STAT 650 Assignment-07

Due: December 02, 2022 5:59PM

Instructions :

- This assignment is based on materials coverved in Lectures 19, 20 and 21.
- We highly recommend that you write your solutions in **Jupyter Notebook** and convert them to a **PDF** file. However, you may write the solutions by hand, scan and upload it as **.pdf** file.
- The PDF file should be under 15MB in size. It must be uploaded as a single file and not separate files for separate pages. Do not take a photo of each page and then paste them into a document - this will make your file too big and the results will generally not be very readable anyway.
- Please make sure that the solutions are neat, legible and in order (even if you choose to solve them in different order).
- Include **STAT650--UIN** at the top of the first page.
- Name the file as **UIN_assign7.pdf** (For eg, if someone's UIN is 123456789, then the file should be named 123456789_assign7.pdf). Otherwise, your submission will not be graded.
- You should upload your file through Canvas. You can make multiple submissions within the deadline, but only the latest submission will be considered for grading.
- You may take 6 hours extra after the due time, but 10% of your marks will be deducted.
- It is strictly prohibited to share or distribute the content in this document.

The aim of this assignment is to get familiar with Classification and Clustering.

Problem 1

This question uses the *california_housing* dataset from the `sklearn` package. To load the dataset, use the following code to load the dataset and set random seed as $SEED = 10$

```
from sklearn.datasets import fetch_california_housing;
california_housing = fetch_california_housing(as_frame=True);
df = california_housing.frame;
SEED = 10;
```

1. Create a new feature called $MedHouseValCat$ by discretizing the atribute $MedHouseVal$ into four classes labeled as 1,2,3, and 4 and adding it to your dataframe. Show that the frequency of these four class labels is approximately 5160 (Hint pd.qcut(feature_to_be _discritized, number_of_classes, retbins=False, labels)).
2. Define the feature matrix X by incluing $MedHouseVal$ and $MedHouseValCat$ and attributes and dependent variable matrix Y including $MedHouseValCat$.
3. Divide the data into training and testing by allocating 25 percent of the samples to testing.
4. Scale the data using the standard scaler and then perform a KNN classification using the five nearest neighbors. Show that the classification accuracies at training and testing are 74.47 and 63.48, respectively.
5. Visualize the confusion matrix as a heatmap and print the classification report for the testing dataset.

6. Tune the hyperparameters, the number of near neighbours, and the weight function using `GridSearchCV`. Show that 15 is the optimal number of nearest neighbors and distance is the best function for computing weight. Calculate the mean squared errors for the training and testing of the classifier using the tuned hyperparameters.
7. Answer this question based on the data in step 3. Create a pipeline that includes a standard scaler and a KNN classifier with 15 nearest neighbors. Use your pipeline to perform 10-fold cross-validation. Display the results of the classification showing an accuracy of 0.688 ± 0.018.
8. Answer this question using the data in step 3. Perform 10-fold cross-validation while changing the number of nearest neighbors from 2 to 20 and then plotting the validation curve.

Problem 2: Using the Pima Indians Database to predict Diabetes Outcome

The Pima are Native Americans based in Arizona. As a result of changes in diet and physical activity, they have developed a very high incidence of *Type 2 diabetes*. The anonymous medical data used in this notebook was obtained from 768 Pima women. It comprises 8 attributes that might be used to predict diabetes status (the 9th column in the dataset, which is the class to be predicted). You can download this data as a csv (comma-separated variable) file using the python code given below.

```
dataDir = "../data"
dataFile = dataDir + '/pima-indians-diabetes.csv'

import os.path
if not os.path.isfile(dataFile):
  import requests # Remember: you may need to install the requests module: conda i
nstall -c anaconda requests \
 url='https://gist.github.com/chaityacshah/899a95deaf8b1930003ae93944fd17d7/raw/3d
35de839da708595a444187e9f13237b51a2cbe/pima-indians-diabetes.csv'
  r = requests.get(url)
  with open(dataFile, 'wb') as f:
        f.write(r.content)

seed = 42
```

