2021

Machine Learning and Data Mining

A REPORT ON COMPARITIVE STUDIES (WORD COUNT: 1250)

STUDENT ID: 2006057

INTRODUCTION

We have requirement from 2 different travel insurance companies to build machine learning model for them.

PART-1

First company wants a modal that can predict whether an insured will claim or not. To work on this requirement, we have been provided historical data that contains categorical target “claim status” along with other information related to insured persons in the past. Claim status is equal to True if insured claimed else it is False.

I have built a classification modal for this task which I chose after doing comparative study among various other classification modal. Below is a summary on what all I did-

1. After loading the data file “CE802_P2_Data.csv” first I checked if the dataset is balanced or not and found that it was not highly imbalanced.
2. After that I described the dataset and found that F15 has half of the missing values. First, I dropped F15 and then trained the modal on remaining data. Second, I included the F15 and imputed the missing values using different approaches (Mean imputation and Model based imputation) and then again trained the models on this data.
3. Before training models, I divided the data in two splits in the ratio of 80:20. Used K-fold cross validation on train split to compare the scores of various models. Tested the best modal on test split as well.
4. Few columns were having slightly different scale from others, so I tried scaling the whole dataset.
5. I did hyperparameter tunning using Grid Search to get the best performance/parameter of all models.
6. Tried using deep learning modal (MLP) as well.
7. Lastly used MLP, which is the best modal as per my analysis, to predict the test dataset provided by the insurance company.

PART-2

Second company wants a modal that is not only predicting if an insured will claim but also the amount if s/he chose to claim. For this task too, we have historical data available, but this time target claim_amount is real valued feature. Target

denotes the claimed amount by insured in the past and it is equal to 0 if an insured has not claimed. I have built 2 models for this task. One for classification and another for regression. Classification modal will classify if an insured claim or not. Regression modal will predict how much will s/he claim. Note that regression will work only if the result of classification is True.

For classification task I have created a new target feature whose value is False if claim amount (Target) in the CE802_P3_Data.csv file is 0, otherwise True. The new target we got by this procedure was making dataset highly imbalanced. So, to check the performance of classification modal I have used f1-score.

Also, I have removed the claim amount target feature for classification task after creating the new target.

In regression, I have proceeded with almost same way as in I did in 1st part. Though the pre-processing and models used we different. Below is a small summary of steps-

Feature Selection, One hot encoding, Ordinal Encoding, Scaling, Hyperparameter tunning, Training MLP.

Eventually I have used MLP with logistic regression to predict the test set provided by the company.

Comparative Study:

Part 1: Classification Task (Insured will claim or not)

I have started off with very basic model without any prepossessing, scaling and hyperparameter tunning. Later, I will show how doing these steps will improve the performance of the model.

I loaded the data and found that F15 feature has half of the values missing. So, I chose to drop this feature and trained different models on remaining data. Below are the cross-validation scores that I received for all of them. You can see Random Forest and Gradient Boosting have given the better result than others.

cross-validation scores after dropping F-15

	Accuracy	F1-Score	Precision	Recall
DecisionTreeClassifier	0.787500	0.787277	0.787241	0.787500
RandomForestClassifier	0.834167	0.834051	0.834029	0.834167
GradientBoostingClassifier	0.844167	0.843785	0.844269	0.844167
LogisticRegression	0.721667	0.722138	0.724055	0.721667
SVC	0.606667	0.605495	0.619703	0.606667
KNeighborsClassifier	0.600000	0.598502	0.598208	0.600000

Then I thought of adding F15 feature as it could be an important in classification. I imputed its missing values, for which I used various technique

- I checked what is mean and median of F15. They both were almost same, so I imputed the missing values with mean.
- I trained a regression modal to predict the missing values for F15

With both techniques I got almost same result. You can see almost each algorithm has performed better than earlier except SVC and KNeighborsClassifier.

cross-validation scores after mean imputation for feature F-15

	Accuracy	F1-Score	Precision	Recall
DecisionTreeClassifier	0.800000	0.800055	0.800127	0.800000
RandomForestClassifier	0.868333	0.868230	0.868248	0.868333
GradientBoostingClassifier	0.851667	0.851287	0.851847	0.851667
LogisticRegression	0.757500	0.757928	0.760300	0.757500
SVC	0.606667	0.605495	0.619703	0.606667
KNeighborsClassifier	0.600833	0.599388	0.599087	0.600833

Then I tried scaling the attributes because there are few columns whose scale is slightly larger than other columns. Below is the result that I received after scaling. As you can see, Logistic Regression, SVC and KNeighborsClassifier scores have gone up.

