this.
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[40]:
```
#Create a correlation table for numeric features

n_features = len(features)

corr = []

for i in range(0, n_features):
    temp = [None] * i

    for j in range(i, n_features):
        temp.append(imputed.corr(features[i], features[j]))
    corr.append([features[i]] + temp)

correlations = spark.createDataFrame(corr, ['Column'] + features)

correlations.show()
```

VBox()

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+--------+----+--------------------+--------------------+
|  Column|year|               price|            odometer|
+--------+----+--------------------+--------------------+
|    year| 1.0|0.010073608079702175|-0.01017961950380…|
|   price|null|                 1.0|-3.70925028989968…|
|odometer|null|                null|                 1.0|
+--------+----+--------------------+--------------------+
```

2.5 Data Visualization

Year Histogram

[41]:
```
#generate bins and count of each bin

histogram_year = (
    imputed
    .select('year')
    .rdd
    .flatMap(lambda record: record)
    .histogram(10)
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[42]:
```
#Generate arrays with list of bin bounds and count of elements in corresponding␣
 ↪bin

for i in histogram_year:
    print(i)

histogram_year
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[1900.0, 1912.1, 1924.2, 1936.3, 1948.4, 1960.5, 1972.6, 1984.7, 1996.8, 2008.9,
2021]
[13, 11, 161, 205, 602, 1864, 1839, 6159, 82245, 237650]
```

```
([1900.0, 1912.1, 1924.2, 1936.3, 1948.4, 1960.5, 1972.6, 1984.7, 1996.8,
2008.9, 2021], [13, 11, 161, 205, 602, 1864, 1839, 6159, 82245, 237650])
```

[43]:
```
#Display bin ranges

for i in range(len(histogram_year[0])-1):
    print('[' + str(round(histogram_year[0][i],2))
        + ',' + str(round(histogram_year[0][i+1],2))
          + ')'
         )
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[1900.0,1912.1)
[1912.1,1924.2)
[1924.2,1936.3)
[1936.3,1948.4)
[1948.4,1960.5)
[1960.5,1972.6)
[1972.6,1984.7)
[1984.7,1996.8)
[1996.8,2008.9)
[2008.9,2021)
```

[44]:
```
#Unpack histogram and pass parameters to Zip

sorted(zip(*histogram_year))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[(1900.0, 13), (1912.1, 11), (1924.2, 161), (1936.3, 205), (1948.4, 602),
(1960.5, 1864), (1972.6, 1839), (1984.7, 6159), (1996.8, 82245), (2008.9,
237650)]
```

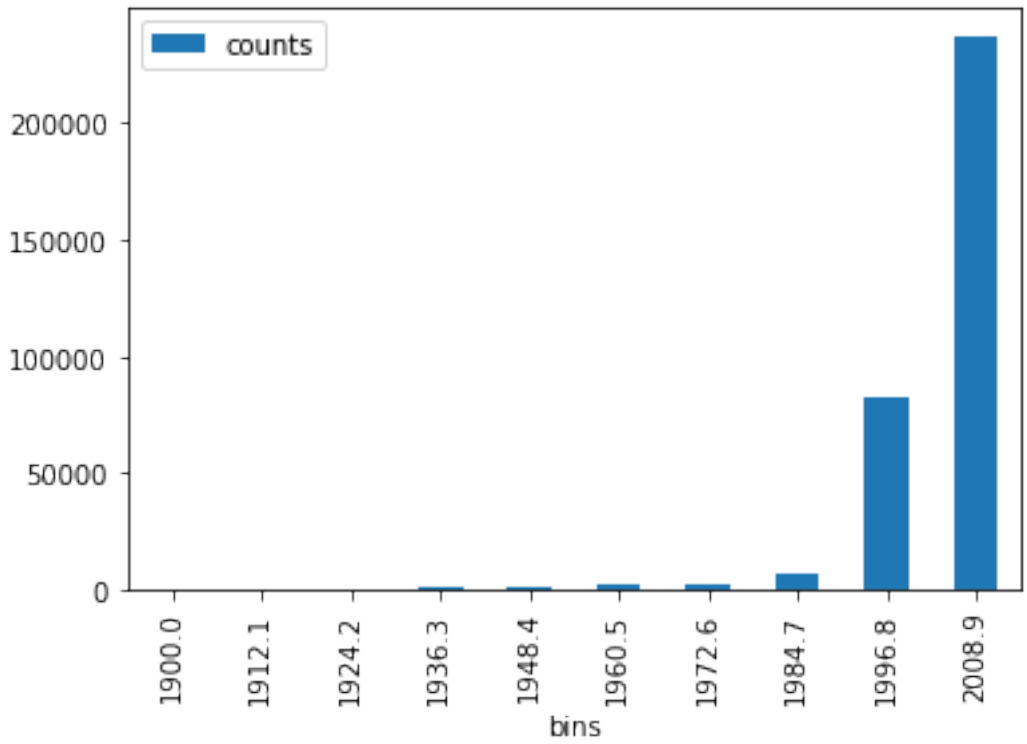
[45]:
```
%%spark -o hist_year
#Export hist_year

hist_year =  spark.createDataFrame(
    list(zip(*histogram_year)),
    ['bins', 'counts'])
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[46]:
```
%matplot plt
import matplotlib
import matplotlib.pyplot as plt
hist_year.set_index('bins'
            ).plot(kind='bar')
plt.show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```



[47]:
```
#Lots of cars in the 2008+ bin
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

Price Histogram

[48]:
```
#generate bins and count of each bin

histogram_price = (
    imputed
    .select('price')
    .rdd
    .flatMap(lambda record: record)
    .histogram(50)
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[49]:
```
#Generate arrays with list of bin bounds and count of elements in corresponding␣
↪bin

for i in histogram_price:
    print(i)

histogram_price
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[1.0, 2469136.76, 4938272.52, 7407408.279999999, 9876544.04, 12345679.799999999,
14814815.559999999, 17283951.32, 19753087.08, 22222222.839999996,
24691358.599999998, 27160494.36, 29629630.119999997, 32098765.879999995,
34567901.64, 37037037.4, 39506173.16, 41975308.919999994, 44444444.67999999,
46913580.44, 49382716.199999996, 51851851.95999999, 54320987.72, 56790123.48,
59259259.239999995, 61728394.99999999, 64197530.75999999, 66666666.519999996,
69135802.28, 71604938.03999999, 74074073.8, 76543209.55999999, 79012345.32,
81481481.08, 83950616.83999999, 86419752.6, 88888888.35999998,
91358024.11999999, 93827159.88, 96296295.63999999, 98765431.39999999,
101234567.16, 103703702.91999999, 106172838.67999999, 108641974.44,
111111110.19999999, 113580245.96, 116049381.71999998, 118518517.47999999,
120987653.24, 123456789]
[330741, 1, 0, 0, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1]
([1.0, 2469136.76, 4938272.52, 7407408.279999999, 9876544.04,
12345679.799999999, 14814815.559999999, 17283951.32, 19753087.08,
22222222.839999996, 24691358.599999998, 27160494.36, 29629630.119999997,
32098765.879999995, 34567901.64, 37037037.4, 39506173.16, 41975308.919999994,
44444444.67999999, 46913580.44, 49382716.199999996, 51851851.95999999,
54320987.72, 56790123.48, 59259259.239999995, 61728394.99999999,
64197530.75999999, 66666666.519999996, 69135802.28, 71604938.03999999,
74074073.8, 76543209.55999999, 79012345.32, 81481481.08, 83950616.83999999,
86419752.6, 88888888.35999998, 91358024.11999999, 93827159.88,
96296295.63999999, 98765431.39999999, 101234567.16, 103703702.91999999,
106172838.67999999, 108641974.44, 111111110.19999999, 113580245.96,
116049381.71999998, 118518517.47999999, 120987653.24, 123456789], [330741, 1, 0,
0, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1])
```

[50]:
```
#Display bin ranges

for i in range(len(histogram_price[0])-1):
    print('[' + str(round(histogram_price[0][i],2))
        + ',' + str(round(histogram_price[0][i+1],2))
```

```
          + ')'
         )
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[1.0,2469136.76)
[2469136.76,4938272.52)
[4938272.52,7407408.28)
[7407408.28,9876544.04)
[9876544.04,12345679.8)
[12345679.8,14814815.56)
[14814815.56,17283951.32)
[17283951.32,19753087.08)
[19753087.08,22222222.84)
[22222222.84,24691358.6)
[24691358.6,27160494.36)
[27160494.36,29629630.12)
[29629630.12,32098765.88)
[32098765.88,34567901.64)
[34567901.64,37037037.4)
[37037037.4,39506173.16)
[39506173.16,41975308.92)
[41975308.92,44444444.68)
[44444444.68,46913580.44)
[46913580.44,49382716.2)
[49382716.2,51851851.96)
[51851851.96,54320987.72)
[54320987.72,56790123.48)
[56790123.48,59259259.24)
[59259259.24,61728395.0)
[61728395.0,64197530.76)
[64197530.76,66666666.52)
[66666666.52,69135802.28)
[69135802.28,71604938.04)
[71604938.04,74074073.8)
[74074073.8,76543209.56)
[76543209.56,79012345.32)
[79012345.32,81481481.08)
[81481481.08,83950616.84)
[83950616.84,86419752.6)
[86419752.6,88888888.36)
[88888888.36,91358024.12)
[91358024.12,93827159.88)
[93827159.88,96296295.64)
[96296295.64,98765431.4)

```
[98765431.4,101234567.16)
[101234567.16,103703702.92)
[103703702.92,106172838.68)
[106172838.68,108641974.44)
[108641974.44,111111110.2)
[111111110.2,113580245.96)
[113580245.96,116049381.72)
[116049381.72,118518517.48)
[118518517.48,120987653.24)
[120987653.24,123456789)
```

[51]:
```
#Unpack histogram and pass parameters to Zip

sorted(zip(*histogram_price))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[(1.0, 330741), (2469136.76, 1), (4938272.52, 0), (7407408.279999999, 0),
(9876544.04, 3), (12345679.799999999, 0), (14814815.559999999, 0), (17283951.32,
0), (19753087.08, 1), (22222222.839999996, 0), (24691358.599999998, 0),
(27160494.36, 0), (29629630.119999997, 0), (32098765.879999995, 0),
(34567901.64, 0), (37037037.4, 0), (39506173.16, 0), (41975308.919999994, 0),
(44444444.67999999, 0), (46913580.44, 0), (49382716.199999996, 0),
(51851851.95999999, 0), (54320987.72, 0), (56790123.48, 0), (59259259.239999995,
0), (61728394.99999999, 0), (64197530.75999999, 0), (66666666.519999996, 0),
(69135802.28, 0), (71604938.03999999, 0), (74074073.8, 0), (76543209.55999999,
0), (79012345.32, 0), (81481481.08, 0), (83950616.83999999, 0), (86419752.6, 0),
(88888888.35999998, 0), (91358024.11999999, 0), (93827159.88, 0),
(96296295.63999999, 1), (98765431.39999999, 1), (101234567.16, 0),
(103703702.91999999, 0), (106172838.67999999, 0), (108641974.44, 0),
(111111110.19999999, 0), (113580245.96, 0), (116049381.71999998, 0),
(118518517.47999999, 0), (120987653.24, 1)]
```

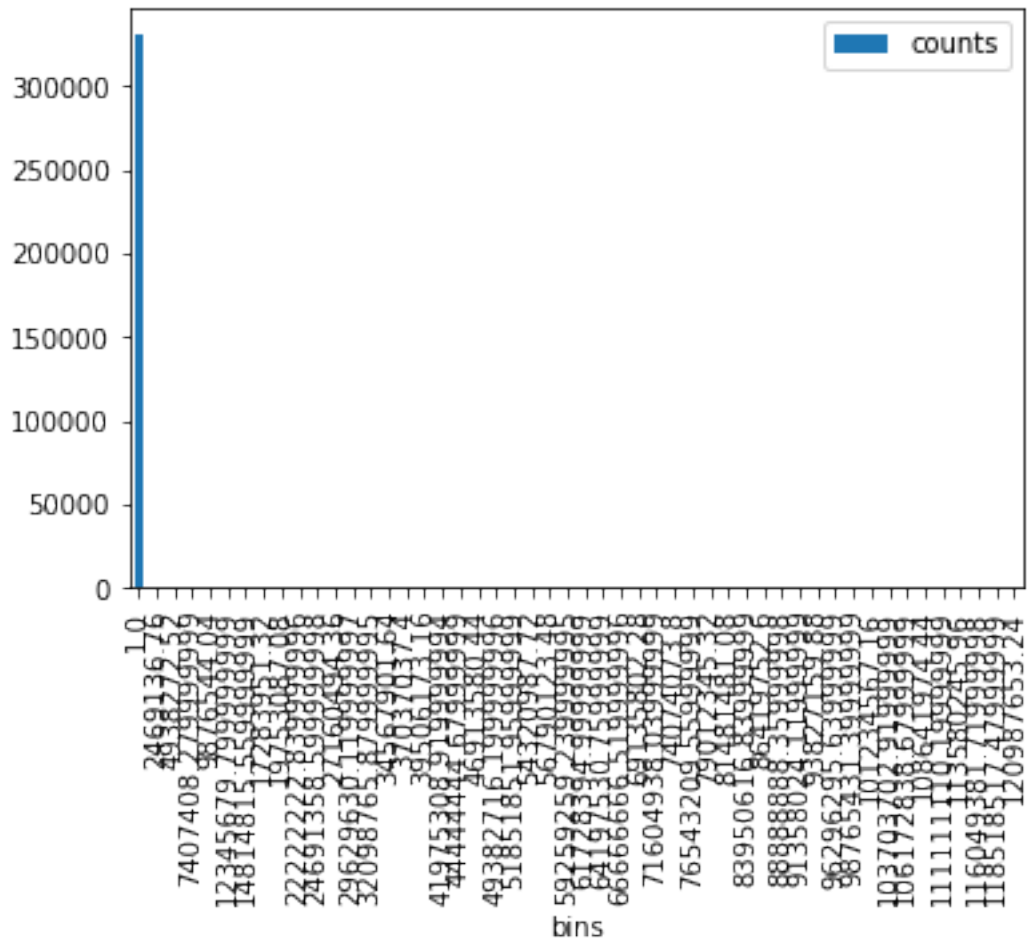
[52]:
```
%%spark -o hist_price
#Export hist_price

hist_price =  spark.createDataFrame(
    list(zip(*histogram_price)),
    ['bins', 'counts'])
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[53]:
```
%matplot plt
import matplotlib
import matplotlib.pyplot as plt
hist_price.set_index('bins'
            ).plot(kind='bar')
plt.show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```



[54]:
```
#Price is significantly skewed
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

