SVM – Activity Sheet

Reference:

Read CSV file:

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html>

Train-test split:

<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html#sklearn.model_selection.train_test_split>

Standard scaler:

<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler>

Simple Imputer:

<https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html#sklearn.impute.SimpleImputer>

Support Vector Classifier:

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>

Confusion Matrix:

<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix>

**Dataset**: Hepatitis

**Attribute information:**

1. **target**: DIE (1), LIVE (2)
2. **age**: 10, 20, 30, 40, 50, 60, 70, 80
3. **gender**: male (1), female (2)

------ no = 2, yes = 1 ------

1. **steroid**: no, yes
2. **antivirals**: no, yes
3. **fatique**: no, yes
4. **malaise**: no, yes
5. **anorexia**: no, yes
6. **liverBig**: no, yes
7. **liverFirm**: no, yes
8. **spleen**: no, yes
9. **spiders**: no, yes
10. **ascites**: no, yes
11. **varices**: no, yes
12. **histology**: no, yes
13. **bilirubin**: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00 --
14. **alk**: 33, 80, 120, 160, 200, 250 ---
15. **sgot**: 13, 100, 200, 300, 400, 500, ---
16. **albu**: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0, ---
17. **protime**: 10, 20, 30, 40, 50, 60, 70, 80, 90, ---

NA's are represented with "?"

Part-A: Read and understand Data

1. Read the ‘csv’ file provided and perform the following functions:
   1. Get the dimensions of the data.
   2. Get the summary.
   3. Print the first 5 rows and last 5 rows.
   4. Explore the data types of each column.
   5. Understand if all the columns are in appropriate datatypes or not.
   6. Get the number of rows for each level in your target variable (use value\_counts function)
   7. Is the data balanced w.r.t target column?

Part-B: Data Pre-processing

1. Are there any columns which are not useful or redundant?
2. Remove the ‘ID’ column. Verify with the head command whether you have removed from the original dataframe.
3. Create an array of all the numeric datatypes. Similarly, create an array of all categorical datatypes.
4. Verify if there are any null values in each of the columns. (don’t impute now).
5. Now, Create X and y from the original data-frame.
6. Apply train-test split (preferably 80:20).
7. Observe the target variable proportion of the target.
8. Now, check the null values in train and test. See if all of them are Categorical or from Numeric datatypes.
9. On the train, Separate the dataframe into numeric and categorical separately.
10. Use Imputer function from sklearn to impute all the null values. (Separately for Numeric and Categorical datatypes).
11. Use the ‘concat’ function from pandas to concatenate numeric and categorical dataframes to a single dataframe.
12. Repeat this again for Test data as well
13. Dummify both the test and train using get\_dummies function.
14. Scale the data using Standard Scaler function.

Part-C: Model Building

1. What are different hyper-parameters that you use in Support Vector Machines?
2. Import SVC class from sklearn and instantiate the class with parameters as “(kernel='linear', C=1, random\_state=0)”.
3. Run fit function on the train data.
4. Look at the accuracy and f1 score of the data and compare train and test performance.
5. Now, re-train the model with a new kernel and different hyperparameters with “(kernel='rbf', random\_state=0, gamma=0.01, C=1)”.
6. Apply fit, predict and observe the performance.
7. Now, perform Grid Search by giving multiple values of kernels, gamma values and slack(‘C’).