Week 8 Quiz

Bryan Gibson - brg2130

In [1]:

```
# import numpy as np and pandas as pd
import numpy as np
import pandas as pd
```

In [2]:

```
# Import data from data/week8_flowershop_data.csv into dataframe df
# Print df.info() to see the number of rows, column names and column datatypes and amount of missing data.
df = pd.read_csv('../data/week8_flowershop_data.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1002 entries, 0 to 1001
Data columns (total 6 columns):
PurchaseID         1002 non-null int64
lastname           1002 non-null object
purchase_date      1002 non-null object
stars              1002 non-null int64
price              980 non-null float64
favorite_flower    823 non-null object
dtypes: float64(1), int64(2), object(3)
memory usage: 47.1+ KB
```

In [3]:

```
# If we run df.duplicated() we get a vector of booleans that indicate duplicated rows.
# Print the sum over df.duplicated() to get the number of duplicates.
sum(df.duplicated())
```

Out[3]:

```
2
```

In [4]:

```
# Use drop_duplicates() to drop the duplicated rows and store back in to df.
# Check the entire row (subset=None) and keep the first duplicate (keep='first')
# Print df.shape to confirm rows were dropped.
df = df.drop_duplicates()
df.shape
```

Out[4]:

```
(1000, 6)
```

In [5]:

```
# From the info above, we see there are missing values in price.
# Before we fill this column, create a new column 'price_missing' in df.
# This column should contain integers, 1 for missing, 0 for not missing.
# Use .isna() and .astype(int) to create the 'price_missing' column.
df['price_missing'] = df.price.isna().astype(int)
```

In [6]:

```
# Now fill the missing values in df.price with the mean of the price column.
# Use .fillna() and .mean()
# Be sure to either use inplace or store back into the existing price column.
df.price = df.price.fillna(df.price.mean())
```

In [7]:

```
# Standardize the price column using the sklearn StandardScaler

# Import StandardScaler from sklearn.
# Use either fit() and transform() or fit_transform() on the price column only.
# NOTE: fit_transform requires a 2D matrix. Use df[['price']] to pass a dataframe instead of a series.
# Store the transformed values into a new column 'price_scaled' in df.
# Call describe on price and price_scaled columns and note the means and standard deviations.

from sklearn.preprocessing import StandardScaler

df['price_scaled'] = StandardScaler().fit_transform(df[['price']])
df[['price','price_scaled']].describe()
```

Out[7]:

```
             price  price_scaled
count  1000.000000  1.000000e+03
mean     73.403241  1.412204e-15
std      11.085129  1.000500e+00
min      57.621566 -1.424392e+00
25%      68.274678 -4.628840e-01
50%      70.197617 -2.893271e-01
75%      88.588789  1.370588e+00
max      92.996317  1.768394e+00
```

In [8]:

```
# There are also missing values in favorite flower.
# Since 'favorite_flower' is categorical, let's treat missing as another category.
# Fill the empty values in favorite_flower with the string 'MISSING'.
# Be sure to either use inplace or store back into the existing favorite_flower column.
df.favorite_flower = df.favorite_flower.fillna('MISSING')
```

In [9]:
```
# Confirm we have no missing data.
# Use .isna().sum().sum() to print the number of missing values in th
e dataframe.
df.isna().sum().sum()
```

Out[9]:
```
0
```

In [10]:
```
# Transform the categorical feature favorite_flower using pd.get_dumm
ies().
# Use prefix='favorite_flower' to add a prefix to each column name.
# pd.get_dummies creates a new dataframe, so save the result of pd.ge
t_dummies to df_flower.
# Print out the first 3 rows of df_flower to see the result.
df_flower = pd.get_dummies(df.favorite_flower, prefix='favorite_flowe
r')
df_flower.iloc[:3]
```

Out[10]:

	favorite_flower_MISSING	favorite_flower_carnation	favorite_flower_daffodil	favorite_flower_dai
0	0	0	0	
1	1	0	0	
4	0	1	0	

In [11]:
```
# OPTIONAL!

# We now need to combine our original dataframe df and this new df_fl
ower
# We have not discussed how to do this yet in class, but if you're in
terested, feel free to try.
# We can use the .join() command here as both dataframes share the sa
me index.
# For info on join see: https://pandas.pydata.org/pandas-docs/stable/
reference/api/pandas.DataFrame.join.html
df = df.join(df_flower)
df.head()
```

Out[11]:

	PurchaseID	lastname	purchase_date	stars	price	favorite_flower	price_missing	p
0	101	PERKINS	2017-04-08	5	69.599886	iris	0	
1	102	ROBINSON	2017-01-01	5	87.983904	MISSING	0	
4	103	WILLIAMSON	2017-03-20	4	69.339138	carnation	0	
5	104	ROBINSON	2017-04-12	5	68.140616	lilac	0	
6	105	RHODES	2017-03-24	1	72.179522	carnation	0	

5 rows × 21 columns