CE 802 Machine Learning

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

University of Essex

School of Computer Science and Electronic Engineer

Comparative Report Study

Submitted by

Supasun khumpraphan

Registration number:2110366

Supervisor: Dr. Vito De Feo

MSc - artificial intelligence

Word count: 1357 words

1. Summary: The National Health Service wants to make predictions to know if people have diabetes. To help people know the disease in advance and be able to deal with it in a timely manner. resulting in fewer deaths or less severe illnesses. we have data (Feature) from National Health Service used to predict results and There is an answer (label). that predict results True(diabetic patient) and False(non-diabetic patient), so it is definitely a classification problem. There is also data(Feature) that needs to be designed to determine the daily dose(continuous) that a diabetic person should take in order for a person with diabetes to receive the appropriate dosage. which is a solution to the problem of regression.

2. Methodology:

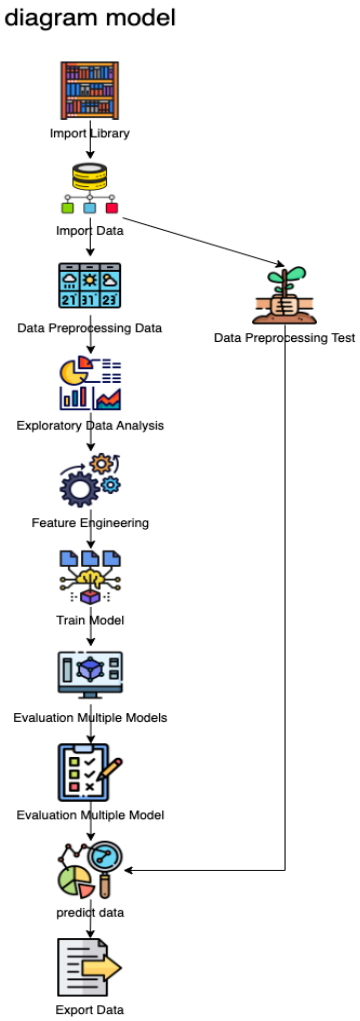


Fig. 1: diagram model of the project.

2.1. Data Preprocessing

From our CE802_P2_Data dataset, it is evident that the data is supervised learning because we provide tutorials with answers and use them to make predictions in Discrete format, which is a classification model. Then you can notice that the data in column “F15”, both Data

and Test dataset, is partially blank. So it fills the blank with the mean of each class, then converts the Data dataset column class to have the "False" equal 0 and the "True" equal 1. to be ready for the train model while the data set CE802_P3_data It showed that the data was supervised learning because it contained features value and predicted outcomes in a continuous. Make it in a regression format. and need to convert column “F5” to Quantitative variable must replace value is numeric and “F6” is Qualitative variable which cannot be numerically measured must be converted with get_dummy.

2.2. Exploratory Data Analysis

We will use Exploratory Data Analysis to display the data in a visual format to make it easier to understand every column of data. In the first step, we use Histogram(Fig. 2) To analyze the distribution of data and detect abnormal data characteristics.

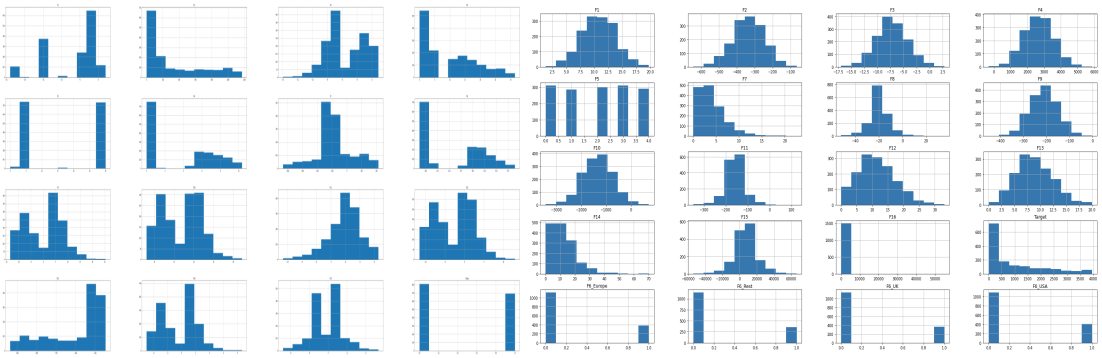


Fig. 2: visualization p2(left) and p3(right) with histogram.

After that, we will use a heatmap(Fig. 3) to Present trend or correlation information simply by using color pairs, and color intensity represent quantity or frequency, especially data that is near -1 or 1, meaning the two are related.

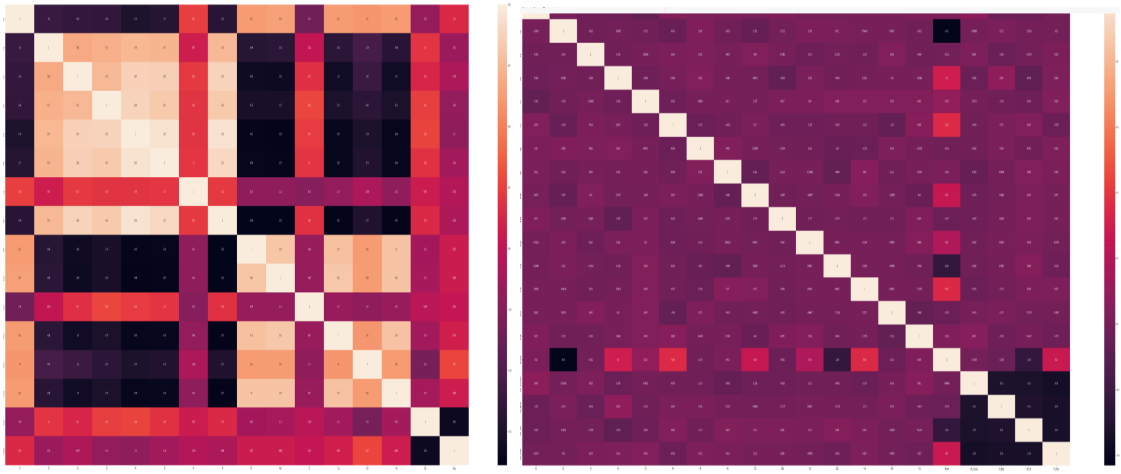


Fig. 3: visualization p2(left) and p3(right)with heatmap.