Odometer Histogram

[55]:
```
#generate bins and count of each bin

histogram_odometer = (
    imputed
    .select('odometer')
    .rdd
    .flatMap(lambda record: record)
    .histogram(100)
```

```
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[56]:
```
#Generate arrays with list of bin bounds and count of elements in corresponding␣
↪bin

for i in histogram_odometer:
    print(i)

histogram_odometer
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
[1.0, 20437556.54, 40875112.08, 61312667.62, 81750223.16, 102187778.69999999,
122625334.24, 143062889.78, 163500445.32, 183938000.85999998,
204375556.39999998, 224813111.94, 245250667.48, 265688223.01999998,
286125778.56, 306563334.09999996, 327000889.64, 347438445.18,
367876000.71999997, 388313556.26, 408751111.79999995, 429188667.34,
449626222.88, 470063778.41999996, 490501333.96, 510938889.5, 531376445.03999996,
551814000.5799999, 572251556.12, 592689111.66, 613126667.1999999, 633564222.74,
654001778.28, 674439333.8199999, 694876889.36, 715314444.9, 735752000.4399999,
756189555.98, 776627111.52, 797064667.06, 817502222.5999999, 837939778.14,
858377333.68, 878814889.2199999, 899252444.76, 919690000.3, 940127555.8399999,
960565111.38, 981002666.92, 1001440222.4599999, 1021877778.0, 1042315333.54,
1062752889.0799999, 1083190444.62, 1103628000.1599998, 1124065555.7,
1144503111.24, 1164940666.78, 1185378222.32, 1205815777.86, 1226253333.3999999,
1246690888.94, 1267128444.48, 1287566000.02, 1308003555.56, 1328441111.1,
1348878666.6399999, 1369316222.1799998, 1389753777.72, 1410191333.26,
1430628888.8, 1451066444.34, 1471503999.8799999, 1491941555.4199998,
1512379110.96, 1532816666.5, 1553254222.04, 1573691777.58, 1594129333.12,
1614566888.6599998, 1635004444.1999998, 1655441999.74, 1675879555.28,
1696317110.82, 1716754666.36, 1737192221.8999999, 1757629777.4399998,
1778067332.98, 1798504888.52, 1818942444.06, 1839379999.6, 1859817555.1399999,
1880255110.6799998, 1900692666.22, 1921130221.76, 1941567777.3, 1962005332.84,
1982442888.3799999, 2002880443.9199998, 2023317999.4599998, 2043755555]
[330747, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
([1.0, 20437556.54, 40875112.08, 61312667.62, 81750223.16, 102187778.69999999,
122625334.24, 143062889.78, 163500445.32, 183938000.85999998,
```

```
204375556.39999998, 224813111.94, 245250667.48, 265688223.01999998,
286125778.56, 306563334.09999996, 327000889.64, 347438445.18,
367876000.71999997, 388313556.26, 408751111.79999995, 429188667.34,
449626222.88, 470063778.41999996, 490501333.96, 510938889.5, 531376445.03999996,
551814000.5799999, 572251556.12, 592689111.66, 613126667.1999999, 633564222.74,
654001778.28, 674439333.8199999, 694876889.36, 715314444.9, 735752000.4399999,
756189555.98, 776627111.52, 797064667.06, 817502222.5999999, 837939778.14,
858377333.68, 878814889.2199999, 899252444.76, 919690000.3, 940127555.8399999,
960565111.38, 981002666.92, 1001440222.4599999, 1021877778.0, 1042315333.54,
1062752889.0799999, 1083190444.62, 1103628000.1599998, 1124065555.7,
1144503111.24, 1164940666.78, 1185378222.32, 1205815777.86, 1226253333.3999999,
1246690888.94, 1267128444.48, 1287566000.02, 1308003555.56, 1328441111.1,
1348878666.6399999, 1369316222.1799998, 1389753777.72, 1410191333.26,
1430628888.8, 1451066444.34, 1471503999.8799999, 1491941555.4199998,
1512379110.96, 1532816666.5, 1553254222.04, 1573691777.58, 1594129333.12,
1614566888.6599998, 1635004444.1999998, 1655441999.74, 1675879555.28,
1696317110.82, 1716754666.36, 1737192221.8999999, 1757629777.4399998,
1778067332.98, 1798504888.52, 1818942444.06, 1839379999.6, 1859817555.1399999,
1880255110.6799998, 1900692666.22, 1921130221.76, 1941567777.3, 1962005332.84,
1982442888.3799999, 2002880443.9199998, 2023317999.4599998, 2043755555],
[330747, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1])
```

[57]:
```
#Display bin ranges

for i in range(len(histogram_odometer[0])-1):
    print('[' + str(round(histogram_odometer[0][i],2))
        + ',' + str(round(histogram_odometer[0][i+1],2))
          + ')'
         )
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


[1.0,20437556.54)
[20437556.54,40875112.08)
[40875112.08,61312667.62)
[61312667.62,81750223.16)
[81750223.16,102187778.7)
[102187778.7,122625334.24)
[122625334.24,143062889.78)
[143062889.78,163500445.32)
[163500445.32,183938000.86)
[183938000.86,204375556.4)
```

[204375556.4,224813111.94)
[224813111.94,245250667.48)
[245250667.48,265688223.02)
[265688223.02,286125778.56)
[286125778.56,306563334.1)
[306563334.1,327000889.64)
[327000889.64,347438445.18)
[347438445.18,367876000.72)
[367876000.72,388313556.26)
[388313556.26,408751111.8)
[408751111.8,429188667.34)
[429188667.34,449626222.88)
[449626222.88,470063778.42)
[470063778.42,490501333.96)
[490501333.96,510938889.5)
[510938889.5,531376445.04)
[531376445.04,551814000.58)
[551814000.58,572251556.12)
[572251556.12,592689111.66)
[592689111.66,613126667.2)
[613126667.2,633564222.74)
[633564222.74,654001778.28)
[654001778.28,674439333.82)
[674439333.82,694876889.36)
[694876889.36,715314444.9)
[715314444.9,735752000.44)
[735752000.44,756189555.98)
[756189555.98,776627111.52)
[776627111.52,797064667.06)
[797064667.06,817502222.6)
[817502222.6,837939778.14)
[837939778.14,858377333.68)
[858377333.68,878814889.22)
[878814889.22,899252444.76)
[899252444.76,919690000.3)
[919690000.3,940127555.84)
[940127555.84,960565111.38)
[960565111.38,981002666.92)
[981002666.92,1001440222.46)
[1001440222.46,1021877778.0)
[1021877778.0,1042315333.54)
[1042315333.54,1062752889.08)
[1062752889.08,1083190444.62)
[1083190444.62,1103628000.16)
[1103628000.16,1124065555.7)
[1124065555.7,1144503111.24)
[1144503111.24,1164940666.78)
[1164940666.78,1185378222.32)

```
[1185378222.32,1205815777.86)
[1205815777.86,1226253333.4)
[1226253333.4,1246690888.94)
[1246690888.94,1267128444.48)
[1267128444.48,1287566000.02)
[1287566000.02,1308003555.56)
[1308003555.56,1328441111.1)
[1328441111.1,1348878666.64)
[1348878666.64,1369316222.18)
[1369316222.18,1389753777.72)
[1389753777.72,1410191333.26)
[1410191333.26,1430628888.8)
[1430628888.8,1451066444.34)
[1451066444.34,1471503999.88)
[1471503999.88,1491941555.42)
[1491941555.42,1512379110.96)
[1512379110.96,1532816666.5)
[1532816666.5,1553254222.04)
[1553254222.04,1573691777.58)
[1573691777.58,1594129333.12)
[1594129333.12,1614566888.66)
[1614566888.66,1635004444.2)
[1635004444.2,1655441999.74)
[1655441999.74,1675879555.28)
[1675879555.28,1696317110.82)
[1696317110.82,1716754666.36)
[1716754666.36,1737192221.9)
[1737192221.9,1757629777.44)
[1757629777.44,1778067332.98)
[1778067332.98,1798504888.52)
[1798504888.52,1818942444.06)
[1818942444.06,1839379999.6)
[1839379999.6,1859817555.14)
[1859817555.14,1880255110.68)
[1880255110.68,1900692666.22)
[1900692666.22,1921130221.76)
[1921130221.76,1941567777.3)
[1941567777.3,1962005332.84)
[1962005332.84,1982442888.38)
[1982442888.38,2002880443.92)
[2002880443.92,2023317999.46)
[2023317999.46,2043755555)
```

[58]:
```
#Unpack histogram and pass parameters to Zip

sorted(zip(*histogram_odometer))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[(1.0, 330747), (20437556.54, 0), (40875112.08, 0), (61312667.62, 0),
(81750223.16, 0), (102187778.69999999, 0), (122625334.24, 1), (143062889.78, 0),
(163500445.32, 0), (183938000.85999998, 0), (204375556.39999998, 0),
(224813111.94, 0), (245250667.48, 0), (265688223.01999998, 0), (286125778.56,
0), (306563334.09999996, 0), (327000889.64, 0), (347438445.18, 0),
(367876000.71999997, 0), (388313556.26, 0), (408751111.79999995, 0),
(429188667.34, 0), (449626222.88, 0), (470063778.41999996, 0), (490501333.96,
0), (510938889.5, 0), (531376445.03999996, 0), (551814000.5799999, 0),
(572251556.12, 0), (592689111.66, 0), (613126667.1999999, 0), (633564222.74, 0),
(654001778.28, 0), (674439333.8199999, 0), (694876889.36, 0), (715314444.9, 0),
(735752000.4399999, 0), (756189555.98, 0), (776627111.52, 0), (797064667.06, 0),
(817502222.5999999, 0), (837939778.14, 0), (858377333.68, 0),
(878814889.2199999, 0), (899252444.76, 0), (919690000.3, 0), (940127555.8399999,
0), (960565111.38, 0), (981002666.92, 0), (1001440222.4599999, 0),
(1021877778.0, 0), (1042315333.54, 0), (1062752889.0799999, 0), (1083190444.62,
0), (1103628000.1599998, 0), (1124065555.7, 0), (1144503111.24, 0),
(1164940666.78, 0), (1185378222.32, 0), (1205815777.86, 0), (1226253333.3999999,
0), (1246690888.94, 0), (1267128444.48, 0), (1287566000.02, 0), (1308003555.56,
0), (1328441111.1, 0), (1348878666.6399999, 0), (1369316222.1799998, 0),
(1389753777.72, 0), (1410191333.26, 0), (1430628888.8, 0), (1451066444.34, 0),
(1471503999.8799999, 0), (1491941555.4199998, 0), (1512379110.96, 0),
(1532816666.5, 0), (1553254222.04, 0), (1573691777.58, 0), (1594129333.12, 0),
(1614566888.6599998, 0), (1635004444.1999998, 0), (1655441999.74, 0),
(1675879555.28, 0), (1696317110.82, 0), (1716754666.36, 0), (1737192221.8999999,
0), (1757629777.4399998, 0), (1778067332.98, 0), (1798504888.52, 0),
(1818942444.06, 0), (1839379999.6, 0), (1859817555.1399999, 0),
(1880255110.6799998, 0), (1900692666.22, 0), (1921130221.76, 0), (1941567777.3,
0), (1962005332.84, 0), (1982442888.3799999, 0), (2002880443.9199998, 0),
(2023317999.4599998, 1)]
```

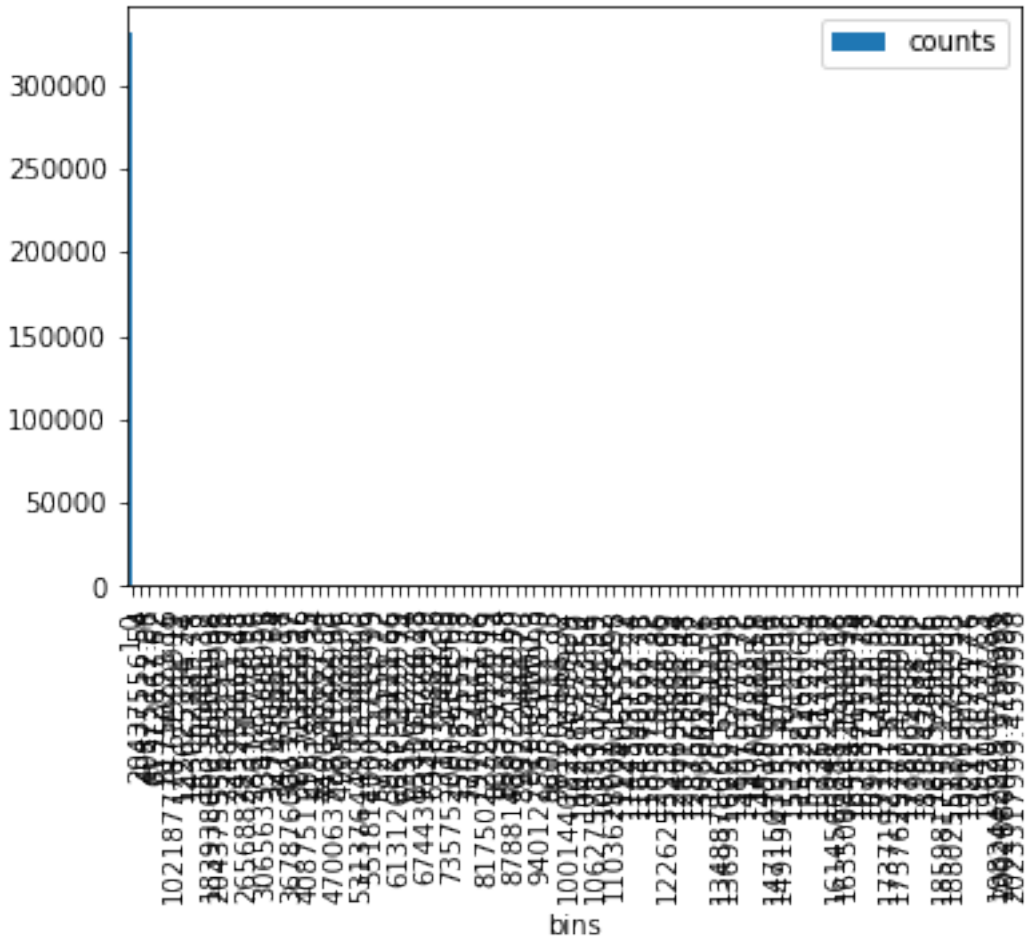
[59]:
```
%%spark -o hist_odometer
#Export hist_odometer

