Lecture 23 - More on Data Cleanup

Let's take a close look at the code for data cleanup.

1. Input
2. Discarding nonexistent data. - The first cleanup step.
3. Fixing Columns of data.

4. Fix Strings.

5. Generating one-hot-encoding

6. Some Graphs

7. Saving data - Output

Your lab... Add in some sections - add some graphs.

Part 1 - read data.

```
 1:
 2: # Step 1 — Just do the input and verify the file works.
 3:
 4: import numpy as np
 5: import pandas as pd
 6: import re
 7: import matplotlib.pyplot as plt
 8:
 9: dataset_path = "./train-data.csv"
10:
11: # Specify the columns
12:
13: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
14:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
15:     'Power', 'Seats', 'New_Price', 'Price']
16:
17: # Read the data.
18:
19: raw_dataset = pd.read_csv(dataset_path, names=column_names,
20:     na_values = "?", comment='\t', skiprows=1, sep=",",
21:     skipinitialspace=True)
22:
23: # Print some conformation (the 1st "test")
24:
25: dataset = raw_dataset.copy()
26: print ( dataset.head() )
```

Part 2 - First cleanup.

Get rid of columns you don't need at all. Get rid of missing data.

```python
 1:
 2: # From Before...
 3:
 4: import numpy as np
 5: import pandas as pd
 6: import re
 7: import matplotlib.pyplot as plt
 8:
 9: dataset_path = "./train-data.csv"
10:
11: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
12:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
13:     'Power', 'Seats', 'New_Price', 'Price']
14: raw_dataset = pd.read_csv(dataset_path, names=column_names,
15:     na_values = "?", comment='\t', skiprows=1, sep=",",
16:     skipinitialspace=True)
17:
18: dataset = raw_dataset.copy()
19: print ( dataset.head() )
20:
21: ############################# #############################
22: # New Code...
23: ############################# #############################
24:
25: # Get rid of columns.
26:
27: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
28: print ( dataset.head() )
29:
30: # To see a good description of the dataset
31:
32: print ( dataset.describe() )
33:
34: # Cleaning the data
35: # The dataset contains a few unknown values. Let's find them and drop them.
36:
37: dataset.isna().sum()
38: dataset = dataset.dropna()
39: dataset = dataset.reset_index(drop=True)
40:
41: print ( dataset.head() )
```

Part 3 - Second cleanup.

Get rid of Text in numeric fields.

```
 1:
 2: import numpy as np
 3: import pandas as pd
 4: import re
 5: import matplotlib.pyplot as plt
 6:
 7: dataset_path = "./train-data.csv"
 8:
 9: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
10:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
11:     'Power', 'Seats', 'New_Price', 'Price']
12: raw_dataset = pd.read_csv(dataset_path, names=column_names,
13:     na_values = "?", comment='\t', skiprows=1, sep=",",
14:     skipinitialspace=True)
15:
16: dataset = raw_dataset.copy()
17: print ( dataset.head() )
18:
19: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
20: print ( dataset.head() )
21:
22: print ( dataset.describe() )
23:
24: dataset.isna().sum()
25: dataset = dataset.dropna()
26: dataset = dataset.reset_index(drop=True)
27:
28: print ( dataset.head() )
29:
30: ############################### ################################
31: # Text in Fields
32: ############################### ################################
33:
34: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '', str(val))
35:     for val in dataset['Mileage']], index = dataset.index)
36: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '', str(val))
37:     for val in dataset['Engine']], index = dataset.index)
38: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '', str(val))
39:     for val in dataset['Power']], index = dataset.index)
40:
41: print ( dataset.head() )
```