We then use a box plot to see if the Outlier data is abnormally high. or abnormally low The outlier data must be less than Q1-1.5*(Q3-Q1) or higher than Q3+1.5*(Q3-Q1)

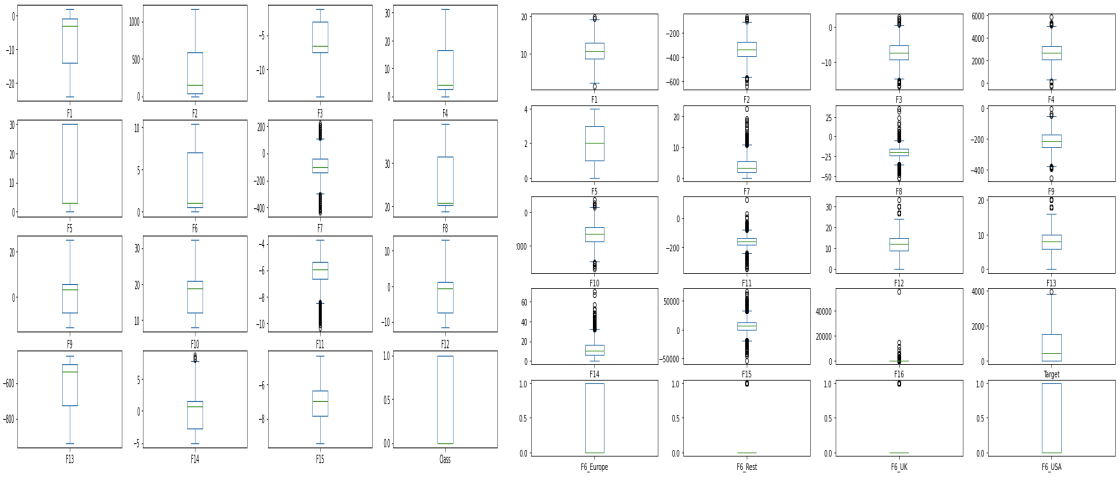


Fig. 4: visualization p2 and p3 with a boxplot.

2.3. Feature Engineering

After doing Exploratory Data Analysis, we can see in the dataset. CE802_P2_Data that the label "class" column contains almost equal data which is balanced. data with a slightly higher False than True. And column F7, F11, F14 has a number of outliers so we need to limit the outliers to optimize the model. Also in the dataset CE802_P3_Data, There are outliers for almost every column except F5, F6_Europe, and F6_USA, so the outlier has to be removed to increase the model's performance as well.

2.4. Train_test_split

We have clean data for both CE802_P2_Data and CE802_P3_Data, we need to take each file to train_test_split. so that data does not overfitting[2]. Therefore, we divide the data into train set 80 percent and test set 20 percent. After separating train data and test data, in CE802_P2_Data there are 1180 sample data for the train set and test set 296 sample data and in CE802_P3_Data there are 1174 sample data for the train set. and test set 294 sample data. Both data files are normal distribution so both standard scalers are used.

2.5.1. Train Model with Classification

The CE802_P2_Data data can be predicted supervised learning in a classification model. We have used Decision Tree Classifier, Logistic Regression, Random Forest Classifier, K-nearest Neighbors, Support Vector Classification, Gaussian Naive Bayes, XGBoost Classifier, Stochastic Gradient Descent Classifier, and deep learning models such as multi-layer perceptron Classifier for a total of 9 models. In the beginning, we will train every model in its default form parameter by looking at Fig. 5 It can be seen that the confusion matrix at The most accurate model is a multi-layer perceptron classifier with a total of only 12 faults. it can be split is 7 false positives and 5 false negatives.

