Assignment 1
Melissa LaHoud

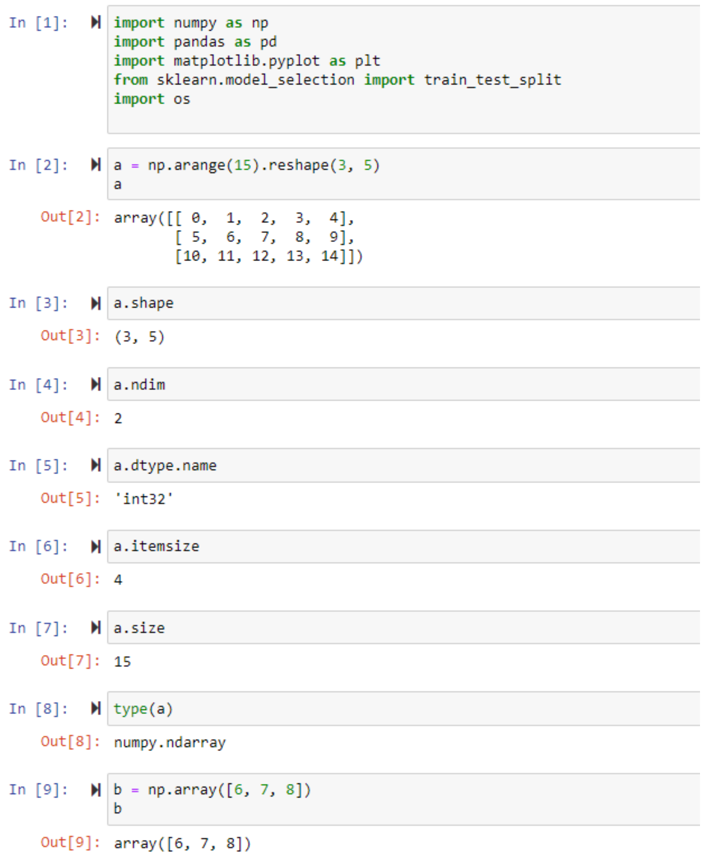
Part 1 - Tools Readiness

Install Python 3.7 (or later) and PyCharm
Add the following libraries: *Numpy*, *Pandas*, *Matplotlib*, and *Scikit-Learn*

Create simple Jupyter notebooks, in which you import the four packages and write minimal Python scripts demonstrating that all libraries have been installed correctly. You may copy examples from the quickstart tutorials for each one of these libraries:

- https://docs.scipy.org/doc/numpy/user/quickstart.html
- https://pandas.pydata.org/pandas-docs/stable/getting_started/
- https://matplotlib.org/users/pyplot_tutorial.html
- https://elitedatascience.com/python-machine-learning-tutorial-scikit-learn

NOTE: *Occasionally, links to packages and manuals change. If you get a broken link, navigate one level up on the respective sites, where the package is located, and look for the link to the reference manual.*

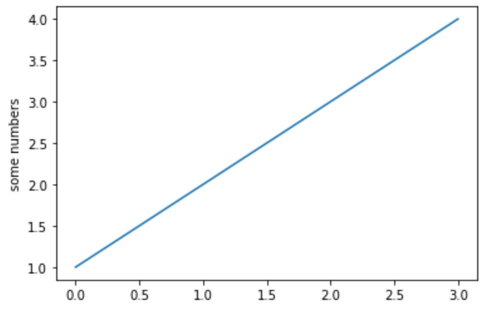


```
In [1]:  import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from sklearn.model_selection import train_test_split
         import os

In [2]:  a = np.arange(15).reshape(3, 5)
         a

Out[2]:  array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14]])

In [3]:  a.shape

Out[3]:  (3, 5)

In [4]:  a.ndim

Out[4]:  2

In [5]:  a.dtype.name

Out[5]:  'int32'

In [6]:  a.itemsize

Out[6]:  4

In [7]:  a.size

Out[7]:  15

In [8]:  type(a)

Out[8]:  numpy.ndarray

In [9]:  b = np.array([6, 7, 8])
         b

Out[9]:  array([6, 7, 8])
```