Part 4 - Fix strings

Convert numbers, get rid of more missing data.

```
 1:
 2: import numpy as np
 3: import pandas as pd
 4: import re
 5: import matplotlib.pyplot as plt
 6:
 7: dataset_path = "./train-data.csv"
 8:
 9: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
10:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
11:     'Power', 'Seats', 'New_Price', 'Price']
12: raw_dataset = pd.read_csv(dataset_path, names=column_names,
13:     na_values = "?", comment='\t', skiprows=1, sep=",",
14:     skipinitialspace=True)
15:
16: dataset = raw_dataset.copy()
17: print ( dataset.head() )
18:
19: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
20: print ( dataset.head() )
21:
22: print ( dataset.describe() )
23:
24: dataset.isna().sum()
25: dataset = dataset.dropna()
26: dataset = dataset.reset_index(drop=True)
27:
28: print ( dataset.head() )
29:
30:
31: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '', str(val))
32:     for val in dataset['Mileage']], index = dataset.index)
33: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '', str(val))
34:     for val in dataset['Engine']], index = dataset.index)
35: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '', str(val))
36:     for val in dataset['Power']], index = dataset.index)
37:
38:
39: ############################### ###############################
40: # Numbers and missing data.
41: ############################### ###############################
42:
43: # The prices are by default in INR Lakhs. So, we have to convert them to USD
44:
45: dataset['Price'] = pd.Series([int(float(val)*1521.22)
46:     for val in dataset['Price']], index = dataset.index)
47:
48: print ( dataset.head() )
49:
50: dataset = dataset.replace(r'^\s*$', np.nan, regex=True)
51: dataset.isna().sum()
52: dataset = dataset.dropna()
53:
54: dataset = dataset.reset_index(drop=True)
55: print ( dataset.head() )
56:
```

Part 5 - Km to Miles (And 1st part of Lab Data Cleanup)

This is the first part where you have some work to do in the data cleanup.

```
 1:
 2: import numpy as np
 3: import pandas as pd
 4: import re
 5: import matplotlib.pyplot as plt
 6:
 7: dataset_path = "./train-data.csv"
 8:
 9: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
10:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
11:     'Power', 'Seats', 'New_Price', 'Price']
12: raw_dataset = pd.read_csv(dataset_path, names=column_names,
13:     na_values = "?", comment='\t', skiprows=1, sep=",",
14:     skipinitialspace=True)
15:
16: dataset = raw_dataset.copy()
17: print ( dataset.head() )
18:
19: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
20: print ( dataset.head() )
21:
22: # To see a good description of the dataset
23:
24: print ( dataset.describe() )
25:
26: # Cleaning the data
27: # The dataset contains a few unknown values. Let's find them and drop them.
28:
29: dataset.isna().sum()
30: dataset = dataset.dropna()
31: dataset = dataset.reset_index(drop=True)
32:
33: print ( dataset.head() )
34:
35:
36: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '',
37:     str(val)) for val in dataset['Mileage']], index = dataset.index)
38: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '',
39:     str(val)) for val in dataset['Engine']], index = dataset.index)
40: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '',
41:     str(val)) for val in dataset['Power']], index = dataset.index)
42:
43: # The prices are by default in INR Lakhs. So, we have to convert them to USD
44:
45: dataset['Price'] = pd.Series([int(float(val)*1521.22) for val in dataset['Price']],
46:         index = dataset.index)
47:
48: print ( dataset.head() )
49:
50: dataset = dataset.replace(r'^\s*$', np.nan, regex=True)
51: dataset.isna().sum()
52: dataset = dataset.dropna()
53:
54: dataset = dataset.reset_index(drop=True)
55: print ( dataset.head() )
56:
57: ########################### ###########################
58: # Part 5 -
59: ########################### ###########################
60:
```

```
61: # Next, we'll convert the strings in the below columns into float values.
62: # Remember that we can only work with numerical values.
63:
64: #  Year  Kilometers_Driven Fuel_Type Transmission Owner_Type  Mileage  Engine   Power  Price
65: # 1. Kilometers_Driven -> Miles_Driven
66: # 2. Milage is in kmpl (Km Per Leter) -> convert to Mi per Gal
67:
68: dataset['Mileage'] = pd.Series([int(float(str(val))*2.3521458)
69:     for val in dataset['Mileage']], index = dataset.index)
70: dataset['Engine'] = pd.Series([float(str(val))
71:     for val in dataset['Engine']], index = dataset.index)
72:
73: ## Lab 09 - TODO - for the column 'Power' in the dataset, convert it to a float
74: ## Lab 09 - TODO - for the column 'Seats' in the dataset, convert it to a float
75: ## Lab 09 - TODO - create the column 'Miles_Driven' from the column
76: ##                   'Kilometers_Driven' by converting to a float and
77: ##                   Multiplying by 0.621371, then convert to an integer so
78: ##                   that we don't have small fractional values.
79: ##
80: ##                   Example of Conversion in just code
81: ##                   x = "23.0"      # A string, with a number in it.
82: ##                   r = int(float(x)*0.621371)
83: ##                       # Convert from string to float,
84: ##                       # Km to Mi, then back to an integer.
85:
86: # Discard the Kilometers_Driven column.
87: dataset = dataset.drop(columns=['Kilometers_Driven'])
88:
89: print ( dataset.head() )
90:
91: # Save data to file so we can open in Excel
92: dataset.to_csv(path_or_buf="new-car-data.csv")
93:
94:
```

