CE802– Machine Learning and Data Mining

Assignment: Design and Application of a Machine Learning System for a Practical Problem

Student’s name: Monali Gothi

Registration Number: 2010397

Professor’s name: Dr. Vito De Feo

Date: 21/07/2021

Report

(Word Count: 794)

In this assignment, we have built different classification models for “CE802_P2_Data” decision trees, support vector machines, naïve bayes, k-nearest neighbors. We have built different regression models for the “CE802_P3_Data” decision trees regressor, support vector machines regressor, k-nearest neighbors regressor, random forest regressor.

I have data given first for both tasks in the form of CSV files. After that, we have to check data if any value NaN then either drop that row or find the mean of that column and replaces the NaN with mean. Also, I need to check if any data is in the categorical form then convert that data into the numeric form using a different method like a one-hot encoder, ordinal encoder. I have split the dataset into 80% and 20% in train and test for both tasks. I have plot features and target scatterplot using seaborn. I have built a model then train that model using the sklearn library. Evaluate model for classification and regression model with different metrics like precision, recall, accuracy, f1-score for classification and mean square error, mean absolute error, accuracy, root mean square error, r^2 score.

We have two CSV files one file for implement the model with features and target and another for the test that model with just features. I have read and load data using the panda read_csv function. I have saved the features and targets in different variables. For, the first task “CE802_P2_Data.csv” file, I have tried two things one drop null value column and another compute mean of that column and replace the NaN value with that mean. For the second task file “CE802_P3_Data.csv”, I have used an ordinal encoder to convert categorical values to numeric. After that, I have split that data into test and train. I have implemented different models for the first task like DecisionTreeClassifier, GaussianNB, KNeighborsClassifier, SVC. I have implemented a grid search to track down the best params and scores for that model. I have found out SVC works best from other models with the mean of that column and replaces the NaN value with it and C=5000. So, using the SVC model I have filled the “CE802_P2_Test.csv “ file column “Class”. I have assessed utilizing the confusion matrix, classification report, and precision-recall curve for that model. I have implemented different models for the second task like DecisionTreeRegressor, LinearRegression, SVR, RandomForestRegressor, KNeighborsRegressor. I have discovered SVR works best from different models with C=5000. So, using the SVR model I have filled the “CE802_P3_Test.csv” file column “Target”. For the first task, I have predicted customers to buy an expensive or cheap product. For the second task, I have predicted the amount of money that a customer usually spends in a month. I have evaluated mean square error, mean absolute error, accuracy, root mean square error, r^2 score for that model.

Decision Tree:

With drop NaN

Accuracy: 0.8683

{'criterion': 'gini', 'max_depth': 1}

With mean NaN

Accuracy: 0.9216

{'criterion': 'gini', 'max_depth': 4}

Support Vector Machine:

With drop NaN

Accuracy: 0.8816

{'C': 5000, 'kernel': 'rbf'}

With mean NaN

Accuracy: 0.9358

{'C': 5000, 'kernel': 'rbf'}

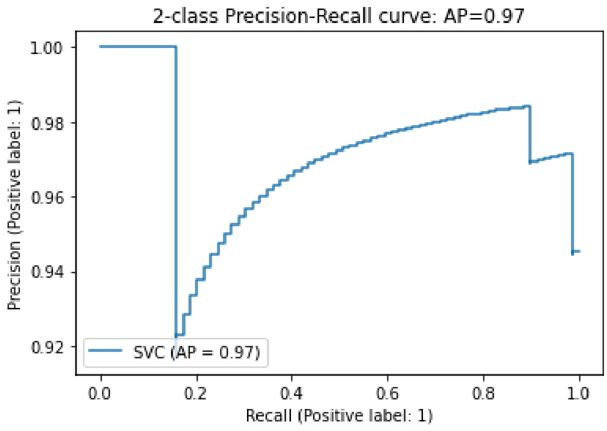

Fig. 1

Average precision-recall score: 0.97

I got 92% accuracy, replacing NaN with mean value using criterion=gini and max_depth=4 parameters with the Decision Tree model. I got 93% accuracy on train data, replacing NaN with the mean value using e using C=5000 and kernel=rbf parameter with the SVM model. So, SVM is better than the Decision Tree model. Utilizing the SVM model we predicted class for task one client purchases a costly or cheap product on test information.

Precision-Recall curve got 97% accuracy shows Fig. 1 using sklearn library metric. Below, Fig. 2 is the metric for that. Using that I got 80% accuracy in test data.

	Precision	**Recall**	**F1-score**	**Support**
False	0.91	0.83	0.86	231
True	0.55	0.71	0.62	69
Accuracy			0.80	300
Macro avg	0.73	0.77	0.74	300
Weighted avg	0.82	0.80	0.81	300

Fig. 2

Fig. 3 shows a scatter plot of the predicted and actual value of the Linear regression model and got 66% accuracy.

Accuracy: 0.668
Mean Absolute Error: 497.957
Mean Squared Error: 386621.600
Root Mean Squared Error: 621.789
R² Score: 0.668

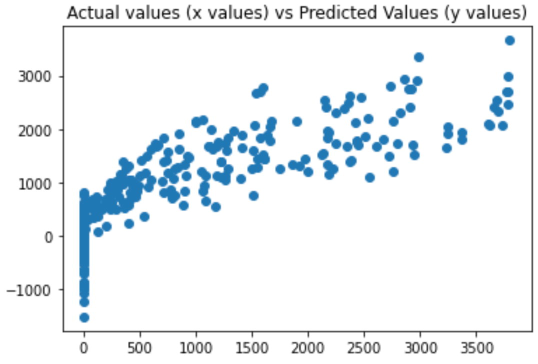

Fig. 3

Fig. 4 shows a scatter plot of the predicted and actual value of the SVR model and got 76% accuracy.

Accuracy: 0.760
Mean Absolute Error: 384.346
Mean Squared Error: 279339.190
Root Mean Squared Error: 528.525
R² Score: 0.760

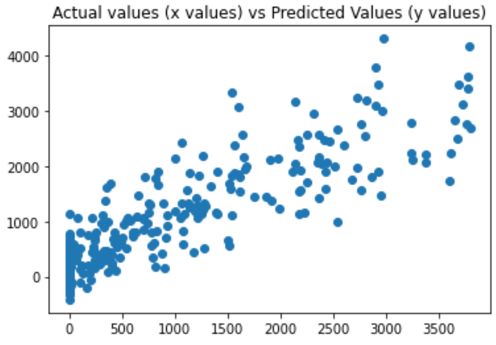

Fig. 4

Fig. 5 shows a scatter plot of the predicted and actual value of the RandomForestRegressor model and got 70% accuracy in train data.

Accuracy: 0.705
Mean Absolute Error: 443.647
Mean Squared Error: 344068.126
Root Mean Squared Error: 586.573
R^2 Score: 0.705

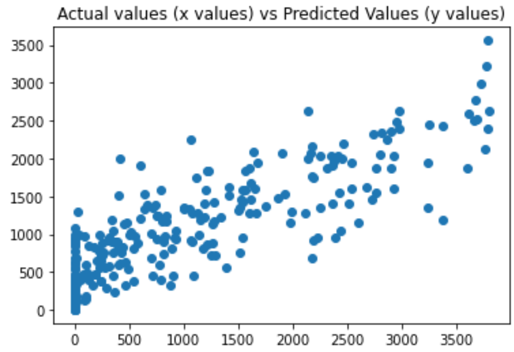

Fig. 5

For the above result, SVR was the better model than the other. So, using that we have predicted what amount spend customer monthly.

Reference

1. Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.