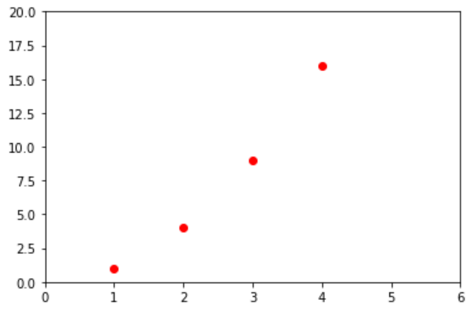
```
[11]:  print(a)

       [[ 0  1  2  3  4]
        [ 5  6  7  8  9]
        [10 11 12 13 14]]
```

```
[12]:  A = np.array( [[1,1],
                      [0,1]] )
       B = np.array( [[2,0],
                      [3,4]] )

       A * B

Out[12]: array([[2, 0],
                [0, 4]])
```

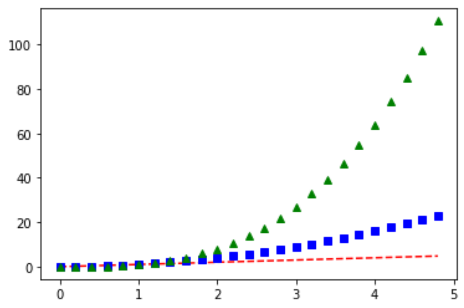
```
[13]:  plt.plot([1,2,3,4])
       plt.ylabel('some numbers')
       plt.show()
```



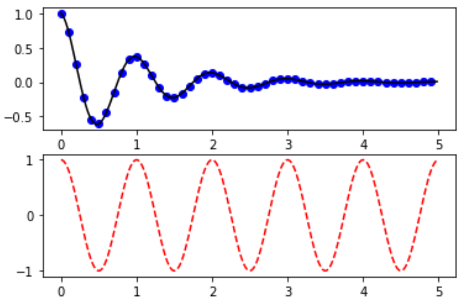
```
[14]:  plt.plot([1,2,3,4], [1,4,9,16], 'ro')
       plt.axis([0, 6, 0, 20])
       plt.show()
```



```
[15]:  # evenly sampled time at 200ms intervals
       t = np.arange(0., 5., 0.2)

       # red dashes, blue squares and green triangles
       plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
       plt.show()
```



```
]:  def f(t):
        return np.exp(-t) * np.cos(2*np.pi*t)

    t1 = np.arange(0.0, 5.0, 0.1)
    t2 = np.arange(0.0, 5.0, 0.02)

    plt.figure(1)
    plt.subplot(211)
    plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

    plt.subplot(212)
    plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
    plt.show()
```



```
In [17]:  os.chdir("C:\\Users\\melis\\Documents\\DSC-530 Predictive Modeling")

           Adult = pd.read_csv('adult.csv')
           Adult.head()
```

Out[17]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K

```
In [18]:  y = Adult.income
           X_train, X_test, y_train, y_test = train_test_split(Adult, y, test_size=0.5)
           print(X_train.shape, y_train.shape)

           (24421, 15) (24421,)
```

Part 2 - Review Predictive Models and Python Proficiency

Consider the task of calculating the appreciation of a real estate property over time. Using concepts from previous courses (e.g., linear regression, predictive modeling), create a model to predict the future value of the property using at least three different input variables. You have the freedom to decide what these variables are. Show the following:

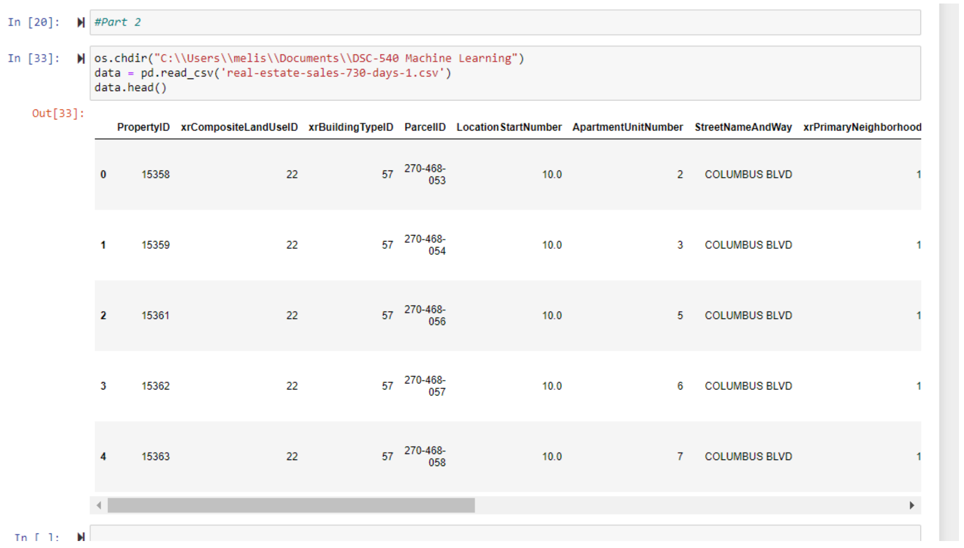
- The mathematical model using formal mathematical notation