hist_odometer =  spark.createDataFrame(
    list(zip(*histogram_odometer)),
    ['bins', 'counts'])
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[60]:
```
%matplot plt
import matplotlib
import matplotlib.pyplot as plt
hist_odometer.set_index('bins'
            ).plot(kind='bar')
plt.show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```



2.6 Outlier Analysis

[61]:
```
#In order to analyze outliers properly, we will look at STD for each numeric␣
 ↪feature

#describe price

descriptive_stats = imputed.describe('price')
descriptive_stats.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+-------+------------------+
|summary|             price|
```

```
+-------+------------------+
|  count|            330749|
|   mean|17763.716797329696|
| stddev|329392.17820051487|
|    min|                 1|
|    max|         123456789|
+-------+------------------+
```

[62]:
```
#Our dataset is incredibly diverse, with a standard deviation which is␣
 ↪significantly larger than our mean
#The edge values are also widely drastic, creating further polarization
#A min value of 1 is highly skewed, so we looked at the number of values with 1␣
 ↪as the price

price_outlier=imputed.filter((fn.col('price') == 1))

price_outlier.count()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

764

[63]:
```
#After further analysis, we found that there were many extremly low values,␣
 ↪especially in the under 100 category.
#We spent time analyzing a potential threshold to remove values to achieve a␣
 ↪more balanced dataset
#Trying to remain within only a 3% removal of the dataset, we aimed to take the␣
 ↪lowest 1.5% and highest 1.5% of values
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[64]:
```
#We attempted to find the cutoff for the lowest 1.5% of values

lowest_removed = imputed.filter((fn.col('price') > 350))

lowest_removed.count()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

325921

[65]:
```
#325921/330749 =.985, or about 2% of our data
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[66]:
```
#We then did the same for the high prices

highest_removed = imputed.filter((fn.col('price') < 55000))

highest_removed.count()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

325954

[67]:
```
#325954/330749 = 98.6, also around 2%
#Using these threshholds, we removed numbers outside of our new range and␣
 ↪graphed the result
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[68]:
```
price_outliers_removed = imputed.filter((fn.col('price') > 350) & (fn.
 ↪col('price') < 55000))

descriptive_stats = price_outliers_removed.describe('price')
descriptive_stats.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+-------+------------------+
|summary|             price|
+-------+------------------+
|  count|            321126|
|   mean|15964.723401406301|
| stddev| 11673.83417745105|
|    min|               351|
```

```
|    max|             54999|
+-------+-------------------+
```

[69]:
```
#generate bins and count of each bin

histogram_price_outliers = (
    price_outliers_removed
    .select('price')
    .rdd
    .flatMap(lambda record: record)
    .histogram(50)
)
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[70]:
```
#Generate arrays with list of bin bounds and count of elements in corresponding␣
 ↪bin

for i in histogram_price_outliers:
    print(i)

histogram_price_outliers
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


[351.0, 1443.96, 2536.92, 3629.88, 4722.84, 5815.8, 6908.76, 8001.72, 9094.68,
10187.64, 11280.6, 12373.560000000001, 13466.52, 14559.48, 15652.44, 16745.4,
17838.36, 18931.32, 20024.28, 21117.24, 22210.2, 23303.16, 24396.120000000003,
25489.08, 26582.04, 27675.0, 28767.96, 29860.920000000002, 30953.88, 32046.84,
33139.8, 34232.76, 35325.72, 36418.68, 37511.64, 38604.6, 39697.56,
40790.520000000004, 41883.48, 42976.44, 44069.4, 45162.36, 46255.32, 47348.28,
48441.240000000005, 49534.200000000004, 50627.16, 51720.12, 52813.08, 53906.04,
54999]
[7108, 11581, 14839, 16899, 17201, 17520, 21250, 12992, 12729, 10886, 10094,
10683, 11499, 11163, 9512, 8779, 9037, 11522, 5185, 5406, 5418, 5418, 5679,
5072, 5487, 5730, 4642, 5192, 6539, 3335, 3155, 3856, 2956, 2823, 2666, 2429,
3115, 1359, 1715, 1968, 1015, 683, 807, 598, 671, 973, 218, 480, 463, 779]
([351.0, 1443.96, 2536.92, 3629.88, 4722.84, 5815.8, 6908.76, 8001.72, 9094.68,
10187.64, 11280.6, 12373.560000000001, 13466.52, 14559.48, 15652.44, 16745.4,
17838.36, 18931.32, 20024.28, 21117.24, 22210.2, 23303.16, 24396.120000000003,
25489.08, 26582.04, 27675.0, 28767.96, 29860.920000000002, 30953.88, 32046.84,
33139.8, 34232.76, 35325.72, 36418.68, 37511.64, 38604.6, 39697.56,
```

```
40790.520000000004, 41883.48, 42976.44, 44069.4, 45162.36, 46255.32, 47348.28,
48441.240000000005, 49534.200000000004, 50627.16, 51720.12, 52813.08, 53906.04,
54999], [7108, 11581, 14839, 16899, 17201, 17520, 21250, 12992, 12729, 10886,
10094, 10683, 11499, 11163, 9512, 8779, 9037, 11522, 5185, 5406, 5418, 5418,
5679, 5072, 5487, 5730, 4642, 5192, 6539, 3335, 3155, 3856, 2956, 2823, 2666,
2429, 3115, 1359, 1715, 1968, 1015, 683, 807, 598, 671, 973, 218, 480, 463,
779])
```

[71]:
```
#Display bin ranges
for i in range(len(histogram_price_outliers[0])-1):
    print('[' + str(round(histogram_price_outliers[0][i],2))
        + ',' + str(round(histogram_price_outliers[0][i+1],2))
          + ')'
         )
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[351.0,1443.96)
[1443.96,2536.92)
[2536.92,3629.88)
[3629.88,4722.84)
[4722.84,5815.8)
[5815.8,6908.76)
[6908.76,8001.72)
[8001.72,9094.68)
[9094.68,10187.64)
[10187.64,11280.6)
[11280.6,12373.56)
[12373.56,13466.52)
[13466.52,14559.48)
[14559.48,15652.44)
[15652.44,16745.4)
[16745.4,17838.36)
[17838.36,18931.32)
[18931.32,20024.28)
[20024.28,21117.24)
[21117.24,22210.2)
[22210.2,23303.16)
[23303.16,24396.12)
[24396.12,25489.08)
[25489.08,26582.04)
[26582.04,27675.0)
[27675.0,28767.96)
[28767.96,29860.92)
[29860.92,30953.88)
```

```
[30953.88,32046.84)
[32046.84,33139.8)
[33139.8,34232.76)
[34232.76,35325.72)
[35325.72,36418.68)
[36418.68,37511.64)
[37511.64,38604.6)
[38604.6,39697.56)
[39697.56,40790.52)
[40790.52,41883.48)
[41883.48,42976.44)
[42976.44,44069.4)
[44069.4,45162.36)
[45162.36,46255.32)
[46255.32,47348.28)
[47348.28,48441.24)
[48441.24,49534.2)
[49534.2,50627.16)
[50627.16,51720.12)
[51720.12,52813.08)
[52813.08,53906.04)
[53906.04,54999)
```

[72]:
```
#Unpack histogram and pass parameters to Zip

sorted(zip(*histogram_price_outliers))
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


[(351.0, 7108), (1443.96, 11581), (2536.92, 14839), (3629.88, 16899), (4722.84,
17201), (5815.8, 17520), (6908.76, 21250), (8001.72, 12992), (9094.68, 12729),
(10187.64, 10886), (11280.6, 10094), (12373.560000000001, 10683), (13466.52,
11499), (14559.48, 11163), (15652.44, 9512), (16745.4, 8779), (17838.36, 9037),
(18931.32, 11522), (20024.28, 5185), (21117.24, 5406), (22210.2, 5418),
(23303.16, 5418), (24396.120000000003, 5679), (25489.08, 5072), (26582.04,
5487), (27675.0, 5730), (28767.96, 4642), (29860.920000000002, 5192), (30953.88,
6539), (32046.84, 3335), (33139.8, 3155), (34232.76, 3856), (35325.72, 2956),
(36418.68, 2823), (37511.64, 2666), (38604.6, 2429), (39697.56, 3115),
(40790.520000000004, 1359), (41883.48, 1715), (42976.44, 1968), (44069.4, 1015),
(45162.36, 683), (46255.32, 807), (47348.28, 598), (48441.240000000005, 671),
(49534.200000000004, 973), (50627.16, 218), (51720.12, 480), (52813.08, 463),
(53906.04, 779)]
```

[73]:
```
%%spark -o hist_pOutliers
#Export hist_price_outliers
```

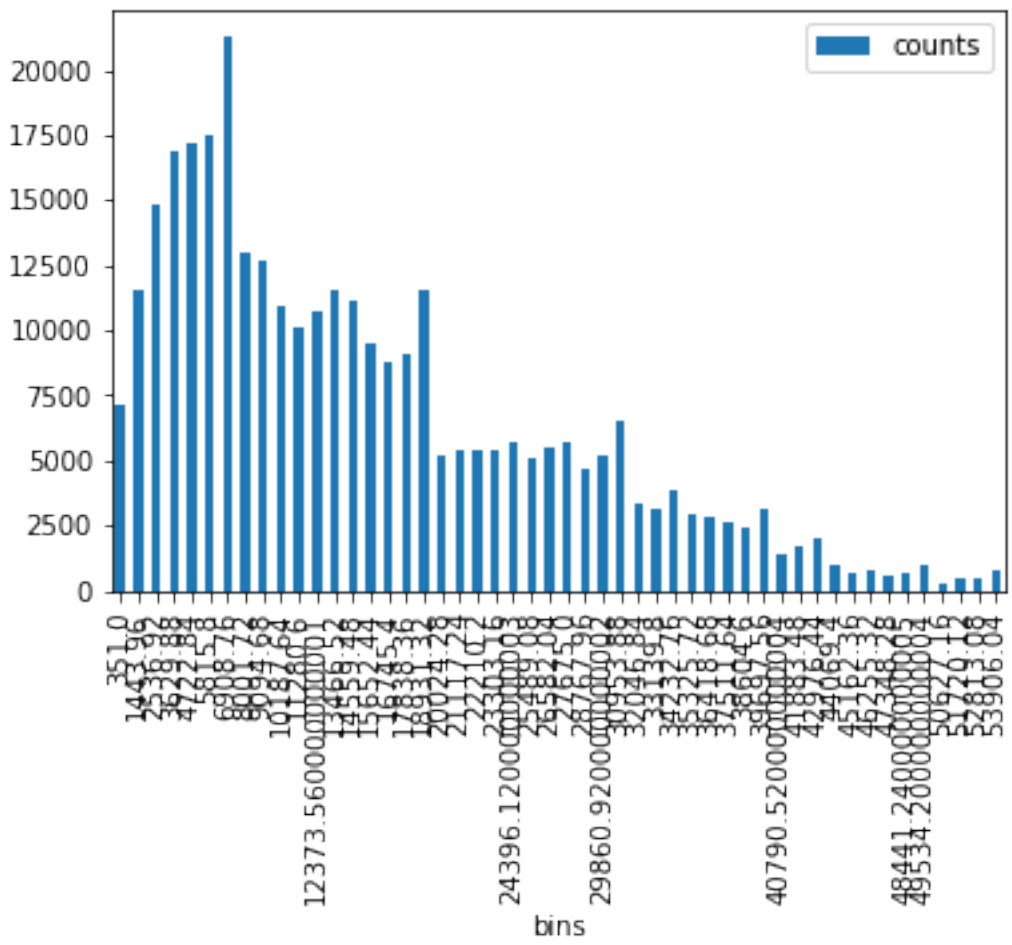
```
hist_pOutliers = spark.createDataFrame(
    list(zip(*histogram_price_outliers)),
    ['bins', 'counts'])
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[74]:
```
%matplot plt
import matplotlib
import matplotlib.pyplot as plt
hist_pOutliers.set_index('bins'
            ).plot(kind='bar')
plt.show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```