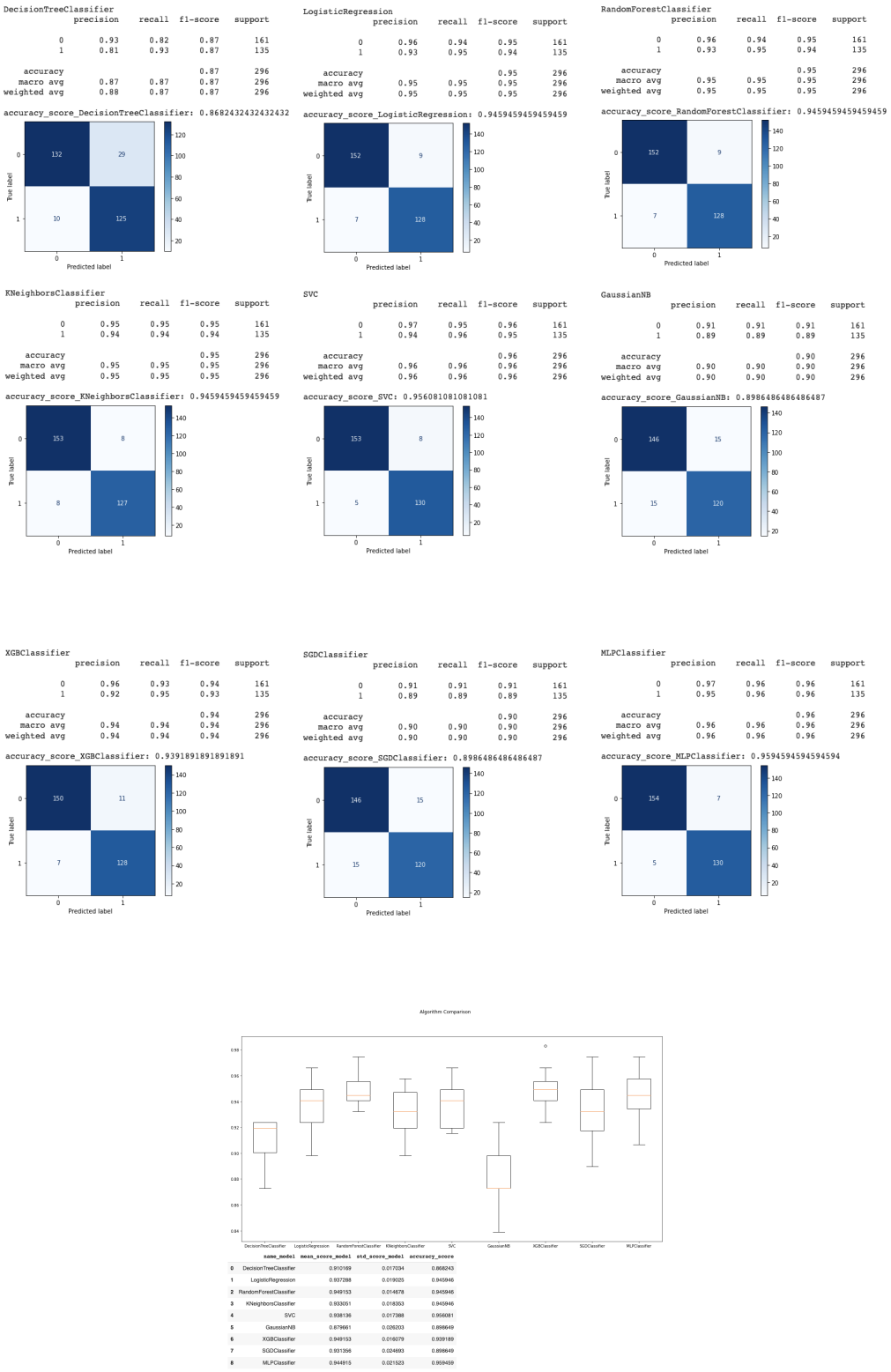
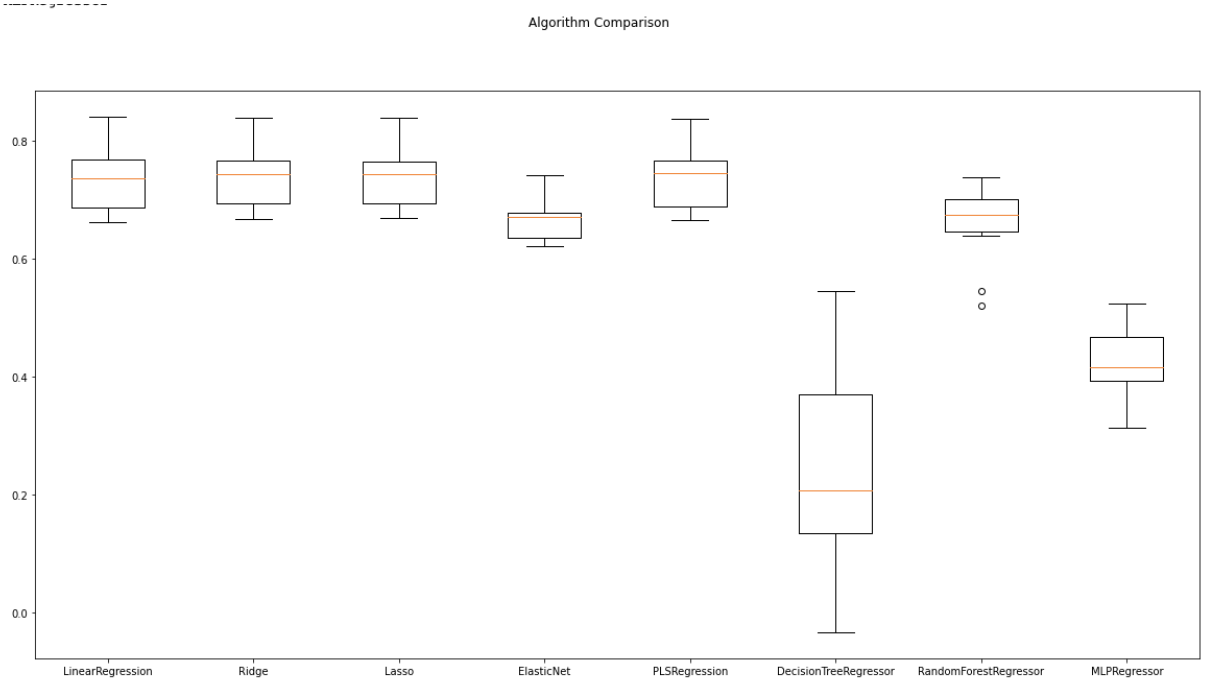


Fig. 5: all accuracy models with default parameter check with a box plot, confusion matrix, and data frame.