1. Load the *dataFile* as a data frame and name it as *pimaDf*.
2. While the existing predictor names are descriptive, they are cumbersome when used in models. Use the names given below to replace them with simpler predictor names `predNames` and `className`.
   $predNames = NumPreg, PlasmaGlucose, DiastolicBP, SkinFold, SerumInsulin, BMI, PedFn, AgeYrs$
   and $className = DiabetesClass$.
3. Given the fact that there are so many predictors (8, though often the number of predictors), it is often advisable to look for:
   a) correlations between predictors and
   b) correlations between predictors and the assigned class labels.
   Compute the `Pearson` correlation coefficient between the predictors as well as the predictors and the lables. Briefly describe the correlation matrix.
4. In order to see the general distribution for each column and also the effects of the missing values, generate histograms of the predictor values. What are the predictors that may contain missing?
5. A way of accommodating missing data is to impute values are statistically "neutral". Do this by assigning, when the predictor value is zero, the median of the remaining values of a predictor. Name the updated database as *filteredPimaDf*.

6. Using the *filteredPimaDf*, create training and tessting datasets by spliting 768 rows into train=514 and test=254 rows. Then, pply _MiniMax_scaler on the predicor metrices.
7. Create a Naive Bayes Classifier (`GaussianNB`) and show the classifcation diagnostics (e.g., accuracy (= 0.75984), Confusion Matrix, Classification Report).
8. Create a support vector machine (SVM) classifier (set the hyperparameter `gamma to auto`) and show the classifcation diagnostics (e.g., accuracy (= 0.7677), Confusion Matrix, Classification Report).

Problem 3

This question is intended to practice `Hierarchical` clustering and `K-Means` by using the *clusterdata.csv* file. The dataset should be loaded as a data frame and named *df*. Before clustering can be performed, the dataset must be processed. That is because it appears that there are two variables, but they are stored under a single variable name *Var1* separated by a comma. As an example, `df.head(3)` outputs

	Var1
0	-2.9191,0.32036
1	-5.4637,-0.87935
2	-3.0553,0.42799

1. Separate two values in df into two columns and name them *Var1* and *Var2* in a new data frame named *df_new*. If you run `df_new.head(3)` , you should see the following output:

	Var1	Var2
0	-2.9191	0.32036
1	-5.4637	-0.87935
2	-3.0553	0.42799

2. Create a scatter plot `Var1 vs Var2` .
3. Using the *df_new*, perform hierarchical clustering using the `linkage` method with the options `method='ward' and metric='euclidean'` . To represent the clustering process, create a *dendogram*.
4. Perform *Agglomerative Clustering* using the following settings: `n_clusters = 4, affinity = 'euclidean', linkage = 'ward'` . Make a scatter plot of the four clusters you created (you should use different colors).
5. Use the `df_new` to perform *K-Means clustering* (set `random_state = 0`). Identify the optimal number of clusters as four. Based on the optimal number of clusters, calculate the clustering parameters cluster centroids, the sum of the squares within each cluster, Cluster labels, and Cluster size.
6. Using the information in step 5, create a scatter plot that shows clusters and their centroids.
7. You should identify a suitable technique that will reduce the inertia value observed in step 5 to approximately 7 (prove this by performing clustering).

Problem 4

The intention of this question is to practice *K-Mean* clustering with a higher dimensional data. The dataset, *Live.csv*, contains information about a set of different reactions (e.g., number of comments, likes, shares, etc) on a public posts made by a group of students.

1. Load the dataset as a data frame *df* and inspect the dataset to identify unncecessery data and remove them.

2. Consider the three features, *status_id*, *status_published*, and *status_type*. Among these three features, identify the features that do not share certain type of common property (support your answer with sufficient evidence) with the remaining features of the dataset.
3. Based on your inspection in step 2, declare the feature vector *X* by removing the features that do share commom property. Also, define a target variable *Y* as the featre that shares the common property with the remaining data.

4. Apply `MiniMaxScaler` on the feature vector *X* and then perform *K-Mean* clustering with 2 clusters (set `n_clusters = 2, random_state = 0`).
5. Show that the *Inertia = 237.757*. Also, print Cluster centroids, cluster labels, and cluster size.
6. Compute `inertia` for different number of clusters (i.e., $k = 1, 2, 3, \cdots, 15$) and then plot `inertia` vs number of clusters. Use your plot show that the optimal number of clusters is approximately 3.
7. Use the `KElbowVisualizer` function to find the optimal number of clusters.
8. Fit `K-Mean` clustering method with the optimal number of clasters you selected in step 6 or 7 and show that the interia is reduced to 161.596.