  Based on my code below, the model is the following: Y = -184800 + 5292*(xrPrimaryNeighborhoodID) + 30.7261*(TotalFinishedArea) + 31290*(LivingUnits)

- An explanation of all variables used

  I used Primary Neighborhood ID, Total Finished Area and Living Units to predict sales price. I kept these variables because when we look at the summary, all variables have a p-value less then .05 meaning they are significant. If we are to look at Total Finished area, when total finished area increases, so does the price of the house. Specifically, when the total finished area will increase by 1, the sales price increases by 30.7261. When Primary Neighborhood ID increases by 1, the Sales price increases by 5292 and when living units increases by one, the sales price increases by 31,290.

- An implementation of the model in Python (as a Jupyter notebook), including relevant visual output (e.g., graphs)



```
In [20]:  #Part 2

In [33]:  os.chdir("C:\\Users\\melis\\Documents\\DSC-540 Machine Learning")
          data = pd.read_csv('real-estate-sales-730-days-1.csv')
          data.head()

Out[33]:
```

	PropertyID	xrCompositeLandUseID	xrBuildingTypeID	ParcelID	LocationStartNumber	ApartmentUnitNumber	StreetNameAndWay	xrPrimaryNeighborhood
0	15358	22	57	270-468-053	10.0	2	COLUMBUS BLVD	1
1	15359	22	57	270-468-054	10.0	3	COLUMBUS BLVD	1
2	15361	22	57	270-468-056	10.0	5	COLUMBUS BLVD	1
3	15362	22	57	270-468-057	10.0	6	COLUMBUS BLVD	1
4	15363	22	57	270-468-058	10.0	7	COLUMBUS BLVD	1

```
In [ ]:
```

n [21]:
```python
import statsmodels.api as sm
```

n [34]:
```python
X = pd.DataFrame(data[['xrPrimaryNeighborhoodID','TotalFinishedArea','LivingUnits']])
Y = pd.DataFrame(data[['SalePrice']])

X = sm.add_constant(X)

model = sm.OLS(Y, X).fit()

model.summary()
```

Out[34]:

OLS Regression Results

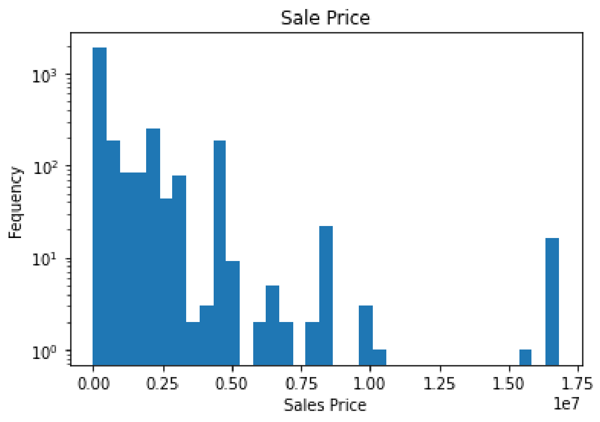
Dep. Variable:	SalePrice	**R-squared:**	0.245
Model:	OLS	**Adj. R-squared:**	0.244
Method:	Least Squares	**F-statistic:**	311.3
Date:	Tue, 23 Mar 2021	**Prob (F-statistic):**	4.36e-175
Time:	17:26:08	**Log-Likelihood:**	-45498.
No. Observations:	2888	**AIC:**	9.100e+04
Df Residuals:	2884	**BIC:**	9.103e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.848e+05	5.26e+04	-3.512	0.000	-2.88e+05	-8.16e+04
xrPrimaryNeighborhoodID	5292.3765	195.006	27.140	0.000	4910.012	5674.741
TotalFinishedArea	30.7261	2.520	12.193	0.000	25.785	35.667
LivingUnits	3.129e+04	4980.308	6.282	0.000	2.15e+04	4.11e+04

Omnibus:	3093.994	**Durbin-Watson:**	0.158
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	238195.100
Skew:	5.348	**Prob(JB):**	0.00
Kurtosis:	46.186	**Cond. No.**	2.46e+04

37]:
```python
plt.hist(Y, bins=35, log=True)
plt.title("Sale Price")
plt.xlabel('Sales Price')
plt.ylabel('Fequency')
plt.show
```