[75]:
```
#Our results are still skewed to the right, but this represents a much more␣
 ↪reasonable graph.
#What this shows us is that many people are either undervaluing or simply␣
 ↪valuing their used cars cheaply.
#The extreme outliers at the higher end seemed to cause more disruption, which␣
 ↪makes sense considering the used-car market
#As buying a used car is risky, it is likely individuals are willing to price␣
 ↪higher, either as an honest price valuation
#Or because they are looking to encourage sales
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

Odometer Outliers

[76]:
```
#We noticed similar issues with our odometer readings. Here, the same notion␣
↪was applied, remove the top and bottom 1.5%
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[77]:
```
#In order to analyze outliers properly, we will look at STD for each numeric␣
↪feature

#describe odometer

descriptive_stats = imputed.describe('odometer')
descriptive_stats.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+-------+-----------------+
|summary|         odometer|
+-------+-----------------+
|  count|           330749|
|   mean|104789.3900057143|
| stddev|3563048.080930306|
|    min|                1|
|    max|       2043755555|
+-------+-----------------+
```

[78]:
```
#Once again, we had a similar scenario. Making the same assumption, we looked␣
↪for the high and low cutoff points
#Because the count of the recods is the same, the ~325,800 cutoff was used for␣
↪both high and low values
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[79]:
```
#We attempted to find the cutoff for the lowest 1.5% of values

lowest_odometer = imputed.filter((fn.col('odometer') > 350))

lowest_odometer.count()
```

VBox()

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
325619
```

[80]:
```
#We attempted to find the cutoff for the highest 1.5% of values

highest_odometer = imputed.filter((fn.col('odometer') < 255000))

highest_odometer.count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
325763
```

[81]:
```
#We create a new dataframe removing the highest and lowest 1.5% and graph

odometer_outliers_removed = imputed.filter((fn.col('odometer') > 350) & (fn.
↪col('odometer') < 255000))

descriptive_stats = odometer_outliers_removed.describe('odometer')
descriptive_stats.show()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+-------+-----------------+
|summary|         odometer|
+-------+-----------------+
|  count|           320633|
|   mean|93204.22629610801|
| stddev|58076.42568042224|
|    min|              353|
|    max|           254969|
+-------+-----------------+
```

[82]:
```
#generate bins and count of each bin

histogram_odometerOutliers = (
    odometer_outliers_removed
    .select('odometer')
    .rdd
```

```
    .flatMap(lambda record: record)
    .histogram(100)
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[83]:
```
#Generate arrays with list of bin bounds and count of elements in corresponding␣
↪bin

for i in histogram_odometerOutliers:
    print(i)

histogram_odometerOutliers
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
[353.0, 2899.16, 5445.32, 7991.48, 10537.64, 13083.8, 15629.96, 18176.12,
20722.28, 23268.44, 25814.6, 28360.76, 30906.92, 33453.08, 35999.24,
38545.399999999994, 41091.56, 43637.72, 46183.88, 48730.03999999999, 51276.2,
53822.36, 56368.52, 58914.67999999999, 61460.84, 64007.0, 66553.16,
69099.31999999999, 71645.48, 74191.64, 76737.79999999999, 79283.95999999999,
81830.12, 84376.28, 86922.44, 89468.59999999999, 92014.76, 94560.92,
97107.07999999999, 99653.23999999999, 102199.4, 104745.56, 107291.72,
109837.87999999999, 112384.04, 114930.2, 117476.35999999999, 120022.51999999999,
122568.68, 125114.84, 127661.0, 130207.15999999999, 132753.32,
135299.47999999998, 137845.63999999998, 140391.8, 142937.96, 145484.12,
148030.28, 150576.44, 153122.59999999998, 155668.75999999998,
158214.91999999998, 160761.08, 163307.24, 165853.4, 168399.56, 170945.72,
173491.88, 176038.03999999998, 178584.19999999998, 181130.36, 183676.52,
186222.68, 188768.84, 191315.0, 193861.15999999997, 196407.31999999998,
198953.47999999998, 201499.63999999998, 204045.8, 206591.96, 209138.12,
211684.28, 214230.44, 216776.59999999998, 219322.75999999998,
221868.91999999998, 224415.08, 226961.24, 229507.4, 232053.56,
234599.71999999997, 237145.87999999998, 239692.03999999998, 242238.19999999998,
244784.36, 247330.52, 249876.68, 252422.84, 254969]
[2980, 3220, 3504, 4096, 4705, 4536, 5235, 5473, 4805, 5108, 5339, 5579, 5805,
5649, 5362, 5211, 5050, 4599, 4378, 4828, 4195, 4270, 3905, 4446, 4435, 4030,
4280, 4310, 4484, 4401, 4428, 4613, 4763, 4257, 4503, 5090, 4446, 5125, 5098,
5653, 4485, 4638, 3958, 4835, 3872, 4484, 4782, 3743, 4396, 3625, 4635, 3499,
3923, 3386, 4419, 3001, 3784, 3577, 3819, 3090, 2640, 2974, 3074, 2827, 2457,
2624, 2703, 2118, 2385, 1968, 2638, 1545, 1908, 1438, 1955, 1229, 1394, 1189,
2237, 1135, 1007, 990, 929, 978, 717, 710, 813, 653, 615, 601, 689, 453, 484,
```

```
283, 495, 312, 379, 239, 451, 252]
([353.0, 2899.16, 5445.32, 7991.48, 10537.64, 13083.8, 15629.96, 18176.12,
20722.28, 23268.44, 25814.6, 28360.76, 30906.92, 33453.08, 35999.24,
38545.399999999994, 41091.56, 43637.72, 46183.88, 48730.03999999999, 51276.2,
53822.36, 56368.52, 58914.67999999999, 61460.84, 64007.0, 66553.16,
69099.31999999999, 71645.48, 74191.64, 76737.79999999999, 79283.95999999999,
81830.12, 84376.28, 86922.44, 89468.59999999999, 92014.76, 94560.92,
97107.07999999999, 99653.23999999999, 102199.4, 104745.56, 107291.72,
109837.87999999999, 112384.04, 114930.2, 117476.35999999999, 120022.51999999999,
122568.68, 125114.84, 127661.0, 130207.15999999999, 132753.32,
135299.47999999998, 137845.63999999998, 140391.8, 142937.96, 145484.12,
148030.28, 150576.44, 153122.59999999998, 155668.75999999998,
158214.91999999998, 160761.08, 163307.24, 165853.4, 168399.56, 170945.72,
173491.88, 176038.03999999998, 178584.19999999998, 181130.36, 183676.52,
186222.68, 188768.84, 191315.0, 193861.15999999997, 196407.31999999998,
198953.47999999998, 201499.63999999998, 204045.8, 206591.96, 209138.12,
211684.28, 214230.44, 216776.59999999998, 219322.75999999998,
221868.91999999998, 224415.08, 226961.24, 229507.4, 232053.56,
234599.71999999997, 237145.87999999998, 239692.03999999998, 242238.19999999998,
244784.36, 247330.52, 249876.68, 252422.84, 254969], [2980, 3220, 3504, 4096,
4705, 4536, 5235, 5473, 4805, 5108, 5339, 5579, 5805, 5649, 5362, 5211, 5050,
4599, 4378, 4828, 4195, 4270, 3905, 4446, 4435, 4030, 4280, 4310, 4484, 4401,
4428, 4613, 4763, 4257, 4503, 5090, 4446, 5125, 5098, 5653, 4485, 4638, 3958,
4835, 3872, 4484, 4782, 3743, 4396, 3625, 4635, 3499, 3923, 3386, 4419, 3001,
3784, 3577, 3819, 3090, 2640, 2974, 3074, 2827, 2457, 2624, 2703, 2118, 2385,
1968, 2638, 1545, 1908, 1438, 1955, 1229, 1394, 1189, 2237, 1135, 1007, 990,
929, 978, 717, 710, 813, 653, 615, 601, 689, 453, 484, 283, 495, 312, 379, 239,
451, 252])
```

[84]:
```
#Display bin ranges

for i in range(len(histogram_odometerOutliers[0])-1):
    print('[' + str(round(histogram_odometerOutliers[0][i],2))
        + ',' + str(round(histogram_odometerOutliers[0][i+1],2))
          + ')'
         )
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


[353.0,2899.16)
[2899.16,5445.32)
[5445.32,7991.48)
[7991.48,10537.64)
[10537.64,13083.8)
[13083.8,15629.96)
```

[15629.96,18176.12)
[18176.12,20722.28)
[20722.28,23268.44)
[23268.44,25814.6)
[25814.6,28360.76)
[28360.76,30906.92)
[30906.92,33453.08)
[33453.08,35999.24)
[35999.24,38545.4)
[38545.4,41091.56)
[41091.56,43637.72)
[43637.72,46183.88)
[46183.88,48730.04)
[48730.04,51276.2)
[51276.2,53822.36)
[53822.36,56368.52)
[56368.52,58914.68)
[58914.68,61460.84)
[61460.84,64007.0)
[64007.0,66553.16)
[66553.16,69099.32)
[69099.32,71645.48)
[71645.48,74191.64)
[74191.64,76737.8)
[76737.8,79283.96)
[79283.96,81830.12)
[81830.12,84376.28)
[84376.28,86922.44)
[86922.44,89468.6)
[89468.6,92014.76)
[92014.76,94560.92)
[94560.92,97107.08)
[97107.08,99653.24)
[99653.24,102199.4)
[102199.4,104745.56)
[104745.56,107291.72)
[107291.72,109837.88)
[109837.88,112384.04)
[112384.04,114930.2)
[114930.2,117476.36)
[117476.36,120022.52)
[120022.52,122568.68)
[122568.68,125114.84)
[125114.84,127661.0)
[127661.0,130207.16)
[130207.16,132753.32)
[132753.32,135299.48)
[135299.48,137845.64)

[137845.64,140391.8)
[140391.8,142937.96)
[142937.96,145484.12)
[145484.12,148030.28)
[148030.28,150576.44)
[150576.44,153122.6)
[153122.6,155668.76)
[155668.76,158214.92)
[158214.92,160761.08)
[160761.08,163307.24)
[163307.24,165853.4)
[165853.4,168399.56)
[168399.56,170945.72)
[170945.72,173491.88)
[173491.88,176038.04)
[176038.04,178584.2)
[178584.2,181130.36)
[181130.36,183676.52)
[183676.52,186222.68)
[186222.68,188768.84)
[188768.84,191315.0)
[191315.0,193861.16)
[193861.16,196407.32)
[196407.32,198953.48)
[198953.48,201499.64)
[201499.64,204045.8)
[204045.8,206591.96)
[206591.96,209138.12)
[209138.12,211684.28)
[211684.28,214230.44)
[214230.44,216776.6)
[216776.6,219322.76)
[219322.76,221868.92)
[221868.92,224415.08)
[224415.08,226961.24)
[226961.24,229507.4)
[229507.4,232053.56)
[232053.56,234599.72)
[234599.72,237145.88)
[237145.88,239692.04)
[239692.04,242238.2)
[242238.2,244784.36)
[244784.36,247330.52)
[247330.52,249876.68)
[249876.68,252422.84)
[252422.84,254969)

[85]:
```
#Unpack histogram and pass parameters to Zip