Part 6 - One hot encoding

This is the second part where you have some work to do.

```
 1:
 2:
 3: import numpy as np
 4: import pandas as pd
 5: import re
 6: import matplotlib.pyplot as plt
 7:
 8: dataset_path = "./train-data.csv"
 9:
10: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
11:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
12:     'Power', 'Seats', 'New_Price', 'Price']
13: raw_dataset = pd.read_csv(dataset_path, names=column_names,
14:     na_values = "?", comment='\t', skiprows=1, sep=",",
15:     skipinitialspace=True)
16:
17: dataset = raw_dataset.copy()
18: print ( dataset.head() )
19:
20: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
21: print ( dataset.head() )
22:
23: # To see a good description of the dataset
24:
25: print ( dataset.describe() )
26:
27: # Cleaning the data
28: # The dataset contains a few unknown values. Let's find them and drop them.
29:
30: dataset.isna().sum()
31: dataset = dataset.dropna()
32: dataset = dataset.reset_index(drop=True)
33:
34: print ( dataset.head() )
35:
36:
37: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '',
38:     str(val)) for val in dataset['Mileage']], index = dataset.index)
39: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '',
40:     str(val)) for val in dataset['Engine']], index = dataset.index)
41: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '',
42:     str(val)) for val in dataset['Power']], index = dataset.index)
43:
44: # The prices are by default in INR Lakhs. So, we have to convert them to USD
45:
46: dataset['Price'] = pd.Series([int(float(val)*1521.22) for val in dataset['Price']],
47:         index = dataset.index)
48:
49: print ( dataset.head() )
50:
51: dataset = dataset.replace(r'^\s*$', np.nan, regex=True)
52: dataset.isna().sum()
53: dataset = dataset.dropna()
54:
55: dataset = dataset.reset_index(drop=True)
56: print ( dataset.head() )
57:
58: dataset['Mileage'] = pd.Series([int(float(str(val))*2.3521458)
59:     for val in dataset['Mileage']], index = dataset.index)
60: dataset['Engine'] = pd.Series([float(str(val))
```

```
 61:      for val in dataset['Engine']], index = dataset.index)
 62:
 63: ## Lab 09 - TODO - for the column 'Power' in the dataset, convert it to a float
 64: ## Lab 09 - TODO - for the column 'Seats' in the dataset, convert it to a float
 65: ## Lab 09 - TODO - create the column 'Miles_Driven' from the column
 66: ##                     'Kilometers_Driven' by converting to a float and
 67: ##                     Multiplying by 0.621371, then convert to an integer so
 68: ##                     that we don't have small fractional values.
 69: ##
 70: ##                     Example of Conversion in just code
 71: ##                     x = "23.0"      # A string, with a number in it.
 72: ##                     r = int(float(x)*0.621371)
 73: ##                         # Convert from string to float,
 74: ##                         # Km to Mi, then back to an integer.
 75:
 76: dataset = dataset.drop(columns=['Kilometers_Driven'])
 77:
 78: print ( dataset.head() )
 79:
 80: dataset.to_csv(path_or_buf="new-car-data.csv")
 81:
 82:
 83: ############################# #############################
 84: # one hot
 85: ############################# #############################
 86:
 87: ## One-Hot the Fule_Type
 88:
 89: print(dataset['Fuel_Type'].unique())
 90: dataset['Fuel_Type'] = pd.Categorical(dataset['Fuel_Type'])
 91: dfFuel_Type = pd.get_dummies(dataset['Fuel_Type'], prefix = 'Fuel_Type')
 92: print ( dfFuel_Type.head() )
 93:
 94: ## One-Hot the Transmission
 95: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 96: ##                     the Transmission column.
 97: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 98: ##                     the Owner_Type column.
 99:
100: ## Concat it all together
101:
102: ## TODO - when you get the 2 sections above working you will need:
103: #### dataset = pd.concat([dataset, dfFuel_Type, dfTransmission, dfOwner_Type], axis=1)
104:
105: ## instead of just the dfFule_type
106: dataset = pd.concat([dataset, dfFuel_Type], axis=1)
107:
108: dataset = dataset.drop(columns=['Owner_Type', 'Transmission', 'Fuel_Type'])
109: print ( dataset.head() )
110:
111:
112: # Save the data again - take a look at it.
113:
114: dataset.to_csv(path_or_buf="new-car-data2.csv")
```

