Homework #3

TECH-GB.2335.10: Data Science for Business

Due: Nov 10, 2021, 18:00

- Code **MUST** be written in Python.
- You have been provided with data containing attributes for customers of a mobile phone provider. Please download *Telecom_Customers_clustering.csv* from Brightspace.
- The file contains 20 attributes on over 7000 customers
- Before proceeding, import numpy (as np) and call the following: **np.random.seed(1103)**. This ensures that the randomness in K-Means will be the same for everyone and so outputs will be uniform. The number above, 1103, is arbitrary, use the same number.
- Please submit a single Jupyter notebook, including any code, text and graphs.

Problem 1 (40 points) Data loading and transformation

i Load the data using *pandas.read_csv()* into a DataFrame

ii Split your data into two dataframes, train and test, as we did in class (using sklearn.train_test_split) with 80% of the data as train.

iii There are 3 numerical features: tenure, MonthlyCharges, TotalCharges. For these features, plot histograms of the training data. Your histograms must:

- Use 25 bins
- Have a grid
- Include the feature name in the title
- Label the Y-axis (determine what the Y-axis represents if you're not sure)

iv For 2 of the numerical features, tenure, MonthlyCharges, calculate the mean, μ, and the standard deviation, σ, **from the training data only**

v Standardize tenure, MonthlyCharges in both your train and test dataframes by subtracting corresponding μ and dividing by the corresponding σ (as computed from the training data). Do not overwrite the original columns. Instead, create two new columns: **tenure_std**, **MonthlyCharges_std**

vi Instantiate an object of the *OneHotEncoder* class from *sklearn*, with *drop='first'*. Call *fit* on this object using only one column, *InternetService* from the training data.

a Using this trained object, call *transform* for the same column from both the training and test datasets and append these newly created columns to the corresponding datafames with column names obtained from the *OneHotEncoder*

b Why did we use *drop='first'*?

Problem 2 (30 points) Clustering

i Using the standardized tenure_std, MonthlyCharges_std columns in your training data, cluster your data using the KMeans class from scikit-learn.

- Experiment with number of clusters = 3, 4, 5 and 6.
- Each time you cluster, call *predict* on your trained object (with the same two columns from your in-sample data) → this gives you cluster membership for each row. Assign these cluster values to a column called *cluster* in your train dataframe.
- Plot a scatter plot with color-coded clusters and centroids, as we did previously in class.
- Code to plot your clusters is below. You can pass in the variable *cluster_centers_* from your trained KMeans object, as is, for the argument centroids. **Include your 4 plots**

```
import matplotlib.pyplot as plt
%matplotlib inline

def plot_centroids(df, col1, col2, cluster_column, centroids):
  df.plot.scatter(col1, col2)
  colors=['red','green', 'blue', 'orange', 'black', 'brown']
  for i in range(len(centroids)):
      plt.plot(df.loc[df[cluster_column]==i, col1], \
               df.loc[df[cluster_column]==i, col2], \
               color=colors[i], ls='', marker='.', label='_nolegend_')
      plt.plot(centroids[i][0], centroids[i][1], color=colors[i], \
               label=f'centroid{i+1}', marker='*', markersize=20, \
               markeredgecolor='black', markeredgewidth=1.5)
  plt.legend(loc='best')
  plt.show()
  plt.close()
```

ii Looking at these 4 plots, pick one that you think segments your customers into useful groups, report the choice you made. Give your segments names (or a brief description) that you think describe these segments.

iii Now, using the choice you made for the number of clusters, and **without re-fitting** your KMeans object, call *predict()* on your test dataframe and add a *cluster* column to the dataframe and create a single plot. **Include your plot**.

iv We fixed the random seed so results are consistent. What part of the K-Means algorithm is random?

Problem 3 (30 points) Instantiate a *sklearn.linear_model.LinearRegression()* object as we did in class and fit it with the feature *tenure* and the one hot encoded features created from the *InternetService* column. Use *MonthlyCharges* as your target variable.

Use the **training data only** for the fit!

i What is the intercept of your linear regression

ii What is the coefficient on the *tenure* feature?

iii How many one hot encoded columns did you obtain from the *InternetService* column and what are the coefficients?

iv Call *predict()* on your trained object with the corresponding feature columns passed in from the **test data**. Calculate *r-squared* using *sklearn.metrics.r2_score()* to compare these predicted values to the target variable from your test data set.

a Report the r-squared value.

b Is this value low or high, in your opinion?