sorted(zip(*histogram_odometerOutliers))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
[(353.0, 2980), (2899.16, 3220), (5445.32, 3504), (7991.48, 4096), (10537.64,
4705), (13083.8, 4536), (15629.96, 5235), (18176.12, 5473), (20722.28, 4805),
(23268.44, 5108), (25814.6, 5339), (28360.76, 5579), (30906.92, 5805),
(33453.08, 5649), (35999.24, 5362), (38545.399999999994, 5211), (41091.56,
5050), (43637.72, 4599), (46183.88, 4378), (48730.03999999999, 4828), (51276.2,
4195), (53822.36, 4270), (56368.52, 3905), (58914.67999999999, 4446), (61460.84,
4435), (64007.0, 4030), (66553.16, 4280), (69099.31999999999, 4310), (71645.48,
4484), (74191.64, 4401), (76737.79999999999, 4428), (79283.95999999999, 4613),
(81830.12, 4763), (84376.28, 4257), (86922.44, 4503), (89468.59999999999, 5090),
(92014.76, 4446), (94560.92, 5125), (97107.07999999999, 5098),
(99653.23999999999, 5653), (102199.4, 4485), (104745.56, 4638), (107291.72,
3958), (109837.87999999999, 4835), (112384.04, 3872), (114930.2, 4484),
(117476.35999999999, 4782), (120022.51999999999, 3743), (122568.68, 4396),
(125114.84, 3625), (127661.0, 4635), (130207.15999999999, 3499), (132753.32,
3923), (135299.47999999998, 3386), (137845.63999999998, 4419), (140391.8, 3001),
(142937.96, 3784), (145484.12, 3577), (148030.28, 3819), (150576.44, 3090),
(153122.59999999998, 2640), (155668.75999999998, 2974), (158214.91999999998,
3074), (160761.08, 2827), (163307.24, 2457), (165853.4, 2624), (168399.56,
2703), (170945.72, 2118), (173491.88, 2385), (176038.03999999998, 1968),
(178584.19999999998, 2638), (181130.36, 1545), (183676.52, 1908), (186222.68,
1438), (188768.84, 1955), (191315.0, 1229), (193861.15999999997, 1394),
(196407.31999999998, 1189), (198953.47999999998, 2237), (201499.63999999998,
1135), (204045.8, 1007), (206591.96, 990), (209138.12, 929), (211684.28, 978),
(214230.44, 717), (216776.59999999998, 710), (219322.75999999998, 813),
(221868.91999999998, 653), (224415.08, 615), (226961.24, 601), (229507.4, 689),
(232053.56, 453), (234599.71999999997, 484), (237145.87999999998, 283),
(239692.03999999998, 495), (242238.19999999998, 312), (244784.36, 379),
(247330.52, 239), (249876.68, 451), (252422.84, 252)]
```

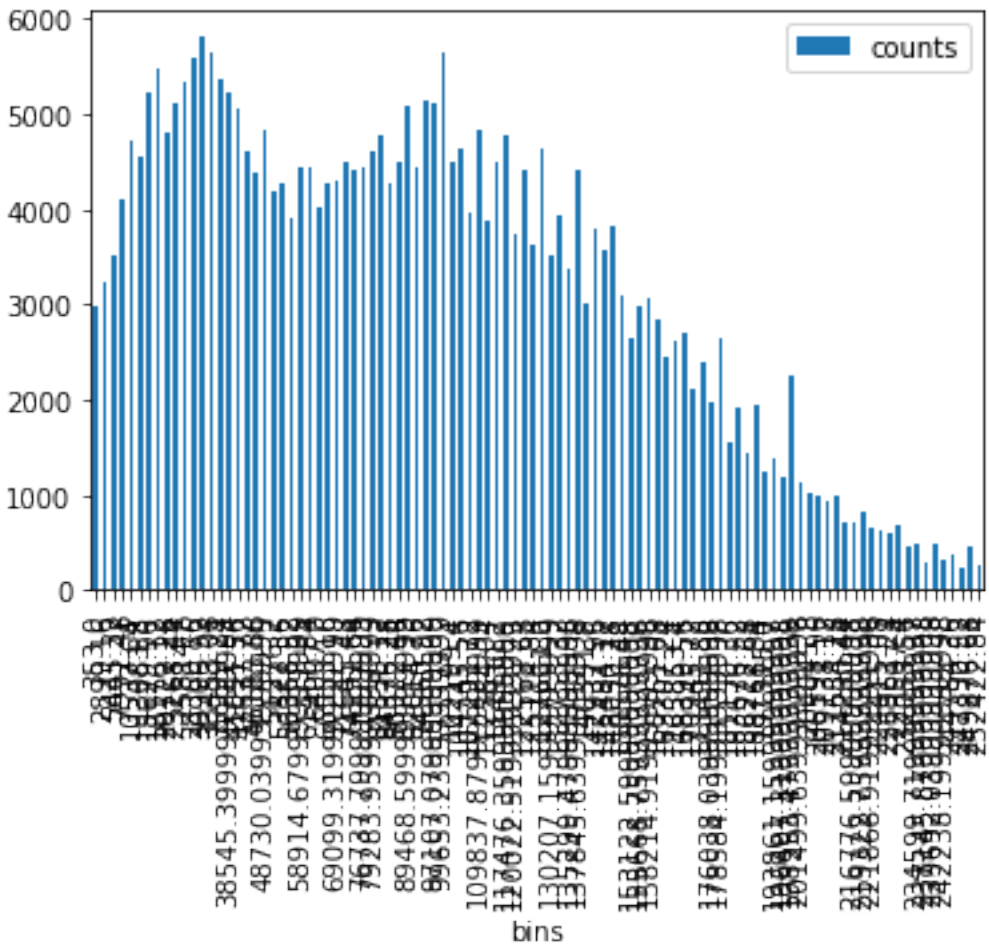
[86]:
```
%%spark -o hist_odometerOutliers

#Export hist_odometer

hist_odometerOutliers =  spark.createDataFrame(
    list(zip(*histogram_odometerOutliers)),
    ['bins', 'counts'])
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[87]:
```
%matplot plt
import matplotlib
import matplotlib.pyplot as plt
hist_odometerOutliers.set_index('bins'
            ).plot(kind='bar')
plt.show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```



[88]:
```
#Considering the skewed nature of the price histogram, we expected the same␣
 ↪result from the odometer histogram.
#The fact that these are skewed primarily to the right tells us that many users␣
 ↪posted very high odometer values and prices
#Why this may be for odometer is a bit less well understood than price, as a␣
 ↪high odometer often means extreme usage
#Nevertheless, it is safe to assume that our dataset is skewed more to the␣
 ↪right than to the left.
#It may be worth considering removing the top 3% rather than the bottom 1.5%␣
 ↪and top 1.5% of values
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

3 Modelling

[89]:
```
#Import psypark.ml.feature and numpy for modelling, as well as pipeline from␣
↪psypark.ml

import pyspark.ml.feature as feat
import numpy as np
from pyspark.ml import Pipeline
import pyspark.ml.regression as rg
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[90]:
```
#To begin modelling, we will create a new dataframe which drops the outlier␣
↪values recorded from our preprocessing

initial_modelling = price_outliers_removed.filter((fn.col('odometer') > 350) &␣
↪(fn.col('odometer') < 255000))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

3.1 Correlation Analysis

[91]:
```
#Here, we will begin analyzing the correlations between price and odometer␣
↪after removing our outliers

(
    initial_modelling
    .corr('odometer', 'price')
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
-0.5705109031992255
```

[92]:
```
#Here, we find a slight negative correlation, which makes sense considering the␣
↪nature of the used car market. As the odometer
#usage increases, we expect price to decrease accordingly. However, while the␣
↪odometer usage is a significant factor, it is only
```

```
#one of several potential variables in the used car market. Things like engine
 ↪and body condition, accident history, year and
#other factors can influence a car's value outside of odometer usage.
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[93]:
```
#We will also take a look at correlation between year and price using our new
 ↪dataframe

(
    initial_modelling
    .corr('year', 'price')
)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

0.4308014342134331

[94]:
```
#We have a slight positive correlation here between year and price, which is to
 ↪be expected. Newer vehicles are often better
#supported by manufacturers, and technicians are likely to have an easier time
 ↪repairing new vehicles as opposed to older ones
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[95]:
```
#Create a correlation table for numeric features

n_features = len(features)

corr = []

for i in range(0, n_features):
    temp = [None] * i

    for j in range(i, n_features):
        temp.append(initial_modelling.corr(features[i], features[j]))
    corr.append([features[i]] + temp)
```

```
correlations = spark.createDataFrame(corr, ['Column'] + features)

correlations.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+--------+----+-------------------+--------------------+
|  Column|year|              price|            odometer|
+--------+----+-------------------+--------------------+
|    year| 1.0|0.43080143421343314|-0.41909802936671653|
|   price|null|                1.0| -0.5705109031992255|
|odometer|null|               null|                 1.0|
+--------+----+-------------------+--------------------+
```

[96]:
```
#In our correlation table, we find and interesting negative correlation between␣
 ↪year and price. This makes sense, as the newer
#a car is, the less time a driver may have had to use the vehicle.
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

3.2 Data Transformation

[97]:
```
initial_modelling.printSchema()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
root
 |-- id: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- region: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- manufacturer: string (nullable = true)
 |-- model: string (nullable = true)
 |-- fuel: string (nullable = true)
 |-- odometer: integer (nullable = true)
 |-- type: string (nullable = true)
 |-- state: string (nullable = true)
```

[98]:
```
#Our first task in data transformation is to create dummy variables out of our␣
 ↪categorical variables

#In our data, we found that region, manufacturer model were particularly large,␣
 ↪and would be cumbersome to turn into
#categorical variables, with many categories having very uneven distributions␣
 ↪and outlier categories

#Thus, we decided to work with fuel and type for our dummy variables

#We found that state contained the actual text of the listing, and thus we␣
 ↪dropped it
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[99]:
```
#examine all distinct values for manufacturer

initial_modelling.select('manufacturer').groupBy('manufacturer').count().show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+---------------+-----+
|   manufacturer|count|
+---------------+-----+
|           null| 8714|
|          dodge|10894|
|        hyundai| 7482|
|        ferrari|    1|
|  mercedes-benz| 7559|
|harley-davidson|   96|
|          buick| 4223|
|     land rover|   17|
|         datsun|   39|
|          rover| 1206|
|           fiat|  763|
|     volkswagen| 7949|
|         nissan|16078|
|         toyota|27421|
|            bmw| 9488|
|           jeep|15143|
|           ford|52094|
|            ram|11667|
```

```
|   aston-martin|   15|
|           mini| 1880|
+---------------+-----+
only showing top 20 rows
```

[100]:
```
#examine all distinct values for model

initial_modelling.select('model').groupBy('model').count().show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


+--------------------+-----+
|               model|count|
+--------------------+-----+
|  cherokee trailhawk|   62|
|                hr-v|  111|
|2500 crew cab diesel|    1|
|    Volkswagon Jetta|    2|
|      e350 cargo van|   26|
|     f150 lariat 4x4|   35|
|      accord touring|   13|
|   aspen awd 3rd row|    3|
|      2002 FOR TAURS|    1|
|3500 crew cab lwb…|    1|
|           clk-class|   27|
|            ls 460 l|    8|
|             boxster|   74|
|      cheyenne super|    2|
|               es350|   76|
|                 v30|    1|
|                 van|   67|
|                 g35|  175|
|smart fortwo elec…|   26|
|transit connect c…|   54|
+--------------------+-----+
only showing top 20 rows
```

[101]:
```
#examine all distinct values for state

initial_modelling.select('state').groupBy('state').count().show()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+--------------------+-----+
|               state|count|
+--------------------+-----+
| Pay here"" finan…|   12|
| MECHANICALLY INS…|   19|
|        Luggage Rack|   31|
| Inc    Year: 202…|    1|
| CA 93710   Phone…|    1|
| Exterior/interio…|    1|
| blue tooth and m…|    1|
|           off lease|    6|
|750 below dealer …|    2|
| call 509-342-704…|    2|
| 4 Corner Air Rid…|    1|
|             Leather|   10|
| Driver And Passe…|    1|
|777  Year: 2014 M…|   11|
| voltage & oil pr…|   11|
| Metal-Look Conso…|    3|
| all new tiresBlu…|    2|
|980  Year: 2013 M…|   12|
| I'm told) Power …|    2|
| driver and front…|    4|
+--------------------+-----+
only showing top 20 rows
```

[ ]:

[102]:
```
#Drop state column and confirm

initial_modelling = initial_modelling.
 ↪select('id','year','region','price','year','manufacturer','model','fuel','odometer','type')

initial_modelling.printSchema()
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


root
 |-- id: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- region: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- manufacturer: string (nullable = true)
 |-- model: string (nullable = true)
```

```
 |-- fuel: string (nullable = true)
 |-- odometer: integer (nullable = true)
 |-- type: string (nullable = true)
```

3.3 Creating Dummies

[103]:
```
#We will create a new dataframe which creates dummies for fuel type

fuelCateg = initial_modelling.select('fuel').distinct().rdd.flatMap(lambda x:x).
 ↪collect()
fuelExprs = [fn.when(fn.col('fuel') == fuelCat,1).otherwise(0)\
             .alias(str(fuelCat)) for fuelCat in fuelCateg]
fuelDummy_df = initial_modelling.select(fuelExprs)

fuelDummy_df.show()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+---+----+------+--------+-----+------+
|gas|None|hybrid|electric|other|diesel|
+---+----+------+--------+-----+------+
|  0|   0|     0|       0|    0|     1|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  0|   0|     0|       0|    0|     1|
|  0|   0|     0|       0|    0|     1|
|  0|   0|     0|       0|    0|     1|
|  0|   0|     0|       0|    0|     1|
|  0|   0|     0|       0|    0|     1|
|  1|   0|     0|       0|    0|     0|
|  0|   0|     0|       0|    0|     1|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  1|   0|     0|       0|    0|     0|
|  0|   0|     0|       0|    0|     1|
+---+----+------+--------+-----+------+
only showing top 20 rows
```

[104]:
```
#We create another dataframe to create dummies for vehicle type

typeCateg = initial_modelling.select('type').distinct().rdd.flatMap(lambda x:x).
 ↪collect()
typeExprs = [fn.when(fn.col('type') == typeCat,1).otherwise(0)\
            .alias(str(typeCat)) for typeCat in typeCateg]
typeDummy_df = initial_modelling.select(typeExprs)

typeDummy_df.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+---+-----+----+---------+--------+---+---+-----+-------+-----+-----+-----------
+-----+------+
|van|wagon|None|hatchback|mini-
van|bus|SUV|truck|offroad|coupe|other|convertible|sedan|pickup|
+---+-----+----+---------+--------+---+---+-----+-------+-----+-----+-----------
+-----+------+
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  1|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
```