Out[37]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[38]:

```python
from sklearn.tree import DecisionTreeClassifier

y = data[['SalePrice']]

x = pd.concat((data[['xrPrimaryNeighborhoodID']], data[['TotalFinishedArea']], data[['xrBuildingTypeID']]), axis=1)

x_names = ["xrPrimaryNeighborhoodID", "TotalFinishedArea","xrBuildingTypeID"]

y_names = ["T","F"]

C50_01 = DecisionTreeClassifier(criterion = "entropy", max_leaf_nodes=10).fit(x,y)

data['C1'] = C50_01.predict(x)

C51 = pd.crosstab(data['SalePrice'], data['C1'])
C51
```

Out[38]:

C1	145000	150000	200000	250000	950000	1030000	1975000	4550000	8500000	16800000
SalePrice										
1000	0	0	0	0	0	0	0	0	0	2
2500	0	0	1	0	0	0	0	0	0	0
3000	0	2	0	0	0	0	0	0	0	0
4800	0	1	0	0	0	0	0	0	0	0
5000	0	0	1	1	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...
8500000	0	0	1	0	0	0	0	0	21	0
9675000	0	0	0	0	0	0	0	0	0	3
10300000	0	0	0	0	0	0	0	0	0	1
15750000	0	0	0	0	0	0	0	0	1	0
16800000	1	0	1	0	0	0	0	0	0	14

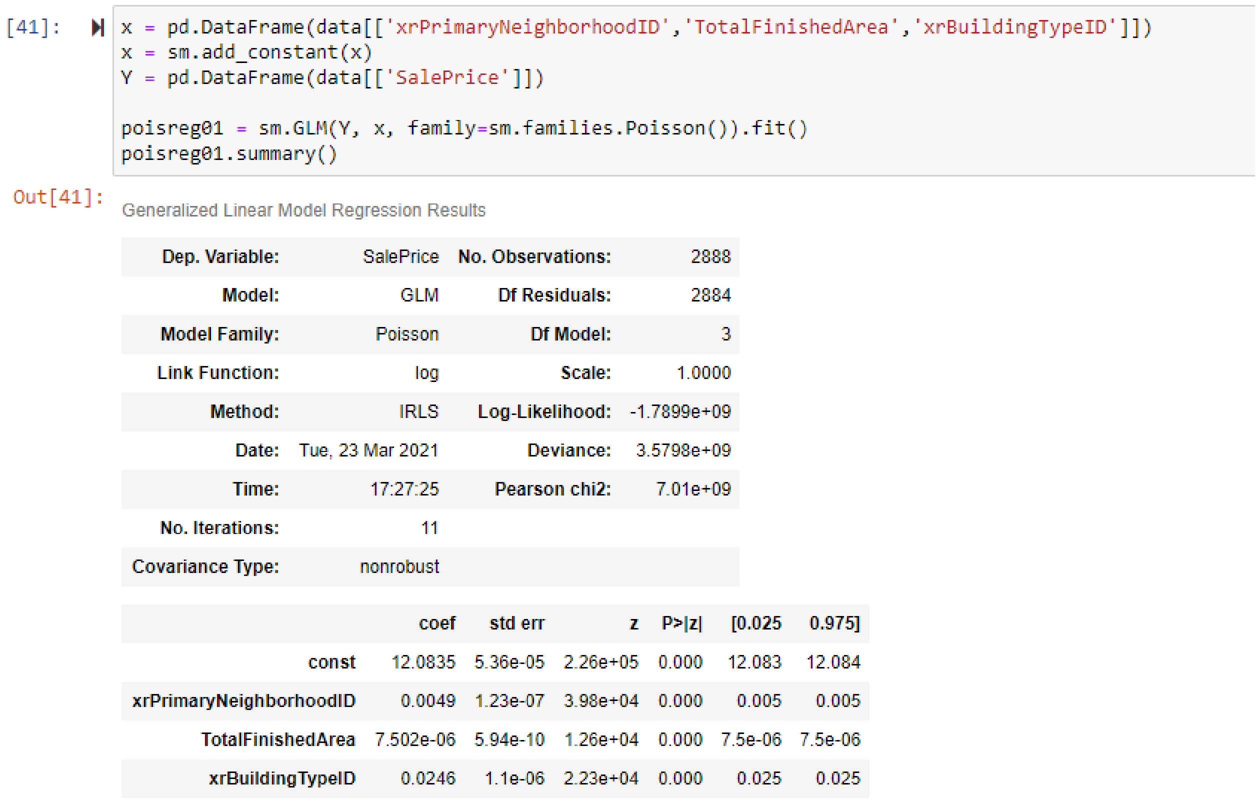
In [40]:

```python
import statsmodels.api as sm
from scipy import stats
import statsmodels.tools.tools as stattools

pred = pd.DataFrame(round(model2.predict(X)))
pred.columns = ['PredictedValues']

Yhat = pd.crosstab(data['SalePrice'], pred['PredictedValues'])
Yhat
```

