ISE-291: Introduction to Data Science

Term 221

Homework 04 2022

Covers: Topics 8-9 Material **Deadline: 15 December 2022**

Homework Presentation & Submission:

- You must ***attempt only the first problem (A), while the others (B, C, D & E) are for practice***.
- It is ***group homework***.
- Provide the ***output solution*** for each sub-problem (part). Otherwise, 50% marks will be deducted.
- Every sub-problem (part) should be answered on a ***DIFFERENT CELL*** as given in the template.
- ***EVERY CELL*** should have a problem and part number clearly written in the first line.
- All cells of your homework should be in ***CHRONOLOGICAL order***. One cell per sub-problem.
- Any text should be written as a comment in the ***code cell***. Do NOT modify the code cell into the markdown cell.
- Submit the entire homework as ***ONE single .ipynb*** document.
- ***Do NOT add/delete*** any cell in the given template.

Problem A [100 Marks]: Solve all the questions using Python. Use Pandas, Seaborn, Sklearn, etc., libraries for all the analysis. Consider the data given in Excel file HW4_DataA. Consider the following data description:

Table 1. Data description

Field	Description
StdID	Student ID
Prog_level	The student background in programming (Excellent, Good, Fair)
Gender	The student gender (Male or Female)
Job_Offered	Whether a job is offered to student or not (Yes, No).
Math_Courses	Number of Math courses taken by the student. For example, math students have many math courses. Business students have few number of math courses.
Stat_Courses	Number of Stat courses taken by the student.
Num_semesters	Number of semesters spent at the University.
Field_Exp	Field experience of the student (Summer Training (ST) or Internship (INT))
GPA	Student's cumulative GPA
Concentration	The analytics concentration selected by the student. This is the class (output) variable. There are four class labels under this variable: Business Analytics (BA); Decision Analytics (DA); Data Science Analytics (DS), and No Concentration (NC).

Do the following tasks (in exact sequence) using the "HW4_DataA" data:

A-1. *[10 marks]:* Given a classification problem along with the corresponding data, describe the steps to build and evaluate a classification model for solving the given problem.

A-2. *[10 marks]:* Discuss in concise words the difference between each of the following:

- **a)** ID3 decision tree classifier and Naive Bayes classifier.
- **b)** Classification rules and association rules

A-3. *[5 marks]:* What does the Entropy measure? What does it mean to say that the Entropy is 0?

A-4. *[5 marks]:* Describe how to avoid overfitting the data in classification.

A-5. *[10 marks]:* Read and display the data given in HW4_DataA and then do the necessary pre-processing as needed for answering the following questions. Refer to Table-1 for the data description. *(Note: For encoding, use dictionary for the ordinal feature, one hot encoding for the nominal features, and label encoding for the class variable.)*

A-6. *[15 marks]:* Build a decision tree classifier for predicting the class label. Fit the classifier using the first 120 records of the data and then test the classifier on the rest of the data. Print the accuracy, precision, and recall of the classifier.

A-7. *[10 marks]:* Calculate the Information Gain (IG) for the class variable given the feature selected as a root node.

A-8. *[10 marks]:* From the decision tree built in A-6, write three classification rules.

A-9. *[10 marks]:* Write an association rule for " Prog_level -> Field_Exp ", which rule has the highest accuracy? Write the corresponding support and accuracy.

A-10. *[15 marks]:* Repeat the scenario in A-6 using the Naïve Bayes CategoricalNB classifier. Set alpha to 1.0, class_prior=None, and fit_prior=True. Compare the performance of the Naïve Bayes against the decision tree classifier.

Problem B (Practice only. No submission required): Solve all the questions using Python. Use Pandas, Seaborn, Sklearn, etc., libraries for all the analysis. Consider the data already given in CSV file HW4_DataA and data description is given in Table 1:

Do the following tasks (in exact sequence):

B-1. Discuss in concise words the difference between each of the following:

- **a)** Good clustering and bad clustering
- **b)** Internal clustering measures and external clustering measures

B-2. What does it mean to say that the Silhouette score is -1 or close to -1?

B-3. Read and display the data given in HW4_DataA. Refer to Table-1 for the data description. Display the unique values for the categorical columns. Drop the StdID and Concentration (the ground truth) columns from the Data and store the modified data frame with the name newDF.

B-4. Draw the dendrogram for the data in newDF. How many clusters can you identify?

B-5. Execute the K-means and the Agglomerative clustering algorithms (from scikit-learn) considering the following:

- **a)** Use the first 120 records of the data for fitting and the remaining for testing.

b) Fit each clustering algorithm on the training data for values of k between 2 to 8 (inclusive) and plot the internal and external measures for each value of K.

c) Report the performance of the two algorithms on the testing data for k=4. Use both the internal and external clustering measures.

Problem C: (Practice only. No submission required.)