```
1|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    1|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     1|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    1|     0|
|  0|    0|   0|        0|       0|  0|  0|    0|      0|    0|    0|
0|    0|     0|
|  0|    0|   0|        0|       0|  0|  1|    0|      0|    0|    0|
0|    0|     0|
+---+-----+----+---------+--------+---+---+-----+-------+-----+-----+-----------
+-----+------+
only showing top 20 rows
```

[105]:
```
#Create placeholder dataframe which drops categorical variables from␣
↪initial_modelling dataframe

placeholder_df =initial_modelling.select('id','price','year','odometer')



#Now we will append the dummy dataframes to our main dataframe, creating new␣
↪column names to avoid ambiguity

combined_df = initial_modelling.select(placeholder_df.columns + fuelExprs +␣
↪typeExprs)

newColumns =␣
↪['id','price','year','odometer','gas','nullFuel','hybrid','electric','otherFuel','diesel','
dummyModelling = combined_df.toDF(*newColumns)
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

[106]:
```
#Confirm new dataframe does not have ambiguous names

dummyModelling.show()
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px

```
+----------+-----+----+--------+---+--------+------+--------+---------+------+--
-+-----+--------+---------+--------+---+---+-----+-------+-----+---------+------
------+-----+------+
|        id|price|year|odometer|gas|nullFuel|hybrid|electric|otherFuel|diesel|va
n|wagon|nullType|hatchback|mini-
van|bus|SUV|truck|offroad|coupe|otherType|convertible|sedan|pickup|
+----------+-----+----+--------+---+--------+------+--------+---------+------+--
-+-----+--------+---------+--------+---+---+-----+-------+-----+---------+------
------+-----+------+
|7237761307|28990|2003|   91567|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7237366028|19988|2009|   93562|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7239170981| 3995|2009|  238000|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7235356070| 4500|2004|  137451|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  1|    0|      0|    0|        0|
0|    0|     0|
|7240709284| 5900|2004|  250000|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7226776086|24999|2006|  160186|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7237477082|36500|2012|  126941|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7234532298|18999|2003|  188745|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7236102344| 9000|2003|  215370|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7226975171| 6490|2011|  125081|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7238072478|34990|2012|  153860|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7230815357|45991|2014|   48100|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7240535821|42900|1933|    5289|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
```

```
|7229038332|19500|1969|   23233|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
1|    0|     0|
|7239724042| 4500|2000|  190625|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    1|      0|    0|        0|
0|    0|     0|
|7238449510| 1999|1989|  200739|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     1|
|7240405045| 8500|2002|  155047|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7239986191|47000|1955|    7539|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    1|     0|
|7240265288|30000|2012|   40000|  1|       0|     0|       0|        0|     0|
0|    0|       0|        0|       0|  0|  0|    0|      0|    0|        0|
0|    0|     0|
|7240647559|13000|1999|  220000|  0|       0|     0|       0|        0|     1|
0|    0|       0|        0|       0|  0|  1|    0|      0|    0|        0|
0|    0|     0|
+----------+-----+----+--------+---+--------+------+--------+---------+------+--
-+-----+--------+---------+--------+---+---+-----+-------+-----+---------+------
-----+-----+------+
only showing top 20 rows
```

4 Regression Analysis

[107]:
```
#We will use Linear Regression as well as Gradient Boosted Trees
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[108]:
```
#We want to look at linear regression to
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

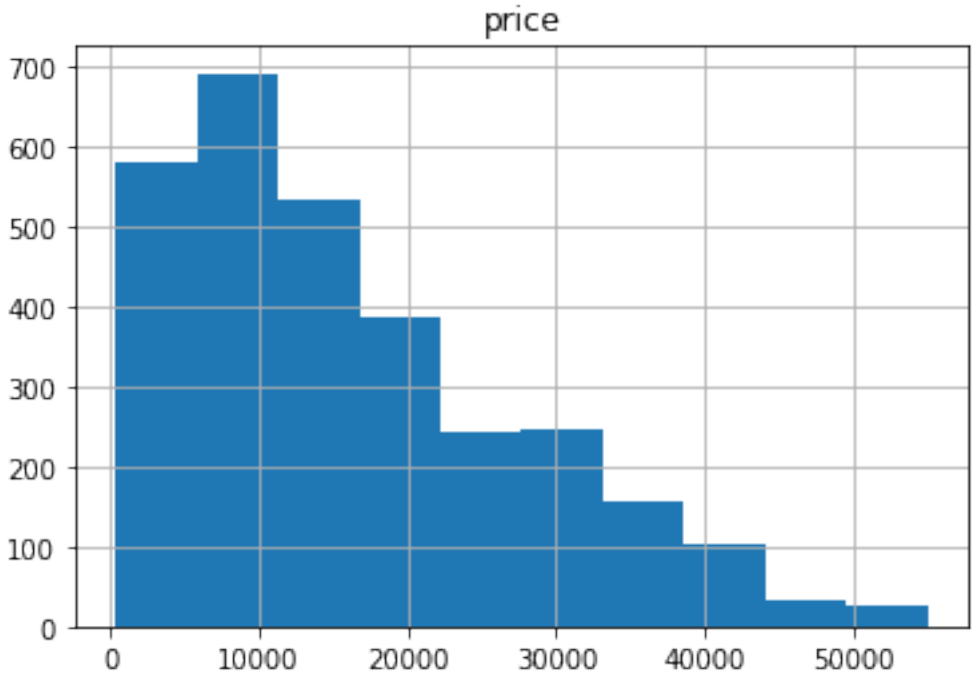
[109]:
```
%%spark -o transformed_df -m sample -n 3000

#Take a random sample from our dataset to see potential shape

transformed_df = dummyModelling.select('price')
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[110]:
```
%%local

#Draw a histogram of our random sample

transformed_df.hist()
```

[110]:
```
array([[<AxesSubplot:title={'center':'price'}>]], dtype=object)
```



[111]:
```
#As noted in preprocessing, our data is skewed to the right.
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

4.1 Correlation Matrix

[130]:
```
#install S3 package needed for saving correlation output to S3
sc.install_pypi_package('s3fs')
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


Collecting s3fs
  Downloading https://files.pythonhosted.org/packages/f7/57/f22362bc2d11ed7a7fa9
4f0997b1229c73b4f28d8595d0fdc78802c011f6/s3fs-2021.4.0-py3-none-any.whl
Collecting fsspec==2021.04.0 (from s3fs)
  Downloading https://files.pythonhosted.org/packages/e9/91/2ef649137816850fa4f4
c97c6f2eabb1a79bf0aa2c8ed198e387e373455e/fsspec-2021.4.0-py3-none-any.whl
(108kB)
Collecting aiobotocore>=1.0.1 (from s3fs)
  Downloading https://files.pythonhosted.org/packages/21/8e/4562029e179226051cd4
aa3135444deb014fc9b0795f80f7f3563745f8d5/aiobotocore-1.3.0.tar.gz (48kB)
Collecting botocore<1.20.50,>=1.20.49 (from aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/68/59/6e28ce58206039ad2592
992b75ee79a8f9dbc902a9704373ddacc4f96300/botocore-1.20.49-py2.py3-none-any.whl
(7.4MB)
Collecting aiohttp>=3.3.1 (from aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/99/f5/90ede947a3ce2d6de161
4799f5fea4e93c19b6520a59dc5d2f64123b032f/aiohttp-3.7.4.post0.tar.gz (1.1MB)
Collecting wrapt>=1.10.10 (from aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/82/f7/e43cefbe88c5fd371f4c
f0cf5eb3feccd07515af9fd6cf7dbf1d1793a797/wrapt-1.12.1.tar.gz
Collecting aioitertools>=0.5.1 (from aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/32/0b/3260ac050de07bf6e918
71944583bb8598091da19155c34f7ef02244709c/aioitertools-0.7.1-py3-none-any.whl
Collecting urllib3<1.27,>=1.25.4 (from
botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/09/c6/d3e3abe5b4f4f16cf0df
c9240ab7ce10c2baa0e268989a4e3ec19e90c84e/urllib3-1.26.4-py2.py3-none-any.whl
(153kB)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/usr/local/lib/python3.7/site-packages (from
botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in
/usr/local/lib/python3.7/site-packages (from
botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs)
Collecting attrs>=17.3.0 (from aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/c3/aa/cb45262569fcc047bf07
0b5de61813724d6726db83259222cd7b4c79821a/attrs-20.3.0-py2.py3-none-any.whl
(49kB)
```

```
Collecting chardet<5.0,>=2.0 (from aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/19/c7/fa589626997dd07bd87d
9269342ccb74b1720384a4d739a1872bd84fbe68/chardet-4.0.0-py2.py3-none-any.whl
(178kB)
Collecting multidict<7.0,>=4.5 (from aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/1c/74/e8b46156f37ca56d10d8
95d4e8595aa2b344cff3c1fb3629ec97a8656ccb/multidict-5.1.0.tar.gz (53kB)
Collecting async_timeout<4.0,>=3.0 (from
aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/e1/1e/5a4441be21b0726c4464
f3f23c8b19628372f606755a9d2e46c187e65ec4/async_timeout-3.0.1-py3-none-any.whl
Collecting yarl<2.0,>=1.0 (from aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/97/e7/af7219a0fe240e8ef6bb
555341a63c43045c21ab0392b4435e754b716fa1/yarl-1.6.3.tar.gz (176kB)
Collecting typing_extensions>=3.6.5 (from
aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/2e/35/6c4fff5ab443b57116cb
1aad46421fb719bed2825664e8fe77d66d99bcbc/typing_extensions-3.10.0.0-py3-none-
any.whl
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-
dateutil<3.0.0,>=2.1->botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs)
Collecting idna>=2.0 (from
yarl<2.0,>=1.0->aiohttp>=3.3.1->aiobotocore>=1.0.1->s3fs)
  Downloading https://files.pythonhosted.org/packages/29/88/c52aae187d3b128a0f13
f36a6c987fc0d408d03a678ad9996516925d8495/idna-3.1-py3-none-any.whl (58kB)
Building wheels for collected packages: aiobotocore, aiohttp, wrapt, multidict,
yarl
  Running setup.py bdist_wheel for aiobotocore: started
  Running setup.py bdist_wheel for aiobotocore: finished with status 'done'
  Stored in directory: /var/lib/livy/.cache/pip/wheels/1d/5b/92/72a4721010997136
4c619c41e29289d4e7d58269f0cc653bf1
  Running setup.py bdist_wheel for aiohttp: started
  Running setup.py bdist_wheel for aiohttp: finished with status 'done'
  Stored in directory: /var/lib/livy/.cache/pip/wheels/15/aa/5f/33df380f4940b1c1
bda8d83967345fcb97d0749e2cfbb06794
  Running setup.py bdist_wheel for wrapt: started
  Running setup.py bdist_wheel for wrapt: finished with status 'done'
  Stored in directory: /var/lib/livy/.cache/pip/wheels/b1/c2/ed/d62208260edbd3fa
7156545c00ef966f45f2063d0a84f8208a
  Running setup.py bdist_wheel for multidict: started
  Running setup.py bdist_wheel for multidict: finished with status 'error'
  Complete output from command /tmp/1620177931521-0/bin/python -u -c "import
setuptools, tokenize;__file__='/mnt/tmp/pip-
build-9gi8b506/multidict/setup.py';f=getattr(tokenize, 'open',
open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code,
__file__, 'exec'))" bdist_wheel -d /tmp/tmpsn9zfccepip-wheel- --python-tag cp37:
  ********************
```

```
  * Accelerated build *
  **********************
  /usr/lib64/python3.7/distutils/dist.py:274: UserWarning: Unknown distribution
option: 'project_urls'
    warnings.warn(msg)
  running bdist_wheel
  running build
  running build_py
  creating build
  creating build/lib.linux-x86_64-3.7
  creating build/lib.linux-x86_64-3.7/multidict
  copying multidict/__init__.py -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/_abc.py -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/_compat.py -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/_multidict_base.py -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/_multidict_py.py -> build/lib.linux-x86_64-3.7/multidict
  running egg_info
  writing multidict.egg-info/PKG-INFO
  writing dependency_links to multidict.egg-info/dependency_links.txt
  writing top-level names to multidict.egg-info/top_level.txt
  warning: manifest_maker: standard file '-c' not found

  reading manifest file 'multidict.egg-info/SOURCES.txt'
  reading manifest template 'MANIFEST.in'
  warning: no previously-included files matching '*.pyc' found anywhere in
distribution
  warning: no previously-included files found matching
'multidict/_multidict.html'
  warning: no previously-included files found matching 'multidict/*.so'
  warning: no previously-included files found matching 'multidict/*.pyd'
  warning: no previously-included files found matching 'multidict/*.pyd'
  no previously-included directories found matching 'docs/_build'
  writing manifest file 'multidict.egg-info/SOURCES.txt'
  copying multidict/__init__.pyi -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/_multidict.c -> build/lib.linux-x86_64-3.7/multidict
  copying multidict/py.typed -> build/lib.linux-x86_64-3.7/multidict
  creating build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/defs.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/dict.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/istr.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/iter.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/pair_list.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
  copying multidict/_multilib/views.h ->
```