Out[40]:

PredictedValues	-336196.0	-321642.0	-321579.0	-317815.0	-314929.0	-313895.0	-310680.0	-310588.0	-309408.0	-309283.0	...	6021524.0	7914387.0	845
SalePrice														
1000	0	0	0	0	0	0	0	0	0	0	...	0	0	
2500	0	0	0	0	0	0	0	0	0	0	...	0	0	
3000	0	0	0	0	0	0	0	0	0	0	...	0	0	
4800	0	0	0	0	0	0	0	0	0	0	...	0	0	
5000	0	0	0	0	0	0	0	0	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8500000	0	0	0	0	0	0	0	0	0	0	...	0	0	
9675000	0	0	0	0	0	0	0	0	0	0	...	0	0	
10300000	0	0	0	0	0	0	0	0	0	0	...	0	0	
15750000	0	0	0	0	0	0	0	0	0	0	...	0	0	
16800000	0	0	0	0	0	0	0	0	0	0	...	0	0	

740 rows × 1876 columns



[41]:

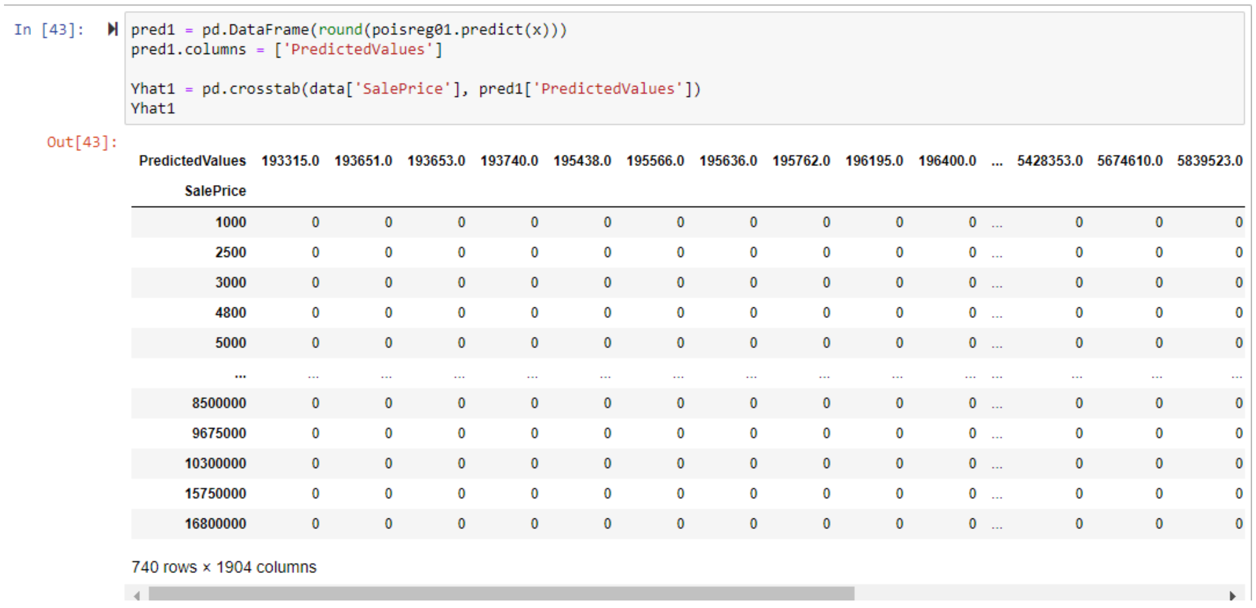
```
x = pd.DataFrame(data[['xrPrimaryNeighborhoodID','TotalFinishedArea','xrBuildingTypeID']])
x = sm.add_constant(x)
Y = pd.DataFrame(data[['SalePrice']])

poisreg01 = sm.GLM(Y, x, family=sm.families.Poisson()).fit()
poisreg01.summary()
```

Out[41]:

Generalized Linear Model Regression Results

Dep. Variable:	SalePrice	**No. Observations:**	2888
Model:	GLM	**Df Residuals:**	2884
Model Family:	Poisson	**Df Model:**	3
Link Function:	log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1.7899e+09
Date:	Tue, 23 Mar 2021	**Deviance:**	3.5798e+09
Time:	17:27:25	**Pearson chi2:**	7.01e+09
No. Iterations:	11		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	12.0835	5.36e-05	2.26e+05	0.000	12.083	12.084
xrPrimaryNeighborhoodID	0.0049	1.23e-07	3.98e+04	0.000	0.005	0.005
TotalFinishedArea	7.502e-06	5.94e-10	1.26e+04	0.000	7.5e-06	7.5e-06
xrBuildingTypeID	0.0246	1.1e-06	2.23e+04	0.000	0.025	0.025



In [43]:

```
pred1 = pd.DataFrame(round(poisreg01.predict(x)))
pred1.columns = ['PredictedValues']

Yhat1 = pd.crosstab(data['SalePrice'], pred1['PredictedValues'])
Yhat1
```

Out[43]:

PredictedValues	**193315.0**	**193651.0**	**193653.0**	**193740.0**	**195438.0**	**195566.0**	**195636.0**	**195762.0**	**196195.0**	**196400.0**	**...**	**5428353.0**	**5674610.0**	**5839523.0**
SalePrice														
1000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
2500	0	0	0	0	0	0	0	0	0	0	...	0	0	0
3000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
4800	0	0	0	0	0	0	0	0	0	0	...	0	0	0
5000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8500000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
9675000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
10300000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
15750000	0	0	0	0	0	0	0	0	0	0	...	0	0	0
16800000	0	0	0	0	0	0	0	0	0	0	...	0	0	0

740 rows × 1904 columns

I tried multiple ways of going about predicting sales price such as logistic regression model, poisson regression model and decision trees. Each of the code is above and have different mathematical notation.

What factors influence the quality of predictions? Discuss these factors and measure the change in outcome when these factors are modified. Estimate the error in your model, both mathematically and in code.

One factor here that I noticed was the multicollinearity which means that's predictors are correlated with other predictors. It undermines the statistical significance of an independent variables. We can look at the std. error on the summary tab for the first model.

Create a private GitHub or Bitbucket repository (you may already have one from a previous class). Then, create a directory for this class and sub-directories for each topic assignments.

Create a README.md file in which you list all the files that make up your submission and the necessary instructions to run the code.

Upload the Python Jupyter notebooks to your repository and submit the link to LoudCloud.

Part 3- Technical Report
Refer to the readings in this topic (textbook and the article "The Seven Tools of Causal Inference, with Reflections on Machine Learning," which describe:

- Four forms of learning (supervised, unsupervised, reinforcement, evolutionary)

  According to Gopal, M. (2019), "Supervised learning is a priori known information in the form of 'direct' training examples consisting of observed values of system states (input vectors): x(1), …, x(N), and the response (output) to each state: y(1), …, y(N).The 'supervisor' has, thus, provided the following data:"

$$\mathcal{D} = \{s^{(i)}, y^{(i)}\};\ i = 1, \ldots, N$$
$$s^{(i)} = \mathbf{x}^{(i)} :\ \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$$

  (Gopal, M., 2019)

  Unsupervised learning "is when output y(i) is not available in training data. In this type of problem, we are given a set of feature vectors x(i), and the goal is to unravel the underlying similarities." (Gopal, M., 2019)

  "Reinforcement learning is that if an action is followed by a satisfactory state of affairs, or by a improved state of affairs, the inclination to produces that action becomes stronger, i.e. reinforced. Another way to put it is that a system that via interaction with its environment enhances its performance by obtaining feedback in the form of a reward – a reinforcement signal, that indicated of the suitability of the response." (Gopal, M., 2019)

  Evolutionary: "It derives ideas from evolutionary biology to develop search and optimization methods that help solve complicated problems. Evolutionary biology essentially states that a population of individuals possessing the ability to reproduce and exposed to genetic variation followed by selection, gives rise to new populations, which are fitter to their environment. Computational abstraction of these processes gave rise to the so-called evolutionary algorithms." (Gopal, M., 2019)

- Four learning tasks (classification, regression, learning association, clustering)

  Classification is a type of supervised learning that is a pattern recognition. “The goal of this task is to predict the output values for the new inputs based on training from examples of each class.” (Gopal, M., 2019)

  Regression is a type of supervised learning that is a numeric prediction. This task consists of fitting a function to the input-output data with the goal of predicting output values for new inputs.” (Gopal, M., 2019)