2.5.2. Train Model with Regression

CE802_P3_Data data is supervised learning in regression format. We use linear regression, ridge regression, lasso regression, Elastic net, Partial least squares regression, Decision Tree, Random forest and deep learning multilayer perceptron regression for a total of 8 models, where we train every model in the default format. parameters to see the performance of the rough model by looking at Fig. 10 It is seen that lasso regression is only slightly more effective than linear regression, ridge regression, Partial least squares regression



	name_model	mean_score_model	std_score_model	root_mean_square_error	r2_score
0	LinearRegression	0.733770	0.054211	4.024104e+14	-1.279317e+23
1	Ridge	0.736847	0.052695	5.752527e+02	7.385691e-01
2	Lasso	0.737082	0.052407	5.759216e+02	7.379607e-01
3	ElasticNet	0.666077	0.035917	6.815075e+02	6.330720e-01
4	PLSRegression	0.736188	0.052989	5.777947e+02	7.362535e-01
5	DecisionTreeRegressor	0.247757	0.177722	8.877735e+02	3.773501e-01
6	RandomForestRegressor	0.656167	0.067368	6.718541e+02	6.433933e-01
7	MLPRegressor	0.424504	0.057787	9.524244e+02	2.833607e-01

Fig. 10: Virtualize box plot and dataframe to check score accuracy, root mean square error, and R-Squared with default parameter

2.6. Tuning Parameter

After we train the model with default parameters, we did not get the best model values, we have to develop the model by tuning parameters in all 9 model of classification(Fig. 6) and 8 model of regression(Fig. 11). We will use RandomizedSearchCV. instead of GridSearchCV as it takes faster and some research says RandomizedSearchCV. Better performance than

GridSearchCV[3]

	model	best_score	best_params	best_estimator
0	DecisionTreeClassifier	0.896610	{'splitter': 'best', 'min_samples_split': 4, '...	DecisionTreeClassifier(max_features='auto', mi...
1	LogisticRegression	0.939831	{'solver': 'saga', 'penalty': 'none', 'multi_c...	LogisticRegression(max_iter=400, multi_class='...
2	RandomForestClassifier	0.950847	{'n_estimators': 400, 'min_samples_split': 10,...	(DecisionTreeClassifier(max_features='auto', m...
3	KNeighborsClassifier	0.938136	{'weights': 'uniform', 'p': 1, 'n_neighbors': ...	KNeighborsClassifier(n_neighbors=7, p=1)
4	SVC	0.937288	{'verbose': True, 'shrinking': True, 'probabil...	SVC(break_ties=True, probability=True, verbose...
5	GaussianNB	0.885593	{'priors': None}	GaussianNB()
6	XGBClassifier	0.955085	{'n_estimators': 75, 'min_child_weight': 5, 'm...	XGBClassifier(colsample_bytree=0.3, gamma=0.2,...
7	SGDClassifier	0.943220	{'shuffle': True, 'penalty': 'l1', 'max_iter':...	SGDClassifier(l1_ratio=0.85, max_iter=2000, pe...
8	MLPClassifier	0.952542	{'warm_start': True, 'verbose': False, 'valida...	MLPClassifier(activation='tanh', max_iter=800,...

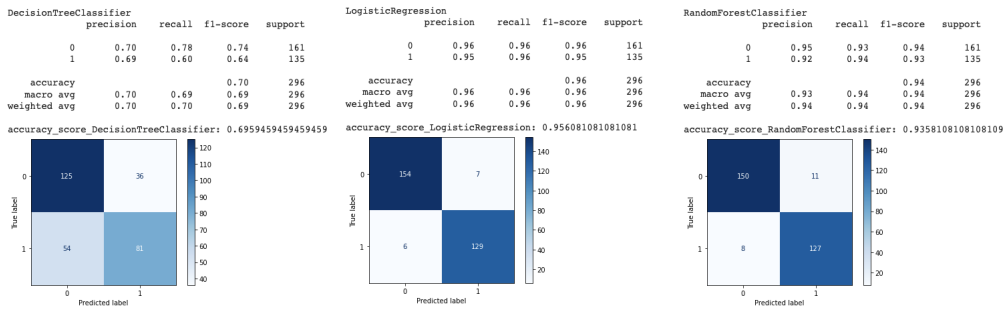
Fig.6 tuning parameter ตามแต่ละmodelในclassifier

	model	best_score	best_params	best_estimator
0	LinearRegression	0.740094	{'positive': False, 'n_jobs': -1, 'fit_interce...	LinearRegression(n_jobs=-1)
1	Ridge	0.742397	{'solver': 'cholesky', 'positive': False, 'max...	Ridge(alpha=4, max_iter=False, solver='cholesky')
2	Lasso	0.743090	{'warm_start': True, 'selection': 'random', 'p...	Lasso(alpha=4, max_iter=500, precompute=True, ...
3	ElasticNet	0.743078	{'warm_start': False, 'selection': 'random', '...	ElasticNet(alpha=4, copy_X=False, l1_ratio=1, ...
4	PLSRegression	0.742362	{'scale': True, 'n_components': 4, 'max_iter':...	PLSRegression(n_components=4)
5	DecisionTreeRegressor	0.398233	{'splitter': 'random', 'min_samples_split': 10...	DecisionTreeRegressor(criterion='friedman_mse'...
6	RandomForestRegressor	0.654654	{'n_estimators': 700, 'min_samples_split': 5, ...	(DecisionTreeRegressor(max_features='sqrt', mi...
7	MLPRegressor	0.923699	{'warm_start': True, 'verbose': True, 'validat...	MLPRegressor(activation='logistic', max_iter=8...