```
build/lib.linux-x86_64-3.7/multidict/_multilib
  running build_ext
  building 'multidict._multidict' extension
  creating build/temp.linux-x86_64-3.7
  creating build/temp.linux-x86_64-3.7/multidict
  gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -O2 -g -pipe -Wall
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-
buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D_GNU_SOURCE -fPIC
-fwrapv -fPIC -I/usr/include/python3.7m -c multidict/_multidict.c -o
build/temp.linux-x86_64-3.7/multidict/_multidict.o -O2 -std=c99 -Wall -Wsign-
compare -Wconversion -fno-strict-aliasing -pedantic
  multidict/_multidict.c:1:10: fatal error: Python.h: No such file or directory
   #include "Python.h"
            ^~~~~~~~~~
  compilation terminated.
  error: command 'gcc' failed with exit status 1

  ----------------------------------------
  Running setup.py clean for multidict
  Running setup.py bdist_wheel for yarl: started
  Running setup.py bdist_wheel for yarl: finished with status 'error'
  Complete output from command /tmp/1620177931521-0/bin/python -u -c "import
setuptools, tokenize;__file__='/mnt/tmp/pip-
build-9gi8b506/yarl/setup.py';f=getattr(tokenize, 'open',
open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code,
__file__, 'exec'))" bdist_wheel -d /tmp/tmprh043tgbpip-wheel- --python-tag cp37:
  **********************
  * Accellerated build *
  **********************
  /usr/lib64/python3.7/distutils/dist.py:274: UserWarning: Unknown distribution
option: 'long_description_content_type'
    warnings.warn(msg)
  running bdist_wheel
  running build
  running build_py
  creating build
  creating build/lib.linux-x86_64-3.7
  creating build/lib.linux-x86_64-3.7/yarl
  copying yarl/__init__.py -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_quoting.py -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_quoting_py.py -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_url.py -> build/lib.linux-x86_64-3.7/yarl
  running egg_info
  writing yarl.egg-info/PKG-INFO
  writing dependency_links to yarl.egg-info/dependency_links.txt
  writing requirements to yarl.egg-info/requires.txt
  writing top-level names to yarl.egg-info/top_level.txt
  warning: manifest_maker: standard file '-c' not found
```

```
  reading manifest file 'yarl.egg-info/SOURCES.txt'
  reading manifest template 'MANIFEST.in'
  warning: no previously-included files matching '*.pyc' found anywhere in
distribution
  warning: no previously-included files matching '*.cache' found anywhere in
distribution
  warning: no previously-included files found matching 'yarl/*.html'
  warning: no previously-included files found matching 'yarl/*.so'
  warning: no previously-included files found matching 'yarl/*.pyd'
  no previously-included directories found matching 'docs/_build'
  writing manifest file 'yarl.egg-info/SOURCES.txt'
  copying yarl/__init__.pyi -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_quoting_c.c -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_quoting_c.pyi -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/_quoting_c.pyx -> build/lib.linux-x86_64-3.7/yarl
  copying yarl/py.typed -> build/lib.linux-x86_64-3.7/yarl
  running build_ext
  building 'yarl._quoting_c' extension
  creating build/temp.linux-x86_64-3.7
  creating build/temp.linux-x86_64-3.7/yarl
  gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -O2 -g -pipe -Wall
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-
buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D_GNU_SOURCE -fPIC
-fwrapv -fPIC -I/usr/include/python3.7m -c yarl/_quoting_c.c -o
build/temp.linux-x86_64-3.7/yarl/_quoting_c.o
  yarl/_quoting_c.c:4:10: fatal error: Python.h: No such file or directory
   #include "Python.h"
            ^~~~~~~~~~
  compilation terminated.
  error: command 'gcc' failed with exit status 1

  ----------------------------------------
  Running setup.py clean for yarl
Successfully built aiobotocore aiohttp wrapt
Failed to build multidict yarl
Installing collected packages: fsspec, urllib3, botocore, attrs, chardet,
multidict, async-timeout, idna, typing-extensions, yarl, aiohttp, wrapt,
aioitertools, aiobotocore, s3fs
  Running setup.py install for multidict: started
    Running setup.py install for multidict: finished with status 'error'
    Complete output from command /tmp/1620177931521-0/bin/python -u -c "import
setuptools, tokenize;__file__='/mnt/tmp/pip-
build-9gi8b506/multidict/setup.py';f=getattr(tokenize, 'open',
open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code,
__file__, 'exec'))" install --record /tmp/pip-euu6dse6-record/install-record.txt
--single-version-externally-managed --compile --install-headers
/tmp/1620177931521-0/include/site/python3.7/multidict:
```

```
    **********************
    * Accelerated build *
    **********************
    /usr/lib64/python3.7/distutils/dist.py:274: UserWarning: Unknown
distribution option: 'project_urls'
      warnings.warn(msg)
    running install
    running build
    running build_py
    creating build
    creating build/lib.linux-x86_64-3.7
    creating build/lib.linux-x86_64-3.7/multidict
    copying multidict/__init__.py -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/_abc.py -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/_compat.py -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/_multidict_base.py -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/_multidict_py.py -> build/lib.linux-x86_64-3.7/multidict
    running egg_info
    writing multidict.egg-info/PKG-INFO
    writing dependency_links to multidict.egg-info/dependency_links.txt
    writing top-level names to multidict.egg-info/top_level.txt
    warning: manifest_maker: standard file '-c' not found

    reading manifest file 'multidict.egg-info/SOURCES.txt'
    reading manifest template 'MANIFEST.in'
    warning: no previously-included files matching '*.pyc' found anywhere in
distribution
    warning: no previously-included files found matching
'multidict/_multidict.html'
    warning: no previously-included files found matching 'multidict/*.so'
    warning: no previously-included files found matching 'multidict/*.pyd'
    warning: no previously-included files found matching 'multidict/*.pyd'
    no previously-included directories found matching 'docs/_build'
    writing manifest file 'multidict.egg-info/SOURCES.txt'
    copying multidict/__init__.pyi -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/_multidict.c -> build/lib.linux-x86_64-3.7/multidict
    copying multidict/py.typed -> build/lib.linux-x86_64-3.7/multidict
    creating build/lib.linux-x86_64-3.7/multidict/_multilib
    copying multidict/_multilib/defs.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
    copying multidict/_multilib/dict.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
    copying multidict/_multilib/istr.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
    copying multidict/_multilib/iter.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
    copying multidict/_multilib/pair_list.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
```

```
    copying multidict/_multilib/views.h ->
build/lib.linux-x86_64-3.7/multidict/_multilib
    running build_ext
    building 'multidict._multidict' extension
    creating build/temp.linux-x86_64-3.7
    creating build/temp.linux-x86_64-3.7/multidict
    gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -O2 -g -pipe -Wall
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-
buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D_GNU_SOURCE -fPIC
-fwrapv -fPIC -I/usr/include/python3.7m -c multidict/_multidict.c -o
build/temp.linux-x86_64-3.7/multidict/_multidict.o -O2 -std=c99 -Wall -Wsign-
compare -Wconversion -fno-strict-aliasing -pedantic
    multidict/_multidict.c:1:10: fatal error: Python.h: No such file or
directory
     #include "Python.h"
              ^~~~~~~~~~
    compilation terminated.
    error: command 'gcc' failed with exit status 1

    ----------------------------------------


  Ignoring idna-ssl: markers 'python_version < "3.7"' don't match your
environment
  Failed building wheel for multidict
  Failed building wheel for yarl
Command "/tmp/1620177931521-0/bin/python -u -c "import setuptools,
tokenize;__file__='/mnt/tmp/pip-
build-9gi8b506/multidict/setup.py';f=getattr(tokenize, 'open',
open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code,
__file__, 'exec'))" install --record /tmp/pip-euu6dse6-record/install-record.txt
--single-version-externally-managed --compile --install-headers
/tmp/1620177931521-0/include/site/python3.7/multidict" failed with error code 1
in /mnt/tmp/pip-build-9gi8b506/multidict/
```

[133]:
```
#%%spark -o corr
import pyspark.ml.stat as st
import numpy as np
import pandas as pd

features_and_label = feat.VectorAssembler(
    inputCols=dummyModelling.columns[1:]
    , outputCol='features'
)

corr = st.Correlation.corr(
    features_and_label.transform(dummyModelling),
    'features',
```

```
    'pearson'
)

print(str(corr.collect()[0][0]))
corr_pd = corr.toPandas()
output_np = np.array(corr_pd.iloc[0, 0].values).reshape(
    (corr_pd.iloc[0, 0].numRows, corr_pd.iloc[0, 0].numCols))

#Change the following path to a path in your own S3 bucket
spark.createDataFrame(pd.DataFrame(output_np)).repartition(1).write.
 ↪format('csv').option('header',True
               ).mode('overwrite').option('sep',',').save('s3://
 ↪pjgarrido-cis4567-project-bucket/Dataset')
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
Exception in thread cell_monitor-120:
Traceback (most recent call last):
  File "/emr/notebook-env/lib/python3.7/threading.py", line 926, in
_bootstrap_inner
    self.run()
  File "/emr/notebook-env/lib/python3.7/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "/emr/notebook-env/lib/python3.7/site-packages/awseditorssparkmonitoringw
idget-1.0-py3.7.egg/awseditorssparkmonitoringwidget/cellmonitor.py", line 178,
in cell_monitor
    job_binned_stages[job_id][stage_id] = all_stages[stage_id]
KeyError: 611
```

```
DenseMatrix([[ 1.        ,  0.43080143, -0.5705109 , -0.21885627,         nan,
              -0.03704987,  0.04309502,  0.10282762,  0.22354085, -0.00861382,
              -0.05379319,         nan, -0.07872188, -0.07430157,  0.01367584,
              -0.03778992,  0.14968765,  0.00259144,  0.0581629 ,  0.16135421,
               0.00604597, -0.19975546,  0.28948742],
             [ 0.43080143,  1.        , -0.41909803, -0.04871115,         nan,
               0.02456369,  0.04369694,  0.07149515, -0.02350898,  0.01196387,
               0.00451021,         nan,  0.05050632, -0.0178912 , -0.01700122,
               0.04537999, -0.02570903, -0.04342587, -0.07451376,  0.07600297,
              -0.11290725,  0.02898478,  0.08126287],
             [-0.5705109 , -0.41909803,  1.        , -0.02313944,         nan,
               0.00343517, -0.07419397, -0.0658567 ,  0.11059376,  0.011619  ,
               0.01793931,         nan, -0.07042165,  0.06399666,  0.0037175 ,
               0.06306598,  0.06230247,  0.0061581 , -0.10235736, -0.13026043,
              -0.0451273 , -0.02258258, -0.08067764],
```

```
[-0.21885627, -0.04871115, -0.02313944,  1.        ,         nan,
 -0.32810678, -0.2170288 , -0.51260094, -0.67927513,  0.00707191,
  0.02066121,         nan, -0.06831512,  0.03133823, -0.0355455 ,
  0.10038769, -0.11870913,  0.00932114,  0.05161072, -0.01961551,
  0.04113883,  0.07561246, -0.16658801],
[        nan,         nan,         nan,         nan,  1.        ,
         nan,         nan,         nan,         nan,         nan,
         nan,         nan,         nan,         nan,         nan,
         nan,         nan,         nan,         nan,         nan,
         nan,         nan,         nan],
[-0.03704987,  0.02456369,  0.00343517, -0.32810678,         nan,
  1.        , -0.00890783, -0.02103942, -0.02788047, -0.01767159,
  0.00493999,         nan,  0.16005777, -0.01265593, -0.00419707,
 -0.02945569, -0.02752474, -0.00438889, -0.02122313, -0.00509575,
 -0.01394879,  0.02740345, -0.03926475],
[ 0.04309502,  0.04369694, -0.07419397, -0.2170288 ,         nan,
 -0.00890783,  1.        , -0.01391669, -0.01844176, -0.0114064 ,
 -0.0036693 ,         nan,  0.14718412, -0.00855161, -0.00277618,
 -0.03475659, -0.02155453, -0.00178935, -0.00631268, -0.00216284,
 -0.00830439,  0.02742613, -0.02651274],
[ 0.10282762,  0.07149515, -0.0658567 , -0.51260094,         nan,
 -0.02103942, -0.01391669,  1.        , -0.04355764,  0.0149023 ,
 -0.0128658 ,         nan,  0.00930089, -0.00670013, -0.00503833,
 -0.03239825, -0.04009071, -0.00540422, -0.00570952,  0.04799105,
 -0.01820387, -0.03791723,  0.16930868],
[ 0.22354085, -0.02350898,  0.11059376, -0.67927513,         nan,
 -0.02788047, -0.01844176, -0.04355764,  1.        , -0.0118991 ,
 -0.01656391,         nan, -0.04009701, -0.02882971,  0.05703599,
 -0.09263321,  0.22266602, -0.00497043, -0.05042785, -0.00566517,
 -0.03015602, -0.10017474,  0.12322146],
[-0.00861382,  0.01196387,  0.011619  ,  0.00707191,         nan,
 -0.01767159, -0.0114064 ,  0.0149023 , -0.0118991 ,  1.        ,
 -0.02438264,         nan, -0.03222472, -0.01900219, -0.00550747,
 -0.07197475, -0.04276046, -0.00575919, -0.03406497, -0.0378198 ,
 -0.01974602, -0.08039358, -0.0525967 ],
[-0.05379319,  0.00451021,  0.01793931,  0.02066121,         nan,
  0.00493999, -0.0036693 , -0.0128658 , -0.01656391, -0.02438264,
  1.        ,         nan, -0.03388348, -0.01998032, -0.00579096,
 -0.07567964, -0.04496155, -0.00605564, -0.03581847, -0.03976657,
 -0.02076244, -0.08453183, -0.05530412],
[        nan,         nan,         nan,         nan,         nan,
         nan,         nan,         nan,         nan,         nan,
         nan,  1.        ,         nan,         nan,         nan,
         nan,         nan,         nan,         nan,         nan,
         nan,         nan,         nan],
[-0.07872188,  0.05050632, -0.07042165, -0.06831512,         nan,
  0.16005777,  0.14718412,  0.00930089, -0.04009701, -0.03222472,
 -0.03388348,         nan,  1.        , -0.0264065 , -0.00765348,
```