  Clustering is part of unsupervised learning. Cluster analysis creates clusters of similar data points. “It is important to see the issue of the selection of an algorithm scheme that will group the vector based on the accepted similarity measure.” (Gopal, M., 2019)

  Association is part of the unsupervised leanings. “Association analysis uses unsupervised learning to discover patterns in the data where no target is specified earlier. It is up to human interpretation to make sense of the patterns.” (Gopal, M., 2019)

- Seven tools of causal inference

  Tool 1: Encoding causal assumptions: Transparency and testability. “The task of encoding assumptions in

  a compact and usable form is not a trivial matter once an analyst takes seriously the requirement of transparency and testability. d Transparency enables analysts to discern whether the assumptions encoded are plausible (on scientific grounds) or whether additional assumptions are warranted.

  Testability permits us (whether analyst or machine) to determine whether the assumptions encoded are compatible with the available data and, if not, identify those that need repair.” (Pearl, J., 2019)

  Tool 2: Do-calculus and the control of confounding. “Confounding, or the presence of unobserved causes of two or more variables, long considered the major obstacle to drawing causal inference from data, has been demystified and “deconfounded” through a graphical criterion called “backdoor.” In particular, the task of selecting an appropriate set of covariates to control for confounding has been reduced to a simple “roadblocks” puzzle manageable through a simple algorithm.” (Pearl, J., 2019)

  Tool 3: The algorithmitization of counterfactuals. “Counterfactual analysis deals with behavior of specific individuals identified by a distinct set of characteristics.” (Pearl, J., 2019)

  Tool 4: Mediation analysis and the assessment of direct and indirect effects. “Mediation analysis concerns the mechanisms that transmit changes from a cause to its effects. The logic of counterfactuals and their graphical representation have spawned algorithms for estimating direct and indirect effects from data or experiments.” (Pearl, J., 2019)

  Tool 5: Adaptability, external validity, and sample selection bias. “The validity of every experimental study is challenged by disparities between the experimental and the intended implementational setups.” (Pearl, J., 2019)

Tool 6: Recovering from missing data. “Problems due to missing data plague every branch of experimental science. Respondents do not answer every item on a questionnaire, sensors malfunction as weather conditions worsen, and patients often drop from a clinical study for unknown reasons.” (Pearl, J., 2019)

Tool 7: Causal discovery. “The d-separation criterion described earlier enables machines to detect and enumerate the testable implications of a given causal model. This opens the possibility of inferring, with mild assumptions, the set of models that are compatible with the data and to represent this set compactly.” (Pearl, J., 2019)

Examine the article *“Using Machine Learning to Translate Applicant Work History into Predictors of Performance and Turnover.”* Write a two-page technical report covering the following:

1. Characterize the article in terms of the forms of learning, learning tasks, and causal inference it reports.
2. Defend your characterization by mapping the concepts onto the specific details mentioned or inferred in the article.
3. Use the GCU digital library to find work describing a form of learning or learning task that is not covered by the categories listed in Part 1. Use your findings to expand your characterization in (2).

Using Machine Learning to Translate Applicant Work History Predictors of Performance and Turnover comparison to Leaning techniques, learning tasks and causal inferences

In the article, the most used learning techniques is supervised. According to Gopal, M. (2019), "Supervised learning is a priori known information in the form of 'direct' training examples consisting of observed values of system states (input vectors): x(1), …, x(N), and the response (output) to each state: y(1), …, y(N)."

They first use supervised learning to evaluate reasons applicants describe for leaving pervious jobs. They put these reasons into 4 separate groupings. They trained the algorithm and it "learned" to evaluate e the probabilities that different words and word combinations predict belonging to each category. There were three hypotheses that that used supervised learning as well. Sajjadiani, S., etc. (2019) states the following hypotheses:

> "Hypothesis 3: Applicant attributions of previous turnover as involuntary, as assessed via supervised machine learning, is (a) negatively associated with performance, and (b) positively associated with voluntary turnover hazard.
>
> Hypothesis 4: Applicant attributions of previous turnover to avoiding bad jobs, as assessed via supervised machine learning, is (a) negatively associated with performance, and (b) positively associated with voluntary turnover hazard.
>
> Hypothesis 5: Applicant attributions of previous turnover to approaching better jobs, as assessed via supervised machine learning, is (a) positively associated with performance, and (b) negatively associated with voluntary turnover hazard."