Consider the following python methods, available in naive Python, or numpy/pandas/sklearn libraries:

C-1. sklearn.model selection.train test split()

C-2. numpy.c

C-3. numpy.average()

C-4. numpy.exp()

C-5. scipy.stats.entropy()

C-6. sklearn.tree.plot tree()

C-7. sklearn.tree.export text()

C-8. sklearn.tree.accuracy score()

C-9. sklearn.tree.confusion matrix()

C-10. scipy.cluster.hierarchy ()

C-11. sklearn.metrics.silhouette_score()

C-12. sklearn.metrics.davies_bouldin_score()

C-13. sklearn.metrics.cluster.adjusted_rand_score()

C-14. sklearn.metrics.cluster.normalized_mutual_info_score()

Answer the following questions for each of the above methods:

- List all the argument of the method.
- Classify the arguments as positional or keyword arguments.
- Identify the data types for each of the arguments.
- Write the default values for each of the arguments.

Consider the following python class, available in sklearn library:

C-15. sklearn.tree.DecisionTreeClassifier()

C-16. sklearn.ensemble.RandomForestClassifier()

C-17. sklearn.naive bayes.CategoricalNB()

C-18. sklearn.naive bayes.GaussianNB()

C-19. sklearn.cluster.AgglomerativeClustering()

C-20. sklearn.cluster. KMeans()

Answer the following questions for each of the above classes:

- List all the methods and properties/attributes.
- Discuss any three input arguments for the above classes.
- Discuss the .fit() method.
- Discuss the .predict() method.
- Discuss the .classes attribute (wherever applicable).

☞ **Note: You must use help() function from python to answer all the above questions.**

Problem D: (Practice only. No submission required.)

Explain the following Python codes. Assume df represents an existing pandas' dataframe, where the columns are C1, C2,...C14,O1. The columns with odd numbers are categorical, and columns with even numbers are numerical. The columns with label 'O1' indicates output column. 'Set1' and 'Set2' are two random subsets of the rows of the dataframe. Also, assume that relevant libraries are imported before executing the following code:

D-1. Code-1:

```
P=df['C1'].value_counts()/len(df.index)
print(P)
```

D-2. Code-2:

```
np.average(df['C2'].values,weights=df['C4'].values)
```

D-3. Code-3:

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0,criterion='entropy',splitter='best')
ndf = pd.get_dummies(df,drop_first=True)
clf = clf.fit(ndf.drop('O1', axis=1), ndf['O1'])
```

D-4. Code-4:

```
from sklearn.ensemble import RandomForestClassifier
ndf = pd.get_dummies(df,drop_first=True)
rf = RandomForestClassifier(n_estimators=6,criterion='entropy',max_depth=2,
random_state=0)  rf = rf.fit(ndf.drop('O1', axis=1), ndf['O1'])
tree.plot_tree(rf.estimators_[2], feature_names = ndf.columns[0:-1],
class_names=['no','yes'], filled = True);
```

D-5. Code-5:

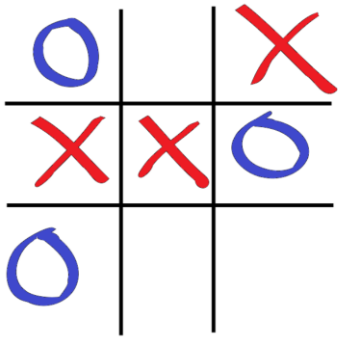
```
from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder()
x = encoder.fit_transform(df.values)
y = encoder.inverse_transform(x)[:,1]
```

D-6. Code-6:

```
X= df.iloc[:,:-1].values
y = df.iloc[:, -1].values
print(np.c_[np.ones(len(df.index)), X])
```

Problem E: (Practice only. No submission required.)

Consider data given in HW4_DataB.csv taken from a public repository. The data is related to tic-tac-toe game. Specifically, the database shows possible board configurations at the end of tic-tac-toe game between two players, 'x' and 'o'. In all the given board configurations, player 'x' played the first move. Each board end configuration is presented in a row (record), and there are 958 instances.



Each end configuration is represented by 9 features, corresponding to nine tic-tac-toe boxes. All the nine features contain exactly one of the following values: ‘x’, ‘o’ or ‘b’, where an ‘x’ indicates player ‘x’ took the box, an ‘o’ indicates player ‘o’ took the box, and a ‘b’ indicates the box is blank at the end of the game. The output column is ‘win-for-x’, where a ‘True’ value indicates player ‘x’ was the winner in that game instance, and a ‘False’ value indicates the game was either draw or player ‘o’ was the winner.

Do the following tasks using data given in HW4_DataB:

E-1. Given Data. Read the data and display the data. Count the number of rows and columns in the data. Count the number of non-null rows for each column. Display the description of both numeric and non-numeric columns.

E-2. Entropy & Information Gain. Do the following:

- Identify the entropy of the input column.
- Identify the input column(s) that has the maximum information gain. Report any ties.

E-3. Classification Rules. Do the following:

- Transform the input and output columns appropriately, so that the data can be used for building decision trees.
- Build a decision tree with max depth of 3. Do you see any pure leaf nodes?
- Build a decision tree with no restriction on max depth.
- Identify two classification rules one for each ‘True’ and ‘False’ class, which has the shortest length (least number of conditions on the if side).
- Build a random forests classifier with 4 estimators, and max depth of 3 for each estimator.
- Identify a classification rule that appears in 2 or more estimators.

☞ **Note: Solve all the above questions using Python (not by hand). Use Pandas, Seaborn, SkLearn, etc. libraries for all the above analysis.**

---------- THE END ----------