```
 -0.10002015, -0.05942233, -0.00800329, -0.0473386 , -0.05255651,
 -0.02744017, -0.11171942, -0.07309133],
[-0.07430157, -0.0178912 ,  0.06399666,  0.03133823,         nan,
 -0.01265593, -0.00855161, -0.00670013, -0.02882971, -0.01900219,
 -0.01998032,         nan, -0.0264065 ,  1.        , -0.00451309,
 -0.05897962, -0.03504001, -0.00471936, -0.0279145 , -0.03099139,
 -0.01618085, -0.06587842, -0.0431003 ],
[ 0.01367584, -0.01700122,  0.0037175 , -0.0355455 ,         nan,
 -0.00419707, -0.00277618, -0.00503833,  0.05703599, -0.00550747,
 -0.00579096,         nan, -0.00765348, -0.00451309,  1.        ,
 -0.01709425, -0.01015576, -0.00136783, -0.00809055, -0.00898233,
 -0.00468975, -0.01909376, -0.0124919 ],
[-0.03778992,  0.04537999,  0.06306598,  0.10038769,         nan,
 -0.02945569, -0.03475659, -0.03239825, -0.09263321, -0.07197475,
 -0.07567964,         nan, -0.10002015, -0.05897962, -0.01709425,
  1.        , -0.13272134, -0.01787555, -0.10573199, -0.11738634,
 -0.06128834, -0.24952825, -0.1632514 ],
[ 0.14968765, -0.02570903,  0.06230247, -0.11870913,         nan,
 -0.02752474, -0.02155453, -0.04009071,  0.22266602, -0.04276046,
 -0.04496155,         nan, -0.05942233, -0.03504001, -0.01015576,
 -0.13272134,  1.        , -0.01061993, -0.06281576, -0.06973965,
 -0.03641163, -0.14824564, -0.09698825],
[ 0.00259144, -0.04342587,  0.0061581 ,  0.00932114,         nan,
 -0.00438889, -0.00178935, -0.00540422, -0.00497043, -0.00575919,
 -0.00605564,         nan, -0.00800329, -0.00471936, -0.00136783,
 -0.01787555, -0.01061993,  1.        , -0.00846033, -0.00939287,
 -0.00490409, -0.01996644, -0.01306285],
[ 0.0581629 , -0.07451376, -0.10235736,  0.05161072,         nan,
 -0.02122313, -0.00631268, -0.00570952, -0.05042785, -0.03406497,
 -0.03581847,         nan, -0.0473386 , -0.0279145 , -0.00809055,
 -0.10573199, -0.06281576, -0.00846033,  1.        , -0.05555785,
 -0.0290072 , -0.11809938, -0.07726535],
[ 0.16135421,  0.07600297, -0.13026043, -0.01961551,         nan,
 -0.00509575, -0.00216284,  0.04799105, -0.00566517, -0.0378198 ,
 -0.03976657,         nan, -0.05255651, -0.03099139, -0.00898233,
 -0.11738634, -0.06973965, -0.00939287, -0.05555785,  1.        ,
 -0.03220452, -0.13111693, -0.08578196],
[ 0.00604597, -0.11290725, -0.0451273 ,  0.04113883,         nan,
 -0.01394879, -0.00830439, -0.01820387, -0.03015602, -0.01974602,
 -0.02076244,         nan, -0.02744017, -0.01618085, -0.00468975,
 -0.06128834, -0.03641163, -0.00490409, -0.0290072 , -0.03220452,
  1.        , -0.06845719, -0.04478744],
[-0.19975546,  0.02898478, -0.02258258,  0.07561246,         nan,
  0.02740345,  0.02742613, -0.03791723, -0.10017474, -0.08039358,
 -0.08453183,         nan, -0.11171942, -0.06587842, -0.01909376,
 -0.24952825, -0.14824564, -0.01996644, -0.11809938, -0.13111693,
 -0.06845719,  1.        , -0.18234678],
[ 0.28948742,  0.08126287, -0.08067764, -0.16658801,         nan,
```

```
-0.03926475, -0.02651274,  0.16930868,  0.12322146, -0.0525967 ,
-0.05530412,         nan, -0.07309133, -0.0431003 , -0.0124919 ,
-0.1632514 , -0.09698825, -0.01306285, -0.07726535, -0.08578196,
-0.04478744, -0.18234678,  1.        ]])
```

4.2 Chi Square Selector

[112]:
```
#We will use the Chi Square Selector
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[113]:
```
#split into train and test sets
dummyModelling_train, dummyModelling_test = (
    dummyModelling
    .randomSplit([0.1, 0.9], seed=666)
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

[114]:
```
#exclude the label CoverType from features vector
vectorAssembler = feat.VectorAssembler(
    inputCols=dummyModelling.columns[2:]
    , outputCol='features'
)

#select top 10 features, store in a new column named selected
selector = feat.ChiSqSelector(
    labelCol='price'
    , numTopFeatures=4
    , outputCol='selected')

pipeline_sel = Pipeline(stages=[vectorAssembler, selector])

model = (
    pipeline_sel
    .fit(dummyModelling_train)
    .transform(dummyModelling_train)
)

#print selected features
model.schema['selected'].metadata
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
{'ml_attr': {'attrs': {'numeric': [{'idx': 0, 'name': 'odometer'}, {'idx': 1,
'name': 'gas'}, {'idx': 2, 'name': 'otherFuel'}, {'idx': 3, 'name': 'diesel'}]},
'num_attrs': 4}}
```

4.3 Linear Regression

[115]:
```
#Create pipeline for linear regression

from pyspark.ml import Pipeline

#Vectorize dataset and select odometer column

vectorAssembler = feat.VectorAssembler(
    inputCols=dummyModelling.columns[2:]
    , outputCol='features')

#We will make use of Generalized Linear Regression to normalize our data, as␣
 ↪the variable price is skewed to the right

lr_obj = rg.GeneralizedLinearRegression(
    labelCol='price'
    , maxIter=10
    , regParam=0.01
    , link='identity'
    , linkPredictionCol="p"
)



#Create pipline, using our vectorAssembler and lr_obj as stages

pip = Pipeline(stages=[vectorAssembler, lr_obj])

#Run pipeline

(
    pip
    .fit(dummyModelling)
    .transform(dummyModelling)
    .select('price', 'prediction')
    .show(20)
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
+-----+------------------+
|price|        prediction|
+-----+------------------+
|28990| 30690.78042147588|
|19988| 21552.36833836499|
| 3995|-18.23074095067568|
| 4500| 8904.297072896967|
| 5900| 8118.117605358129|
|24999| 25121.79390700301|
|36500|30388.154880580725|
|18999|21340.051200983347|
| 9000|18778.121901186998|
| 6490|11536.285755586345|
|34990| 27797.93610795273|
|45991| 19977.32388150238|
|42900|-3813.691441253992|
|19500| 9283.815853274427|
| 4500| 8094.583734259824|
| 1999| 4348.050301550073|
| 8500| 5551.720293713617|
|47000| 111.2777222651057|
|30000|20067.581917624688|
|13000|10252.230389296543|
+-----+------------------+
only showing top 20 rows
```

[116]:
```
import pyspark.ml.regression as rg

# let's predict elevation (first column) using the rest of features
vectorAssembler = feat.VectorAssembler(
    inputCols=dummyModelling.columns[2:]
    , outputCol='features')

#Create price as a label
price_dataset = (
    vectorAssembler
    .transform(dummyModelling)
    .withColumn(
        'label'
        , fn.col('price'))
    .select('label', 'features')
)
```

```
#create a linear regression object and fit to dataset
lr_obj = rg.LinearRegression(
    maxIter=10
    , regParam=0.01
    , elasticNetParam=1.00)
lr_model = lr_obj.fit(price_dataset)

#examine model coefficients
lr_model.coefficients
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
DenseVector([341.3034, -0.0965, -2296.8602, 0.0, -2301.0161, 2289.0017,
-381.5907, 8647.6696, 590.8701, -2198.7428, 0.0, -5736.437, -2225.577, 3197.068,
945.9438, 6632.6543, 5931.0413, 2777.4282, 4672.7613, 2355.4883, -3454.7273,
7644.5016])
```

[117]:
```
summary = lr_model.summary

print(
    summary.r2
    , summary.rootMeanSquaredError
    , summary.meanAbsoluteError
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
0.5570972285758827 7750.9357774963555 5692.776489954486
```

4.4 Gradient Boosted Trees

[118]:
```
#import evaluator as ev
import pyspark.ml.evaluation as ev

#Create pipeline for Gradient Boosted Trees

gbt_obj = rg.GBTRegressor(
    labelCol='price'
    , minInstancesPerNode=10
    , minInfoGain=0.1
)
```

```
pip = Pipeline(stages=[vectorAssembler, gbt_obj])

results = (
    pip
    .fit(dummyModelling)
    .transform(dummyModelling)
    .select('price', 'prediction')
)

evaluator = ev.RegressionEvaluator(labelCol='price')
evaluator.evaluate(results, {evaluator.metricName: 'r2'})
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
0.6311562692628663
```

4.5 Tuning Hyperparameters

[127]:
```
import pyspark.ml.tuning as tune

vectorAssembler = feat.VectorAssembler(
    inputCols=dummyModelling.columns[2:]
    , outputCol='features')

#Create price as a label
price_dataset = (
    vectorAssembler
    .transform(dummyModelling_train)
    .withColumn(
        'label'
        , fn.col('price'))
    .select('label', 'features')
)

selector = feat.ChiSqSelector(
    labelCol='price'
    , numTopFeatures=4
    , outputCol='selected')

lr_obj = rg.LinearRegression(
    maxIter=10
    , regParam=0.01
    , elasticNetParam=1.00)
lr_model = lr_obj.fit(price_dataset)
```

```
#use ParamGridBuilder to build a grid of parameters
lr_grid = (
    tune.ParamGridBuilder()
    #try 2 values for regParam
    .addGrid(lr_obj.regParam
             , [0.01, 0.1]
        )
    #try 2 values for elasticNetParam
    .addGrid(lr_obj.elasticNetParam
             , [1.0, 0.5]
        )
    .build()
)

linReg_ev = ev.RegressionEvaluator(
    predictionCol='prediction'
    , labelCol='price')

# use K-fold cross validation for grid search
# CrossValidator binds all of these together
# default value is k=3
cross_v = tune.CrossValidator(
    estimator=lr_obj
    , estimatorParamMaps=lr_grid
    , evaluator=linReg_ev
)

pipeline = Pipeline(stages=[vectorAssembler, selector])
data_trans = pipeline.fit(dummyModelling_train)

linReg_modelTest = cross_v.fit(
    data_trans.transform(dummyModelling_train)
    .withColumn(
        'label'
        , fn.col('price'))
)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px
```

```
Exception in thread cell_monitor-114:
Traceback (most recent call last):
  File "/emr/notebook-env/lib/python3.7/threading.py", line 926, in
_bootstrap_inner
```

```
    self.run()
  File "/emr/notebook-env/lib/python3.7/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "/emr/notebook-env/lib/python3.7/site-packages/awseditorssparkmonitoringw
idget-1.0-py3.7.egg/awseditorssparkmonitoringwidget/cellmonitor.py", line 178,
in cell_monitor
    job_binned_stages[job_id][stage_id] = all_stages[stage_id]
KeyError: 562
```

[129]:
```
# measure performance of best model
data_trans_test = data_trans.transform(dummyModelling_train)
results = linReg_modelTest.transform(data_trans_test)

print(linReg_ev.evaluate(results, {linReg_ev.metricName: 'rmse'}))
print(linReg_ev.evaluate(results, {linReg_ev.metricName: 'mse'}))
print(linReg_ev.evaluate(results, {linReg_ev.metricName: 'r2'}))
print('Best params, regParam: %s, elasticNetParam: %s'
      %(linReg_modelTest.bestModel._java_obj.getRegParam(),
      linReg_modelTest.bestModel._java_obj.getElasticNetParam()))
```

```
VBox()


FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px


7692.549258120204
59753453.39849906
0.5626247099496577
Best params, regParam: 0.01, elasticNetParam: 1.0
```

[ ]:
```

```