Part 7 - Plot some stuff.

This is the third part where you have some work to do in the data cleanup.

```
 1:
 2: import numpy as np
 3: import pandas as pd
 4: import re
 5: import matplotlib.pyplot as plt
 6:
 7: dataset_path = "./train-data.csv"
 8:
 9: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
10:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
11:     'Power', 'Seats', 'New_Price', 'Price']
12: raw_dataset = pd.read_csv(dataset_path, names=column_names,
13:     na_values = "?", comment='\t', skiprows=1, sep=",",
14:     skipinitialspace=True)
15:
16: dataset = raw_dataset.copy()
17: print ( dataset.head() )
18:
19: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
20: print ( dataset.head() )
21:
22: # To see a good description of the dataset
23:
24: print ( dataset.describe() )
25:
26: # Cleaning the data
27: # The dataset contains a few unknown values. Let's find them and drop them.
28:
29: dataset.isna().sum()
30: dataset = dataset.dropna()
31: dataset = dataset.reset_index(drop=True)
32:
33: print ( dataset.head() )
34:
35:
36: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '',
37:     str(val)) for val in dataset['Mileage']], index = dataset.index)
38: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '',
39:     str(val)) for val in dataset['Engine']], index = dataset.index)
40: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '',
41:     str(val)) for val in dataset['Power']], index = dataset.index)
42:
43: # The prices are by default in INR Lakhs. So, we have to convert them to USD
44:
45: dataset['Price'] = pd.Series([int(float(val)*1521.22) for val in dataset['Price']],
46:         index = dataset.index)
47:
48: print ( dataset.head() )
49:
50: dataset = dataset.replace(r'^\s*$', np.nan, regex=True)
51: dataset.isna().sum()
52: dataset = dataset.dropna()
53:
54: dataset = dataset.reset_index(drop=True)
55: print ( dataset.head() )
56:
57: dataset['Mileage'] = pd.Series([int(float(str(val))*2.3521458)
58:     for val in dataset['Mileage']], index = dataset.index)
59: dataset['Engine'] = pd.Series([float(str(val))
60:     for val in dataset['Engine']], index = dataset.index)
```

```
 61:
 62: ## Lab 09 - TODO - for the column 'Power' in the dataset, convert it to a float
 63: ## Lab 09 - TODO - for the column 'Seats' in the dataset, convert it to a float
 64: ## Lab 09 - TODO - create the column 'Miles_Driven' from the column
 65: ##                     'Kilometers_Driven' by converting to a float and
 66: ##                     Multiplying by 0.621371, then convert to an integer so
 67: ##                     that we don't have small fractional values.
 68: ##
 69: ##                     Example of Conversion in just code
 70: ##                     x = "23.0"       # A string, with a number in it.
 71: ##                     r = int(float(x)*0.621371)
 72: ##                         # Convert from string to float,
 73: ##                         # Km to Mi, then back to an integer.
 74:
 75: dataset = dataset.drop(columns=['Kilometers_Driven'])
 76:
 77: print ( dataset.head() )
 78:
 79: dataset.to_csv(path_or_buf="new-car-data.csv")
 80:
 81:
 82:
 83: ## One-Hot the Fule_Type
 84:
 85: print(dataset['Fuel_Type'].unique())
 86: dataset['Fuel_Type'] = pd.Categorical(dataset['Fuel_Type'])
 87: dfFuel_Type = pd.get_dummies(dataset['Fuel_Type'], prefix = 'Fuel_Type')
 88: print ( dfFuel_Type.head() )
 89:
 90: ## One-Hot the Transmission
 91: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 92: ##                     the Transmission column.
 93: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 94: ##                     the Owner_Type column.
 95:
 96: ## Concat it all together
 97:
 98: ## TODO - when you get the 2 sections above working you will need:
 99: #### dataset = pd.concat([dataset, dfFuel_Type, dfTransmission, dfOwner_Type], axis=1)
100:
101: ## instead of just the dfFule_type
102: dataset = pd.concat([dataset, dfFuel_Type], axis=1)
103:
104: dataset = dataset.drop(columns=['Owner_Type', 'Transmission', 'Fuel_Type'])
105: print ( dataset.head() )
106:
107:
108: # Save the data again - take a look at it.
109:
110: dataset.to_csv(path_or_buf="new-car-data2.csv")
111:
112: ########################### ############################
113: # Plot some stuff.
114: ########################### ############################
115:
116:
117: dataset.plot(kind='scatter',x='Price',y='Year',color='blue')
118: plt.show()
119:
120: ## Lab - 09 - TODO - Plot Price v.s. Miles_Driven
121: ## Lab - 09 - TODO - Plot Price v.s. Power
122: ## Lab - 09 - TODO - Plot Price v.s. Milage
123: ## Lab - 09 - TODO - Plot Price v.s. Seats
```

Your Code to start with for the lab.