Fig.11 tuning parameter ตามแต่ละmodelในregression

2.7. Evaluation Multiple Model with tuning parameter

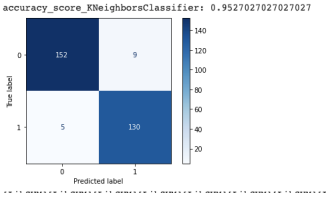
After we have tuning parameter with RandomizedSearchCV Then we use it instead of the default parameter to make our model better predictive performance. From Fig 7 it can be seen that the model multi-layer perceptron Classifier in the data frame and box plot measured with cross-validation has better accuracy than all models with an accuracy of 0.952542 percent. The matrix shows that there are only 8 errors divided into 5 false positives and 3 false negatives.

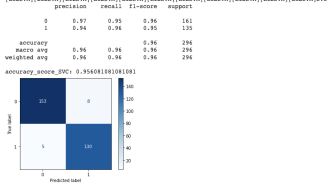


KNeighborsClassifier

	precision	recall	f1-score	support
0	0.97	0.94	0.96	161
1	0.94	0.96	0.95	135
accuracy			0.95	296
macro avg	0.95	0.95	0.95	296
weighted avg	0.95	0.95	0.95	296

accuracy_score_KNeighborsClassifier: 0.9527027027027027

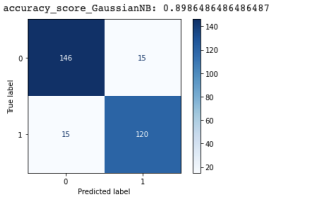




GaussianNB

	precision	recall	f1-score	support
0	0.91	0.91	0.91	161
1	0.89	0.89	0.89	135
accuracy			0.90	296
macro avg	0.90	0.90	0.90	296
weighted avg	0.90	0.90	0.90	296

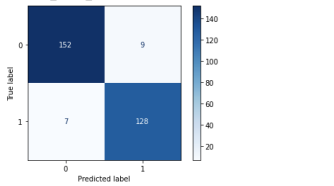
accuracy_score_GaussianNB: 0.8986486486486487



XGBClassifier

	precision	recall	f1-score	support
0	0.96	0.94	0.95	161
1	0.93	0.95	0.94	135
accuracy			0.95	296
macro avg	0.95	0.95	0.95	296
weighted avg	0.95	0.95	0.95	296

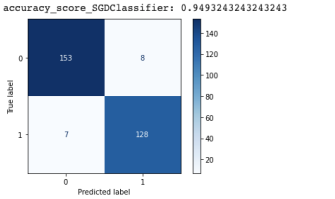
accuracy_score_XGBClassifier: 0.9459459459459459



SGDClassifier

	precision	recall	f1-score	support
0	0.96	0.95	0.95	161
1	0.94	0.95	0.94	135
accuracy			0.95	296
macro avg	0.95	0.95	0.95	296
weighted avg	0.95	0.95	0.95	296

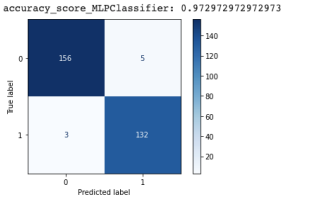
accuracy_score_SGDClassifier: 0.9493243243243243

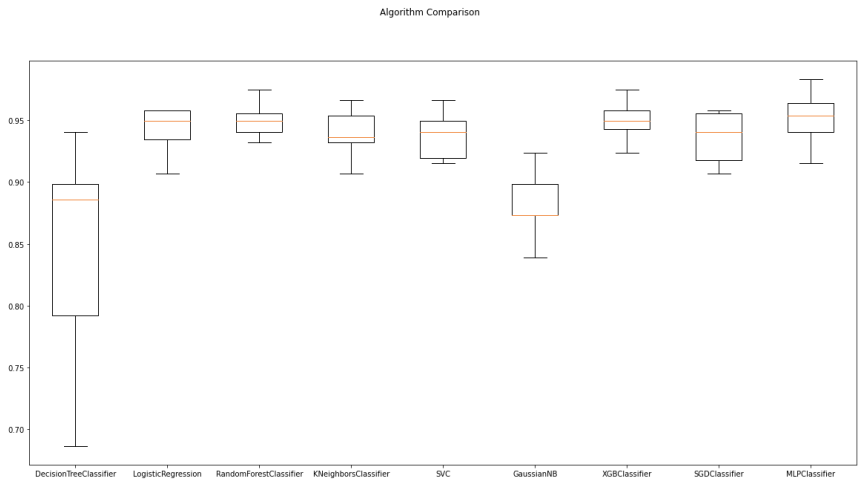


MLPClassifier

	precision	recall	f1-score	support
0	0.98	0.97	0.97	161
1	0.96	0.98	0.97	135
accuracy			0.97	296
macro avg	0.97	0.97	0.97	296
weighted avg	0.97	0.97	0.97	296

