FINAL PROJECT – AI PRACTITIONER
BANKING DATASET – MARKETING TARGETS COMPARISON OF MULTIPLE CLASSIFIERS TO DETECT POTENTIAL CUSTOMERS FOR TERM DEPOSIT CAMPAIGN

Ngô Đại Phương – Liam Ngo

Friday, August 28th, 2020

COTAI - VTC Academy, HCMC, Vietnam

Scope and Agenda

- Scope:
  - Use different classification methods to predict more accurately how many customers can be targeted for the next marketing campaign of banking term deposit sales
- Agenda:
  1. Overview
  2. Problem and target statement
  3. How to proceed
  4. Application

Overview

- “Term deposits are a major source of income for a bank.
  A term deposit is a cash investment held at a financial institution. Your money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing.
  Telephonic marketing campaigns still remain one of the most effective way to reach out to people. However, they require huge investment as large call centers are hired to actually execute these campaigns. Hence, it is crucial to identify the customers most likely to convert beforehand so that they can be specifically targeted via call.”

Problem and Target Statement

1. Problem statement: Imbalanced dataset could cause overfitting and inaccurate prediction. We can not rely on Accuracy as a metric for imbalanced dataset (will be usually high and misleading) so we would use Confusion Matrix, Balanced Accuracy, Geometric Mean, Precision / Recall, F1 score instead.
2. Target statement: select the best classification method which also can avoid overfitting
3. **Target achievement: RUS Boost has the highest Balanced Accuracy, Geometric Mean, Recall, F1 score among all classifiers the best performance on Confusion Matrix. 'Duration' is the most important feature. This banking dataset is not significantly overfitting as scores on Train and Test sets of multiple classification methods are minorly different.**

How to proceed

1. Use different models such as Single Decision Tree, Ensemble Classifiers (Bagging, Balanced Bagging, Random Forest, Balanced Random Forest, Easy Ensemble Classifier, RUS Boost), XGBoost, Deep Neural Network to evaluate their performances on both imbalanced Train and Test set while avoid fitting.
2. Different metrics such as Accuracy (we do not rely on this one), Balanced Accuracy, Geometric Mean, Precision, Recall, F1 score, Confusion Matrix will be calculated.
3. Find the most important features of this dataset which can be used in policies to predict number of ‘Not Subscribed’ or ‘Subscribed’ customers after applying those new changes.
4.

Application

Agenda

1. TEPFA
2. Library
3. Plan

TEPFA

- **Task:**
  - **Input:** Banking's Imbalanced Dataset – Marketing Targets' features
  - **Output:**
    - Draw importance chart of most influential features
    - Draw tree branches (only for decision tree)
    - Subscribe or Not Subscribe: Compare performance results between different classifiers
    - New policy to know which changes within which features may help to subscribe potential customers more precisely.
- **Experience:** imbalanced dataset
- **Performance measure:** Balanced accuracy, Geometric mean, Accuracy, Precision, Recall, F1-score, Mean ROC AUC, Confusion Matrix
- **Function space:** Different Ensemble Classifiers, XGBoost, DNN
- **Algorithm:** Ensemble, Gradient Boosting, Deep Neural Network

```
kernel_initializer='glorot_normal'
```

Library and model in Python

1. Library
    1. Keras
    2. Seaborn
    3. Matplotlib
    4. Pandas
    5. Imblearn
2. Model:
    1. Decision Tree
    2. Bagging
    3. Balanced Bagging
    4. Random Forest
    5. Balanced Random Forest
    6. Easy Ensemble
    7. RUS Boost
    8. XGBoost
    9. Deep Neural Network

Further actions

- Apply Label Encoding instead of One Hot Encoding to fit Ensemble Classifiers better
- Use Graphviz to draw tree, H2O to create models (of tree, forest) from beginning instead of Scikit-learn
- Use other methods: Oversampling, SMOTE, etc to avoid Overfitting more efficiently
- Practice more coding
- Review previous theories, algorithms
- Read new materials

- Links:
- https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/ensemble/plot_comparison_ensemble_classifier.html
- https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets
- https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/applications/porto_seguro_keras_under_sampling.html?highlight=porto
- https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
- https://codelabs.developers.google.com/codelabs/fraud-detection-ai-explanations/index.html?index=..%2F..index#4