This code is in the lab pdf also - with a link to download your file.

```
 1:
 2: import numpy as np
 3: import pandas as pd
 4: import re
 5: import matplotlib.pyplot as plt
 6:
 7: dataset_path = "./train-data.csv"
 8:
 9: column_names = ['Ind', 'Name', 'Location', 'Year', 'Kilometers_Driven',
10:     'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine',
11:     'Power', 'Seats', 'New_Price', 'Price']
12: raw_dataset = pd.read_csv(dataset_path, names=column_names,
13:     na_values = "?", comment='\t', skiprows=1, sep=",",
14:     skipinitialspace=True)
15:
16: dataset = raw_dataset.copy()
17: print ( dataset.head() )
18:
19: dataset = dataset.drop(columns=['Ind', 'Name', 'Location', 'New_Price'])
20: print ( dataset.head() )
21:
22: # To see a good description of the dataset
23:
24: print ( dataset.describe() )
25:
26: # Cleaning the data
27: # The dataset contains a few unknown values. Let's find them and drop them.
28:
29: dataset.isna().sum()
30: dataset = dataset.dropna()
31: dataset = dataset.reset_index(drop=True)
32:
33: print ( dataset.head() )
34:
35:
36: dataset['Mileage'] = pd.Series([re.sub('[^.0-9]', '',
37:     str(val)) for val in dataset['Mileage']], index = dataset.index)
38: dataset['Engine'] = pd.Series([re.sub('[^.0-9]', '',
39:     str(val)) for val in dataset['Engine']], index = dataset.index)
40: dataset['Power'] = pd.Series([re.sub('[^.0-9]', '',
41:     str(val)) for val in dataset['Power']], index = dataset.index)
42:
43: # The prices are by default in INR Lakhs. So, we have to convert them to USD
44:
45: dataset['Price'] = pd.Series([int(float(val)*1521.22) for val in dataset['Price']],
46:         index = dataset.index)
47:
48: print ( dataset.head() )
49:
50: dataset = dataset.replace(r'^\s*$', np.nan, regex=True)
51: dataset.isna().sum()
52: dataset = dataset.dropna()
53:
54: dataset = dataset.reset_index(drop=True)
55: print ( dataset.head() )
56:
57: dataset['Mileage'] = pd.Series([int(float(str(val))*2.3521458)
58:     for val in dataset['Mileage']], index = dataset.index)
59: dataset['Engine'] = pd.Series([float(str(val))
60:     for val in dataset['Engine']], index = dataset.index)
```

```
 61:
 62: dataset['Power'] = pd.Series([float(str(val))
 63:      for val in dataset['Power']], index = dataset.index)
 64: dataset['Seats'] = pd.Series([float(str(val))
 65:      for val in dataset['Seats']], index = dataset.index)
