Dataverse Hack

Insurance Claim Prediction

Authors

VISHNU VARDHAN DADI

MANOJ KUMAR DARA

Date: 2022-11-13

Contents

List of Figures

1 Abstract

CarIns is a startup that provides insurance for cars. It is one of the best car insurance brands known for the highest claim settlement ratio. It was launched back in Oct 2020 and acquired its initial policyholders by providing a hassle-free claim process, instant policy issuance, and claim settlements at minimum coverages.

As it's a fast growing startup, the company would like to optimize the cost of the insurance by identifying the policyholders who are more likely to claim in the next 6 months. Now the company would like to use Data Science to identify the policyholders whose chances of filing a claim are high in the next 6 months. The company challenges the Data Science community to build a high-performance algorithm to predict if the policyholder will file a claim in the next 6 months or not based on the set of car and policy features.

2 Tasks to be performed

- We are provided with information on policyholders containing the attributes like policy tenure, age of the car, age of the car owner, population density of the city, make and model of the car, power, engine type, etc and the target variable indicating whether the policyholder files a claim in the next 6 months or not.

3 Packages used

Numpy,Pandas, Scikit Learn, Scipy, Seaborn

4 Data Sets

- The test data is used for this project from the following link `https://datahack.analyticsvidhya.com/contest/dataverse/download/test-file`
- The train data is used for this project from the following link `https://datahack.analyticsvidhya.com/contest/dataverse/download/train-file`

5 Steps to be done

- Load the datasets.
- Find the missing values.
- Check the correlation between the features by heatmap.
- Differentiate the Numerical and categorical data types.
- One hot encoding for categorical datatype.
- Standardize the numerical values between -1 to 1.
- Use PCA method to reduce the number of dimensions.
- Checking the outliers with the Z-score.
- Before fitting the features to the model, check the data types of all the features.
- Apply the classification method in accordance with the pair plot, and experiment with the hyperparameters.
- Algorithms used:
  - Decision Tree Classifier
  - Random Forest Classifier (Best result)
  - SVM
  - xGBoost

6 Results

6.1 Best Model Sample

```
# create random forest model
from sklearn.ensemble import RandomForestClassifier

# hyperparameter tuning using grid search
# from sklearn.model_selection import GridSearchCV

# param_grid = {
#     'n_estimators': [100, 200, 300, 400, 500],
#     'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None]
# }

# grid_search = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 2, n_jobs = -1, verbose = 2)
# grid_search.fit(pp_train_df.drop('is_claim', axis=1), pp_train_df['is_claim'])
# # find best parameters
# grid_search.best_params_
# hyperparameters are found using grid search and are used to train the model
rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=123, class_weight='balanced')
rf.fit(pp_train_df.drop(['is_claim'], axis=1), pp_train_df['is_claim'])

# create submission file
submission_df = create_submission_file(rf, pp_test_df, policy_id_list, 'submission.csv')
submission_df.is_claim.value_counts()
```

Figure 6.1: **Sample model**

- F1 Score for the model achieved **0.1643424**