CS 675 - Machine Learning, Summer, 2021
Professor Ioannis Koutis

Review Others' work - HR Analytics: Job change of Data Scientists

Reviewing the work of
https://www.kaggle.com/mangotreefish/ds-job-change
Data Available in Kaggle.

What is the Objective of the project?
HR Analytics: Job change of Data Scientists project is to predict if a candidate will work for the company or will move to a new job. The objective of the project is to predict the probability of a candidate to look for a new job or will work for the company.

Classification models used:
Logistic Regression, K-nearest neighbors, Decision Tree and Random forest Classifier.

Steps performed:

- Importing necessary libraries and data.

There are two separate files given, train and test data set.

- Cleaning the data including Exploratory Data Analysis (EDA)

Univariate analysis of each continuous and categorical variable separately, cleaned it, performed EDA, visualized data and prepared it for modeling. Categorical missing values,kept them and used for the analysis because there is a lot of valuable information in missing data and deleting it could change the character of the data set. Numeric variables, missing data is filled in with random samples of that column because there is a very small number of missing values.

- Model preparation

Assign the features like experience, male,Graduate, etc as X, dummy variables for categorical variables and the target that is 0 – Not looking for job change, 1 – Looking for a job change as y. Then we split the data to our training and validation sets.

- Establishing Baseline

To understand if the models hold any weight, establish a baseline model to test models against. The ratio of number of records available for each target and total number of records is computed. Since the all negative model i.e. target=0 has a higher accuracy 75.03%, using that as a baseline. That means that model must beat an accuracy score of 75.03%.

- Model Selection

A function is defined to efficiently train each model on the training data and makes predictions for our validation set.

- Model Fitting

Each model is fitted using a function and the ROC curve for each model is plotted. Hyperparameter used for each model are as follows:
Logistic Regression(max_iter=1000), K-nearest neighbors(n_neighbors=50), Decision Tree(criterion='entropy', max_depth=5, max_leaf_nodes=10), Random forest Classifier(n_estimators=100).

- Predicting and finding the accuracy for each model.

Classifier	Accuracy Score	AUC Score
Logistic Regression	77%	0.5964870886445817
K-nearest neighbors	75%	0.5030253113700281
Random forest Classifier	78%	0.6692081522334636
Decision Tree	79%	0.7134438803462508

Conclusion

Decision tree out-performed the baseline model accuracy by 4%, logistic regression accuracy by 2% and random forest classifier by 3%. This means that if we want to predict which candidates would be most likely to change jobs based on our features, we would use the Decision Tree Model.

Critiquing

Not sure how the baseline is useful. Since the data is imbalanced there is a % difference between target = 0 and target = 1 number of data. If the data is balanced there is a 50% target = 0 and target = 1 number of data. So in that case, would any model performing better than 50% be a good model.

The AUC score for DT is higher than other models, the better the performance of the model at distinguishing between the positive and negative classes. So Decision Tree is the best choice in this case. Also the objective of the project is to predict the probability of a candidate to look for a new job or will work for the company. So this file seems not to have the submission file necessary for testing.

Since here DT seems to be the best approach,
https://www.kaggle.com/josephchan524/hranalytics-lightgbm-classifier-auc-80/comments#Predictions-for-aug_test.csv

- LightGBM classifier is used. Light GBM is a fast, distributed, high-performance gradient boosting framework based on a decision tree algorithm, used for ranking, classification and many other machine learning tasks.
- Stratified data split is performed.
- Label encoding is the approach used here instead of one-hot encoding.
- Hyperparameter tuning seems to be a challenging task. There are algorithms like Bayesian optimization for Hyperparameter optimization. Bayesian optimization methods are efficient because they select hyperparameters in an informed manner. In Grid Search you try all the possible hyperparameter combinations within some ranges.
- Feature Importance can be identified using Light GBM plot_importance or Shap package.

Classifier	AUC Score
Light GBM	80%

Submitted by
Parvathy Neelakanta Sarma (pn29)