Eren Köseoğlu

Project 3

CMPE 541- Data Mining

TED UNIVERSITY
June 02, 2020

Summary

Here, after creating corpus files for both Global and Personalized prediction in Project 2, I have decided to use only the corpus file namely “CorpusGlobal75” that has been created for Global prediction in machine learning. To be able to use the corpus files that have been created for Personalized prediction, a few amendments were needed. Since it was so costly to amend and the results have not changed significantly, I have used “CorpusGlobal75” file during this project. In the end, mean squared error (MSE), root of mean squared error (RMSE) and percentage of the predictions fall into 2%, 5%, 10%, and 20% intervals for the 3 settings have been calculated. For Project 3, Python programming language was used.

Introduction

For the three settings, ”CorpusGlobal75” file which contains the 75% of the features has been used. If I chose “CorpusGlobal90” file, almost all of the most correlated features with “Value” would be in the file, which is good. However, the correlation among the features would also be higher. It is better to state that correlated features might worsen the model. That’s why, to decrease bias, I preferred CorpusGlobal75 file. This file contains the data of 70 patients.

For the three settings, the data set has been divided into 3 parts namely, “warm-up data”(20%), “training data”(60%) and “testing data”(20%). The first 20% of data which is called “warm-up data” was dropped before starting each setting by using a for loop. Here, if the each patient’s file length is an odd number, 20% of it will be decimal. Then, “int” function rounds it down. This time, it can be less than 20%. To make sure that it is exactly at least 20%, as seen in Figure 1, I added 1 to “warmup”variable.

```
for i in range(1, 71):
    data=pd.read_csv(r"C:\\Users\\Lenovo\\Desktop\\CorpusGlobal75\\"+ str(i) + ".txt")
    warmup=int(len(data)*0.2+1)
    data.drop(range(warmup),axis=0, inplace=True)
    data.to_csv(r"C:\\Users\\Lenovo\\Desktop\\CorpusGlobal75\\"+ str(i) + ".txt", index=False)
```

Figure 1

After dropping “warm-up data”, in order to split the data set in training and testing data, 25% has been used as test size and 75% has been used as training size.(if it is 60% in 80%, then it is 75% in 100%) For this project, Linear Regression, Support Vector Machine Regression(SVR), Random Forest Regression, Decision Tree Regression, Bayesian Ridge, Ridge and AdaBoost Regression algorithms have been applied. Since there is no C4.5 algorithm in Python, I used Decision Tree Regression which does the same work.

Global Prediction

In order to set up machine learning models for Global prediction, after dropping the warm-up data from each file, these 70 files have been combined again. Then, machine learning algorithms have been applied to the data. In the end, MSE and RMSE values have been calculated as seen in Table 1.

	MSE	RMSE
Linear Regression	5386.51	73.39
SVR	5462.55	73.91
Random Forest Regression	5478.48	74.02
Decision Tree Regression	10110.39	100.55
Bayesian Ridge	5357.33	73.19
Ridge	5386.2	73.39
Ada Boost Regression	5954.08	77.16

Table 1

According to Table 1, it can be concluded that the Bayesian Ridge algorithm is the best at predicting values since MSE level is the least.

Personalized Prediction

In order to make the predictions, the same corpus file was used. This file has been divided into 70 files since the prediction was made individually. There was no need to drop the warm-up data since it was already dropped.

After machine learning algorithms have been applied to the data, the average of MSE and RMSE values and the patient who has the lowest MSE and RMSE have been found as seen in Table 2.

	Average of MSE	Average of RMSE	Patient with the least error	Least MSE	Least RMSE
Linear Regression	8370.59	86.39	68	811.9	28.49
SVR	6964.49	79.87	49	884.25	29.74
Random Forest Regression	7100.91	80.94	31	711.89	26.68
Decision Tree Regression	12666.31	107.91	31	967.19	31.1
Bayesian Ridge	6610.91	78.17	68	808.79	28.44
Ridge	7826.18	83.86	49	796.05	28.21
Ada Boost Regression	7290.78	82.17	68	937.9	30.63

Table 2

According to Table 2, Bayesian Ridge Algorithm is the best at predicting since its average MSE value is the least. However, on an individual basis, Random Forest Regression Algorithm is the best since Patient 31 has the least MSE value.