accuracy_score_MLPClassifier: 0.972972972972973

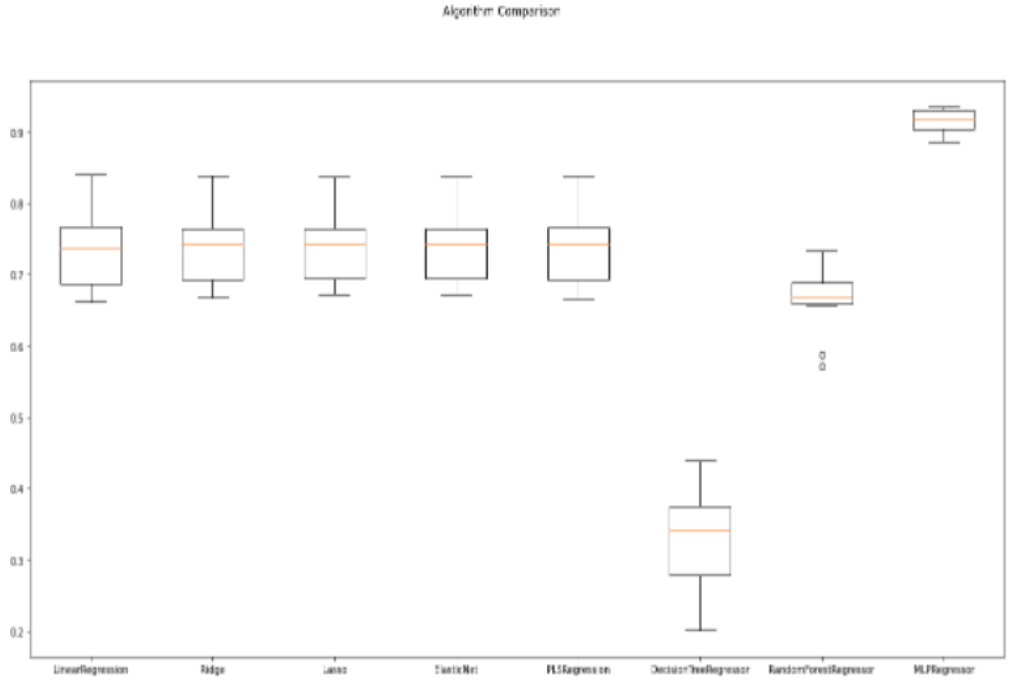




	name_model	mean_score_model	std_score_model	accuracy_score
0	DecisionTreeClassifier	0.849153	0.079661	0.695946
1	LogisticRegression	0.943220	0.016542	0.956081
2	RandomForestClassifier	0.949153	0.011370	0.935811
3	KNeighborsClassifier	0.938983	0.016864	0.952703
4	SVC	0.938136	0.017388	0.956081
5	GaussianNB	0.879661	0.026203	0.898649
6	XGBClassifier	0.949153	0.015626	0.945946
7	SGDClassifier	0.936441	0.018663	0.949324
8	MLPClassifier	0.952542	0.018644	0.972973

Fig. 7: all accuracy models with tuning parameters check with a box plot, confusion matrix, and data frame.

while regression The best prediction for CE802_P3_Data was multi-layer perceptron Regression with the highest score of 0.916145 percent, the lowest score fluctuation at 0.016271 percent, the lowest root mean square error at 388.6276, while the highest R2_score is 0.8806818(Fig. 12).



	name_model	mean_score_model	std_score_model	root_mean_square_error	r2_score
0	LinearRegression	0.733770	0.054211	4.024104e+14	-1.279317e+23
1	Ridge	0.736909	0.052443	5.756135e+02	7.382411e-01
2	Lasso	0.737715	0.051311	5.776387e+02	7.363959e-01
3	ElasticNet	0.737717	0.051307	5.768921e+02	7.370769e-01
4	PLSRegression	0.736817	0.052802	5.751054e+02	7.387029e-01
5	DecisionTreeRegressor	0.328891	0.069027	8.046842e+02	4.884471e-01
6	RandomForestRegressor	0.663438	0.047496	6.776767e+02	6.371855e-01
7	MLPRegressor	0.916145	0.016271	3.886276e+02	8.806818e-01

Fig. 12: Virtualize box plot and dataframe to check score accuracy, root mean square error, and R-Squared with tuning parameter

2.8. Feature Important

After we get the model the best We have to virtualize feature importance to tell how important a feature is to a model compared to other features from Fig. 8 in file CE802_P2_Data It can be seen that the most important features are F2,F5,F6,F7,F8,F13,F15 and the important feature, let's go down to F1,F3,F4,F9,f10. and the less important features are F11,F12,F14.

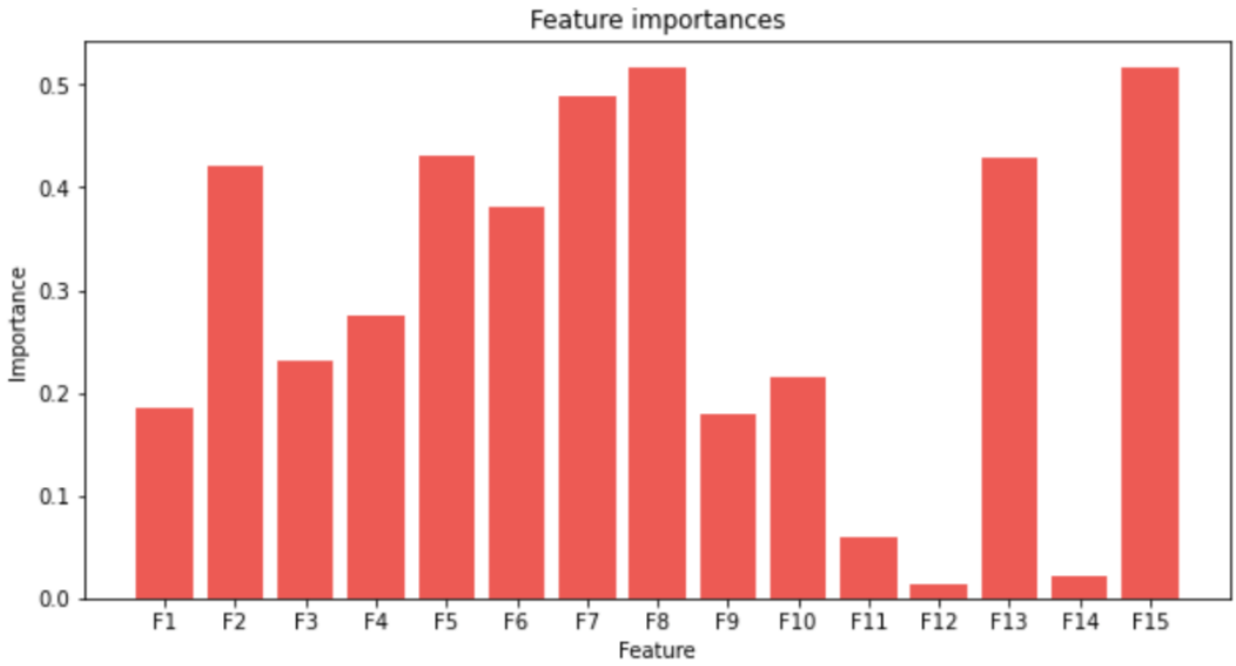


Fig. 8: virtualize Feature Importances with CE802_P2_Data file.