 66: ## Lab 09 - TODO - for the column 'Power' in the dataset, convert it to a float
 67: ## Lab 09 - TODO - for the column 'Seats' in the dataset, convert it to a float
 68: ## Lab 09 - TODO - create the column 'Miles_Driven' from the column
 69: ##                  'Kilometers_Driven' by converting to a float and
 70: ##                  Multiplying by 0.621371, then convert to an integer so
 71: ##                  that we don't have small fractional values.
 72: ##
 73: ##                  Example of Conversion in just code
 74: ##                  x = "23.0"       # A string, with a number in it.
 75: ##                  r = int(float(x)*0.621371)
 76: ##                      # Convert from string to float,
 77: ##                      # Km to Mi, then back to an integer.
 78:
 79: dataset = dataset.drop(columns=['Kilometers_Driven'])
 80:
 81: print ( dataset.head() )
 82:
 83: dataset.to_csv(path_or_buf="new-car-data.csv")
 84:
 85:
 86:
 87: ## One-Hot the Fule_Type
 88:
 89: print(dataset['Fuel_Type'].unique())
 90: dataset['Fuel_Type'] = pd.Categorical(dataset['Fuel_Type'])
 91: dfFuel_Type = pd.get_dummies(dataset['Fuel_Type'], prefix = 'Fuel_Type')
 92: print ( dfFuel_Type.head() )
 93:
 94: ## One-Hot the Transmission
 95: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 96: ##                  the Transmission column.
 97: ## Lab -09 - TODO - do a similar one-hot encoding for the values in
 98: ##                  the Owner_Type column.
 99:
100: ## Concat it all together
101:
102: ## TODO - when you get the 2 sections above working you will need:
103: #### dataset = pd.concat([dataset, dfFuel_Type, dfTransmission, dfOwner_Type], axis=1)
104:
105: ## instead of just the dfFule_type
106: dataset = pd.concat([dataset, dfFuel_Type], axis=1)
107:
108: dataset = dataset.drop(columns=['Owner_Type', 'Transmission', 'Fuel_Type'])
109: print ( dataset.head() )
110:
111:
112: # Save the data again - take a look at it.
113:
114: dataset.to_csv(path_or_buf="new-car-data2.csv")
115:
116: ############################# #############################
117: # Plot some stuff.
118: ############################# #############################
119:
120:
121: dataset.plot(kind='scatter',x='Price',y='Year',color='blue')
122: plt.show()
123:
124: ## Lab - 09 - TODO - Plot Price v.s. Miles_Driven
125: ## Lab - 09 - TODO - Plot Price v.s. Power
126: ## Lab - 09 - TODO - Plot Price v.s. Milage
```

```
127: ## Lab — 09 — TODO — Plot Price v.s. Seats
```