INFO 6105. Data Sci Eng Methods Spring 2020
Homework 3

Instructor: Dr. Handan Liu

Naïve Bayes and Linear Regression
March 16th, 2020

This homework will require you to implement and interpret some of concepts of Naïve Bayes and linear regression that were introduced in class. Further, you will be working with real-world data retrieved from an online repository, and while you will be asked to utilize a variety of modules and functions. Please finish the homework in Jupyter and submit ipynb file on Blackboard.

Working with Text Data and Naive Bayes in Scikit-Learn (50 points)

Representing text as data

1. Give a simple dataset

```
simple_train = ['call you tonight', 'Call me a cab', 'please call me... PLEASE!']
```

2. learn the 'vocabulary' of the training data
   use CountVectorizer() to "convert text into a matrix of token counts":

3. transform training data into a 'document-term matrix' (which is a sparse matrix)
   use “transform()”
4. print the sparse matrix
5. convert the sparse matrix to a dense matrix
   use “toarray()”
6. examine the vocabulary and document-term matrix together
   use pandas DataFrame and columns by using “get_feature_names()”
7. transform testing data into a document-term matrix (using existing vocabulary)
   use the test data as:
   simple_test = ["please don't call me"]
8. examine the vocabulary and document-term matrix together

Reading SMS data

9. read tab-separated file “sms.tsv”; give the names of columns as ['label', 'message']; and use head() to view part of the data.
10. convert label to a numeric variable
11. define X and y
12. split into training and testing sets by train_test_split(); and print the shape of training set and test set.

Vectorizing SMS data

13. instantiate the vectorizer by CountVectorizer()
14. learn training data vocabulary, then create document-term matrix “X_train_dtm”

15. transform testing data (using fitted vocabulary) into a document-term matrix

Building a Naive Bayes model by using Multinomial Naïve Bayes

16. train a Naive Bayes model using the matrix “X_train_dtm”
17. calculate accuracy of predictions
18. give the confusion matrix
19. print message text for the false positives
20. print message text for the false negatives

Reference:
https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

Linear Regression (50 points)

Please use the Linear Regression model from Scikit-Learn to predict housing prices in Boston.

The Data:

Load the Boston dataset. This is a dataset that's installed within Scikit-Learn.
Also refer to “Boston Data Set”:
https://www.kaggle.com/c/boston-housing

The Goal: Using Linear Regression on the dataset

The goal with this exercise: predict the housing price, using other columns (features) in the dataset.

What to Do

First, load the Boston housing data with the line below.

```
from sklearn.datasets import load_boston
boston = load_boston()
```

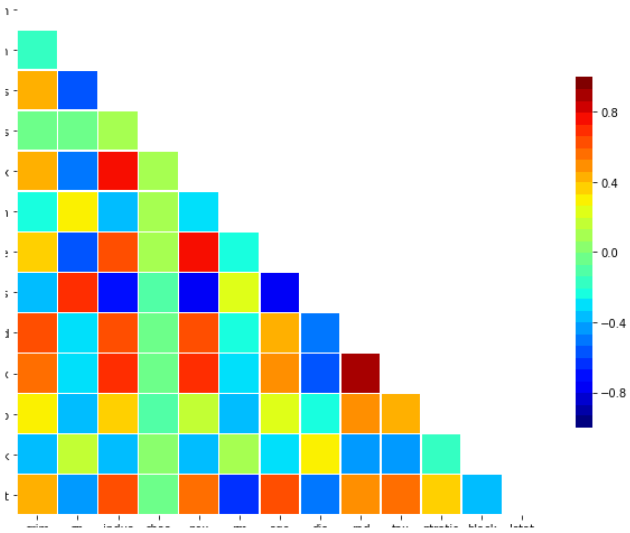
Next, separate the data into the features and target using the following code:

```
y = boston.target
boston = pd.DataFrame(boston.data)
```

The Tasks:

Now that the data is in the right format, please plot a correlation matrix to show what features are correlated with each other. For reference, -1 is uncorrelated, and 1 is highly correlated.

1. First, use boston.corr() to show what features are correlated with each other.
2. The columns don't have any labels. This happens with some datasets. According to the column values shown in https://www.kaggle.com/c/boston-housing, you need to label the columns. Note: only label 13 columns (the website has 14 labels for the columns), remove the last one of 'medv'. And use boston.corr() again.
3. Now, you have the numbers from the correlation matrix, but it's not as easy to view or interpret as a plot. Please plot correlations by color according to the following figure and giving the labels of x and y:



Given this output, respond to the following questions:

4. In the above plot, what features are highly correlated? Which features are highly uncorrelated?

Building the Linear Regression Model

5. First, please split the data into two datasets as “training” dataset and “test” dataset by train_test_split(). According to the four steps taught in the classes, please train the model by linear regression.

Predict and Score Model

6. Then, please predict new values using the test set.

Please give the coefficient for your model.

7. The sign of a regression coefficient tells you whether there is a positive or negative correlation between each independent variable and the dependent variable. What does a positive coefficient and a negative coefficient indicate respectively?

8. Finally, to gain an understanding of how your model is performing, please score the model against three metrics: R squared, mean squared error, and mean absolute error. Write the lines of code to get your output; and answer the questions:
   a) Google R Squared, Mean Squared Error, and Mean Absolute Error. What do these metrics mean? What are the numbers telling you?
   b) What do you think could improve the model? Try the possible improved model in coding lines as a bonus.