Whereas the CE802_P3_Data file predicting regression can be seen from Fig. 13 that the most important features are F2, F4 and the important feature, let's try to come down. F14, F12, F7, F10 and less important features are F1,F3,F5,F6,F8,F9,F11,F13,F15,F16,F6_Europe,F6_Rest,F6_UK,F6_USA.

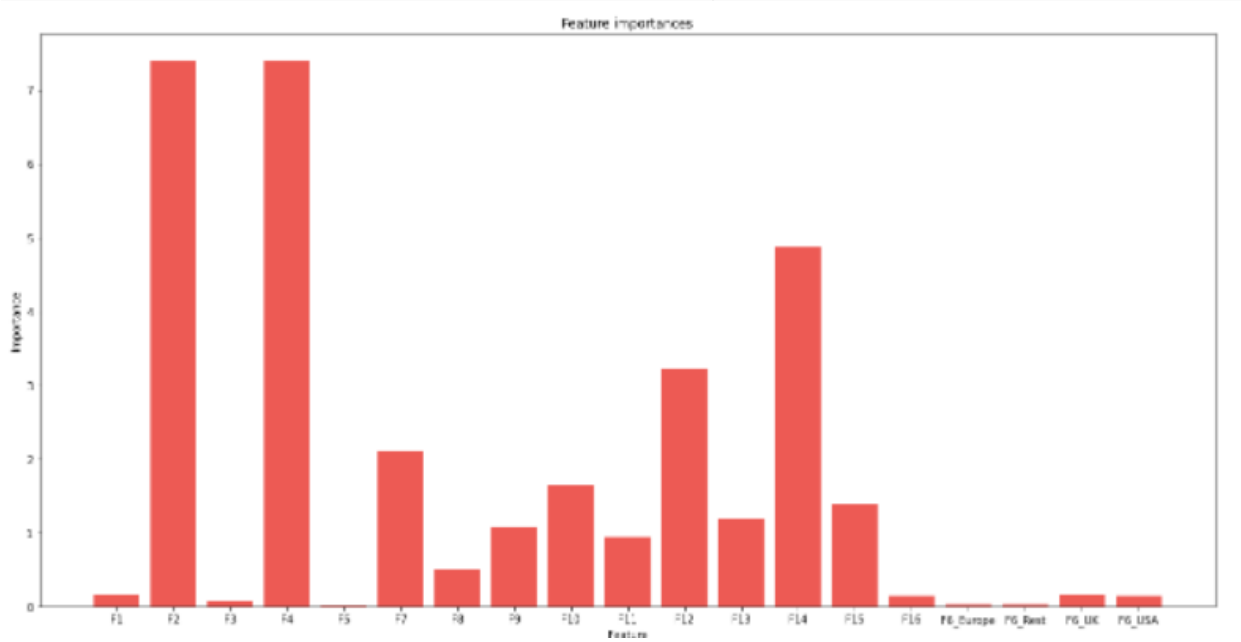


Fig. 13: virtualize Feature Importances with CE802_P3_Data file.

2.9.Predict all data in file

After we have selected the model and the tuning parameters, we need to predict them by converting the files CE802_P2_Test, CE802_P3_Test to a standard scaler, and then making predictions using the build-in function “predict”. In CE802_P2_Test, 0 is equal to False and 1 is equal to True, arranged in a dataframe as shown in Fig. 9. CE802_P3_Test does not need to be converted. it is like in fig. 14.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	Class
0	-4	765	-3.72	26.40	30	5.15	18.66	31.06	-4.80	15.18	-5.05	-6.56	-766.22	-1.98	-4.670000	False
1	-3	3	-7.10	1.20	3	0.36	-119.34	20.00	6.39	19.92	-5.29	0.32	-524.22	2.12	-7.076347	False
2	-4	735	-1.70	14.50	30	6.50	183.66	37.36	-11.88	11.22	-7.79	-9.64	-776.22	-3.88	-6.570000	False
3	-14	90	-2.55	13.30	30	7.95	-236.34	32.06	-3.09	15.86	-4.33	-8.76	-916.22	-3.41	-7.076347	False
4	-1	174	-7.63	3.74	3	0.94	-143.34	19.58	11.67	19.62	-7.63	1.12	-506.22	3.08	-7.076347	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	-3	81	-7.20	2.60	3	0.97	-125.34	19.32	4.80	19.44	-6.93	0.10	-498.22	1.22	-7.076347	False
1496	1	57	-6.95	2.88	3	0.04	-65.34	20.64	6.87	19.74	-7.63	-0.34	-480.22	2.03	-7.076347	True
1497	-1	63	-6.93	1.30	3	0.66	-125.34	20.06	4.50	22.68	-5.60	7.82	-518.22	2.28	-7.340000	True
1498	-14	75	-3.16	12.90	30	7.40	-221.34	32.56	-6.60	22.30	-4.03	-7.50	-546.22	-3.40	-8.810000	True
1499	-19	285	-5.77	10.00	30	7.60	-116.34	31.76	-9.81	10.44	-4.80	-9.24	-756.22	-0.44	-8.260000	True