For the first step, they used supervised machine learning techniques to develop an algorithm that classified self-reported job titles and job descriptions into an O*NET standardized occupation code. They used a learning task called classification which is a part of supervised learning. "Classification is recommended for theory-driven studies and for cases in which a strong, external, ground-truth dataset exists on which to train the algorithm. Such classifiers were developed by learning the characteristics of different classes from a training sample of pre classified documents." (Sajjadiani, S., etc., 2019).

Sajjadiani, S., etc. (2019) states, “To check the accuracy of this classification, the RA categorized a random sample of 350 classified turnover attributions that were not in the training sample. There was a 93% agreement between the machine and RA classification.” Along with: “The race and gender retrieved from administrative data matched with the algorithm classification with 95% accuracy” This is important because as accuracy along with other classification evaluation measure such as sensitivity and specificity is a great way to make sure your model is good at predicting.

Pearl J. has an article on the seven tools of casual inference. One connection I seen was with tool 6: Recovering from missing data. Pearl, J., (2019) states that tool 6 is “Problems due to missing data plague every branch of experimental science. Respondents do not answer every item on a questionnaire, sensors malfunction as weather conditions worsen, and patients often drop from a clinical study for unknown reasons.” In the article, there are demographic variables that has missing data because 37% of the applicants did not report their race and gender. They chose not to drop the applicants that did not report these variables because they do not know if this was random or a choice from the groups of applicants. They used Minnesota statewide administrative data to help classify the applicants into female and male categories and into white and nonwhite categories. They validated this approach with a random sample of 100 who also did not report their race and gander and the data taken from the Minnesota statewide administrative matched with the algorithm classification with 95% accuracy. (Sajjadiani, S., etc., 2019)

Another form of learning that goes along with evolution is Swarming. Gopal, M. (2019) states, “Swarm intelligence is a feature of systems of unintelligent agents with inadequate individual abilities, displaying collectively intelligent behavior. It includes algorithms derived from the collective behavior of social insects and other animal societies. The primary lines of research that can be recognized within swarm intelligence are: (i) Based on social insects (Ant Colony Optimization) (ii) Based on the ability of human societies to process knowledge (Particle Swarm Optimization).” There is an article from the GCU library called “Sum-Rate Maximization for UAV-Assisted Visible Light Communications Using NOMA: Swarm Intelligence Meets Machine Learning” and it is about how “as the integration of unmanned aerial vehicles (UAVs) into visible light communications (VLCs) can offer many benefits for massive-connectivity applications and services in 5G and beyond, this article considers a UAV-assisted VLC using nonorthogonal multiple-access. More specifically, we formulate a joint problem of power allocation and UAV's placement to maximize the sum rate of all users, subject to constraints on power allocation, quality of service of users, and UAV's position.” (Pham, Q., etc., 2020) This article is related to swarm intelligence by the following content from the article:

> “Motivated by a competitive performance with high reliability and fast convergence, swarm intelligence is considered as a key approach for optimizing the 5G and beyond networks. Harris hawks optimizer (HHO), as a swarm intelligence technique, is one of the most recent algorithms that has been popular since the proposal [22]. It has been considered in many engineering problems, e.g., the control chart patterns recognition for the manufacturing industry in [23], the design of microchannel heat sinks for the minimization of entropy generation in [24], and resource allocation in wireless networks [25], [26], which all show supervisor performance of the HHO algorithm.” (Pham, Q., etc., 2020)

References:

Gopal, M. (2019). *Applied Machine Learing*. New Delhi, India: McGraw-Hill Education.

Pearl, J. (2019). The Seven Tools of Causal Inference, with Reflections on Machine Learning. Communications of the ACM, 62(3), 54–60. https://doi-org.lopes.idm.oclc.org/10.1145/3241036

Pham, Q., Huynh-The, T., Alazab, M., Zhao, J., & Hwang, W. (2020). Sum-Rate Maximization for UAV-Assisted Visible Light Communications Using NOMA: Swarm Intelligence Meets Machine Learning. IEEE Internet of Things Journal, Internet of Things Journal, IEEE, IEEE Internet Things J, 7(10), 10375–10387. https://doi-org.lopes.idm.oclc.org/10.1109/JIOT.2020.2988930

Sajjadiani, S., Sojourner, A. J., Kammeyer-Mueller, J. D., & Mykerezi, E. (2019). Using machine learning to translate applicant work history into predictors of performance and turnover. Journal of Applied Psychology, 104(10), 1207–1225. https://doi-org.lopes.idm.oclc.org/10.1037/apl0000405