Similarity-Based Prediction

Instead of predicting 70 patients' blood sugar value, only blood sugar value of Patient 68 has been predicted in this setting. I have splitted the each data set of 70 patients in training and test data set as seen in Figure 2 and stored the test data set in lists to use them later. Then, as guided in the project file, I took the average of "x_train" data set and appended them to another list.

```
for i in range(1,71):
    datapatient = pd.read_csv(r"C:\\Users\\Lenovo\\Desktop\\Corpus75_Personalized\\"+ str(i) + ".txt")
    temp=datapatient.drop(['Value',"Time_Stamp","User ID"],axis=1)

    x=temp
    y=datapatient["Value"]
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=45)
    xtestliste.append(x_test)
    ytestliste.append(y_test)
    liste.append(x_train.mean())
```

Figure 2

For the next step, I have selected Patient 68 as mentioned. The closest 5 patients have been found by using Euclidian distance. After this, the data set of the 5 closest patients has been combined to train the models. As a next step, machine learning algorithms have been applied to the data. The test data set of the reference patient has been used to predict the dependent variable. In the end, MSE and RMSE values was found as seen in Table 3.

	MSE	RMSE
Linear Regression	788.08	28.07
SVR	989.79	31.46
Random Forest Regression	1287.07	35.88
Decision Tree Regression	2492.31	49.92
Bayesian Ridge	815.33	28.55
Ridge	767.76	27.71

Ada Boost Regression	1102.72	33.21

Table 3

According to Table 3, it can be stated that the Ridge algorithm is the best at predicting values for Patient 68 since MSE level is the least.

Conclusion

For this project, by using the same corpus file, after dropping "warm-up data", Bayesian Ridge, Ridge, Ada Boost Regression, Linear Regression, SVR, Random Forest Regression and Decision Tree(CART) algorithms have been applied in the settings. Then, MSE, RMSE and the percentages of predictions fall into the given intervals have been calculated. Percentages of the predictions fall into the given intervals can be seen in Table 4.

	Percentage of the predictions fall into					
	0-2%	**0-5%**	**0-10%**	**0-20%**	**20%-100%**	**Setting**
Bayesian Ridge	3.78	9.05	17.84	35.01	64.99	Global
Ridge	3.41	8.42	17.21	34.83	65.17	Global
Ada Boost	3.34	8.38	17.47	33.57	66.43	Global
Linear Regression	3.41	8.38	17.21	34.79	65.21	Global
SVR	3.00	8.01	16.95	33.72	66.28	Global
Random Forest	4.34	9.46	18.84	36.64	63.35	Global
Decision Tree	3.3	8.01	16.21	30.34	69.66	Global
Bayesian Ridge	2.76	7.17	14.02	28.91	71.09	Personalized
Ridge	3.13	7.7	14.9	28.72	71.28	Personalized
Ada Boost	2.75	7.08	14.57	29.23	70.77	Personalized
Linear Regression	2.82	7.38	15.0	29.25	70.75	Personalized
SVR	1.65	6.13	13.43	27.09	72.91	Personalized
Random Forest	3.39	8.42	15.65	29.75	70.25	Personalized
Decision Tree	2.29	6.37	12.67	23.61	76.37	Personalized
Bayesian Ridge	6.6	23.58	36.79	57.54	42.45	Similarity-Based
Ridge	4.72	20.76	35.85	55.66	44.34	Similarity-Based
Ada Boost	5.66	16.04	28.3	46.22	53.77	Similarity-Based
Linear Regression	5.66	21.7	35.85	56.6	43.4	Similarity-Based
SVR	4.72	17.93	27.36	52.83	47.17	Similarity-Based
Random Forest	7.55	19.81	30.19	51.89	48.11	Similarity-Based
Decision Tree	0.94	5.66	17.92	37.73	62.26	Similarity-Based

Table 4

By considering Table 1 and Table 2, it can be concluded that the results of Global Prediction are better than the average results of Personalized Prediction. By taking into account all of the results, it can be said that the features are not capable enough of explaining the change in blood sugar measurements. This leads to high MSE and RMSE values and low prediction accuracy.