cross-validation scores after standard scaling

	Accuracy	F1-Score	Precision	Recall
DecisionTreeClassifier	0.796667	0.796749	0.796870	0.796667
RandomForestClassifier	0.864167	0.864072	0.864074	0.864167
GradientBoostingClassifier	0.850000	0.849548	0.850322	0.850000
LogisticRegression	0.834167	0.834078	0.834047	0.834167
SVC	0.812500	0.812788	0.814159	0.812500
KNeighborsClassifier	0.754167	0.751760	0.755743	0.754167

There could be parameters, with which these algorithms could perform even better. So I did hyperparameter tunning and below is the result which I received. As you can see almost each algorithm has better score than earlier, but GradientBoostingClassifier is giving the best score

cross-validation scores after hyperparameter tunning

	Accuracy	F1-Score	Precision	Recall
DecisionTreeClassifier	0.800833	0.800901	0.800996	0.800833
RandomForestClassifier	0.869167	0.869075	0.869082	0.869167
GradientBoostingClassifier	0.880833	0.880641	0.880895	0.880833
SVC	0.869167	0.869053	0.869086	0.869167
LogisticRegression	0.873333	0.873276	0.873262	0.873333
KNeighborsClassifier	0.793333	0.792041	0.794279	0.793333

Performance on Test Dataset

Because GradientBoosting, LogisticRegression and RandomForest cross validation scores were high I thought of checking all of their performance on test split as well.

Score on Test Data Set

	Accuracy	F1-Score	Precision	Recall
RandomForestClassifier	0.853333	0.852489	0.856603	0.853333
GradientBoostingClassifier	0.886667	0.886132	0.889532	0.886667
LogisticRegression	0.856667	0.855593	0.861580	0.856667
MLP	0.903333	0.902828	0.906936	0.903333

Later on I also trained a MLP which worked slightly better than above algorithms I got more than 90% score with it on test split and 88.40% on validation split.

Conclusion:

Final Modal that I chose for this task is MLP because it has given almost same performance as GradientBoostingClassifier was giving on validation dataset, but on test set its performance is better than GB

Part 2: (Classification + Regression) if insured claims, then how much

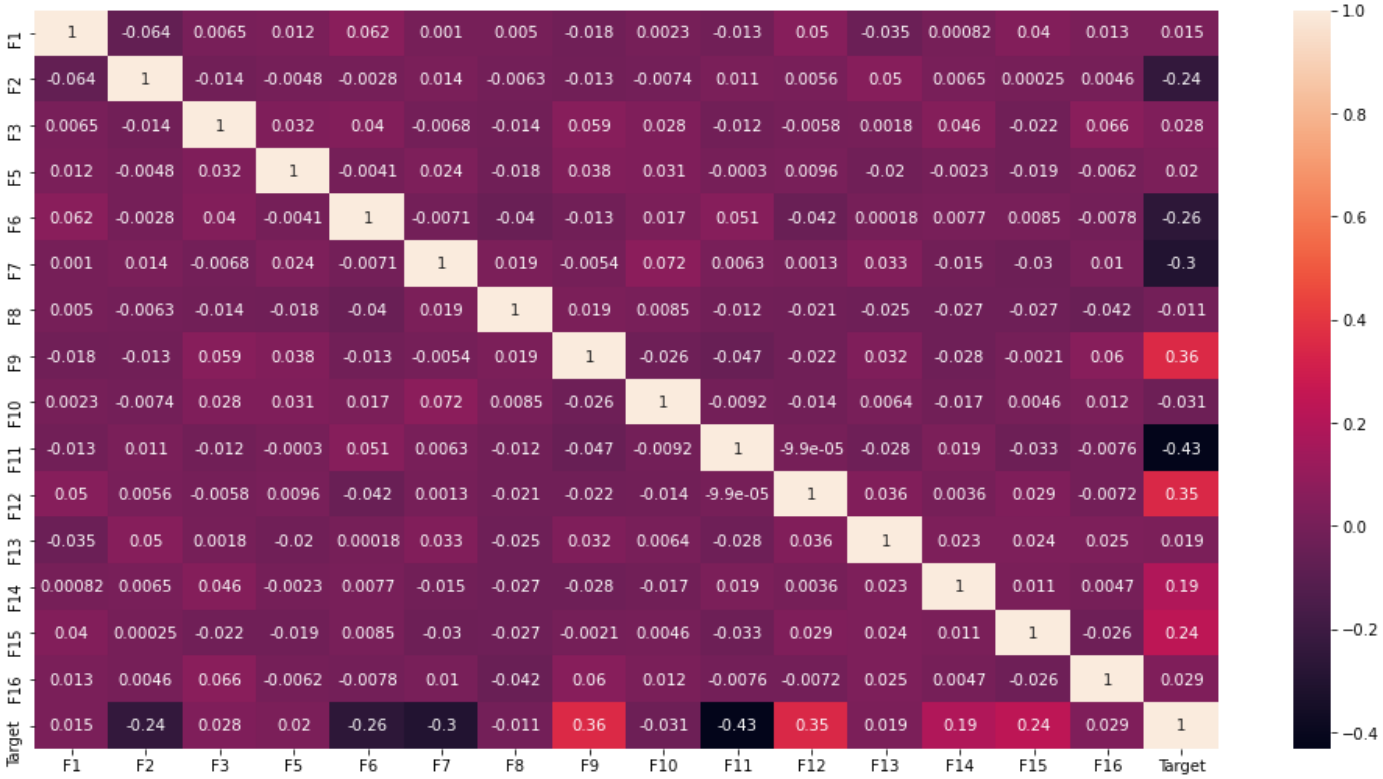
In this task too, I have started off with basic modal with just two pre-processing steps, which is to convert categorical column into numeric one and scaling numeric attributes. There are 2 columns in our data which are categorical F4 and F15. For F4 I have used one hot encoding as categories are not large in numbers and they also don't have any ordinal pattern. For F15 I have used ordinal mapping as there is an ordinal relationship. Below are cross validation scores that I received afterward.