1500 rows × 16 columns

Fig. 9: virtualization result predict by dataframe in CE802_P2_Test

	F1	F2	F3	F4	F5	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	Target	F6_Europe	F6_Rest	F6_UK	F6_USA
0	11.23	-195.54	-1.19	1468.56	4	8.97	-23.62	-249.36	-854.18	-155.20	12	10	12.39	-3480.87	0.04	936.095897	0	0	0	1
1	14.89	-426.24	-1.18	3049.08	4	6.33	-39.26	-226.26	-2126.68	-159.42	9	8	5.19	8831.19	43.68	1610.812822	0	0	0	1
2	6.76	-493.47	-13.55	3197.13	3	1.77	-25.84	-238.30	-2270.78	-212.73	12	10	3.30	-4468.44	0.52	481.941994	0	0	0	1
3	15.12	-320.04	-12.17	2436.00	3	5.42	-17.32	-203.64	-304.24	-100.34	18	12	6.51	22851.60	758.54	855.440616	0	1	0	0
4	10.12	-387.99	-7.11	2800.89	3	1.39	-12.78	-265.16	-1419.76	-137.49	0	14	14.22	24396.09	0.68	-40.543822	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	6.04	-308.37	-12.30	4346.82	2	8.96	-18.28	-266.14	-3132.76	-91.62	9	8	15.63	-1231.92	8.32	2267.126444	0	0	0	1
1496	14.13	-280.62	-6.00	2600.13	4	2.70	-0.80	-101.86	-1305.60	-147.74	18	18	13.23	7516.71	0.78	50.979168	1	0	0	0
1497	13.03	-301.05	-5.22	2279.07	3	0.23	-17.06	-168.30	-1413.06	-102.67	0	14	8.58	9689.01	4.80	-40.543825	0	1	0	0
1498	3.29	-347.40	-7.19	1985.31	3	2.63	-22.20	-299.66	-1183.62	-118.93	6	8	6.21	-2832.51	0.46	15.631153	1	0	0	0
1499	1.96	-456.06	-7.08	2821.95	0	7.75	-14.40	-176.46	-1754.88	-210.39	15	8	3.30	5500.77	0.00	569.841713	0	0	1	0

1500 rows × 20 columns

Fig. 14: virtualization result predict by dataframe in CE802_P3_Test

2.10. Export File

Export the file to use the data obtained from the prediction for further use.

3. Conclusion

It can be clearly seen that the default parameters are less accurate and more error-prone than tuning parameters. Both classification and regression are because the default parameters are made to fit every situation. But it's not the best prediction in each situation.

Multi-Layer Perceptron model both classification and regression are more accurate than All other models. because it is a deep learning model, but the disadvantage is that it takes so long to train that RandomizedSearchCV is required. instead of GridSearchCV According to many kinds of research, RandomizedSearchCV is faster and more accurate than GridSearchCV.

The Multi-Layer Perceptron regression model had the highest accuracy at 0.916145 percent and the lowest volatility in the model at 0.016271 percent while having very little data misperception with a root mean square error of 388.6276 and R. -Squared, the closer to 1, the better is 0.8806818, which is the closest.

while Multi-Layer Perceptron classification is considered the most accurate. In the classifier there are only 8 error values, divided into 5 false positives and 3 false negatives, resulting in precision. It is a measure of the accuracy of the data, and recall. is a measure of the accuracy of the Model, Accuracy It is a measure of the accuracy of the model that is higher than every model.

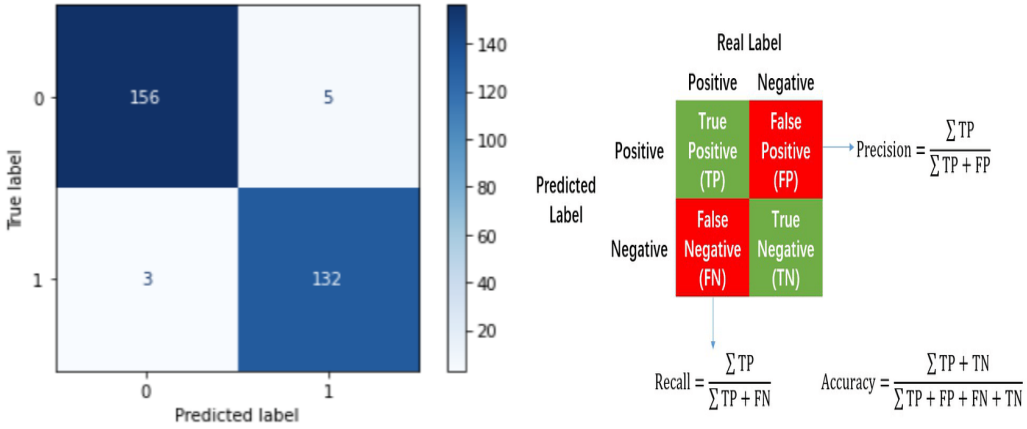


Fig. 15 the best model virtualize with confusion matrix[1]

4. Reference

[1]Jun M.,(2019). *Analyzing the Leading Causes of Traffic Fatalities Using XGBoost and Grid-Based Analysis: A City Management Perspective* <https://www.researchgate.net/figure/Calculation-of-Precision-Recall-and-Accuracy-in-the-confusion-matrix_fig3_336402347>

[2]Mayukh B.,(2019). *3 Things You Need To Know Before You Train-Test Split* <https://towardsdatascience.com/3-things-you-need-to-know-before-you-train-test-split-869dfabb7e50>

[3]Jason B.,(2019). *Hyperparameter Optimization With Random Search and Grid Search* <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>