cross-validation scores

	RMSE	R2
LinearRegression	513.862275	0.775991
Ridge	512.795857	0.776920
Lasso	511.877806	0.777718
SVR	1176.769129	-0.174774
GradientBoostingRegressor	483.203249	0.801924
RandomForestRegressor	615.979536	0.678112

There could be some features that more relevant in predicting the target over others, so I thought to perform a check on that. I plotted a correlation graph and found that below features were more related to target in comparisons to others. So, I just chose these features to train the model.

'F2','F6','F7','F9','F11','F12','F14','F15' have strong correlation.



As you can see score has slightly improved for all the algorithms except GradientBoostingRegressor

cross-validation scores after selecting 'F2','F4','F6','F7','F9','F11','F12','F14','F15'

	RMSE	R2
LinearRegression	507.425041	0.781568
Ridge	506.532250	0.782336
Lasso	506.565917	0.782307
SVR	1168.619869	-0.158559
GradientBoostingRegressor	483.131958	0.801983
RandomForestRegressor	555.766835	0.737966

Then I tuned all of these algorithms and check if the performance can be improved further. Indeed, performance for GradientBoostingRegressor and SVR has got increased. GradientBoostingRegressor is best so far. Refer below scores-

cross-validation scores after hyperparameter tunning

	RMSE	**R2**
Ridge	506.638072	0.782245
Lasso	506.565917	0.782307
RandomForestRegressor	561.657658	0.732382
GradientBoostingRegressor	410.441842	0.857086
SVR	525.606852	0.765634

Then I trained a MLP for this task and found its performance was way better than other algorithm used so far. I checked its performance of train, val and test split with each it was performing the best. Below are scores of validation and test dataset.

Performance score of MLP

	RMSE	R2
MLP_Val_Dataset	111.786624	0.988311
MLP_Test_Datset	136.461928	0.985062

Here because the task was to first predict whether an insured will claim or not and if he/she claim then predict how much can he claim. So I also trained a

classification model which can predict if insured will claim. For this I created a new target variable whose value was False if claim amount in our data was 0, otherwise True.

I trained couple of classification model and found logistic regression performing the best with more than 99% f1-score.

```
scores = cross_val_score(DecisionTreeClassifier(), x, y_c, cv=10,scoring='f1') # df1 dataset contains standard scaled features
print('DecisionTreeClassifier=', np.mean(scores))

scores = cross_val_score(LogisticRegression(), x, y_c, cv=10, scoring='f1')
print('LogisticRegression=',np.mean(scores))

scores = cross_val_score(GradientBoostingClassifier(), x, y_c, cv=10, scoring='f1')
print('GradientBoostingClassifier=',np.mean(scores))
```

```
DecisionTreeClassifier= 0.8479769971864618
LogisticRegression= 0.9922575780448272
GradientBoostingClassifier= 0.9279454246144377
```

Then I used MLP model for predicting the claim amount only if Logistic model gave me True, otherwise I returned the claim amount to be 0. Below is the score that I received with this modal and this the highest score that I could achieve.

Performance score on test split using MLP and Logistic together

	RMSE	R2
Logistic_and_MLP on Test Dataset	133.835709	0.985631

Conclusion:

MLP with logistic regression, I chose as my final modal to predict the Test dataset provided. Note that I am using the selected features only to predict the values.