Automate the Process of Fitting and Summarizing Supervised Machine Learning - Classification Models on the Data

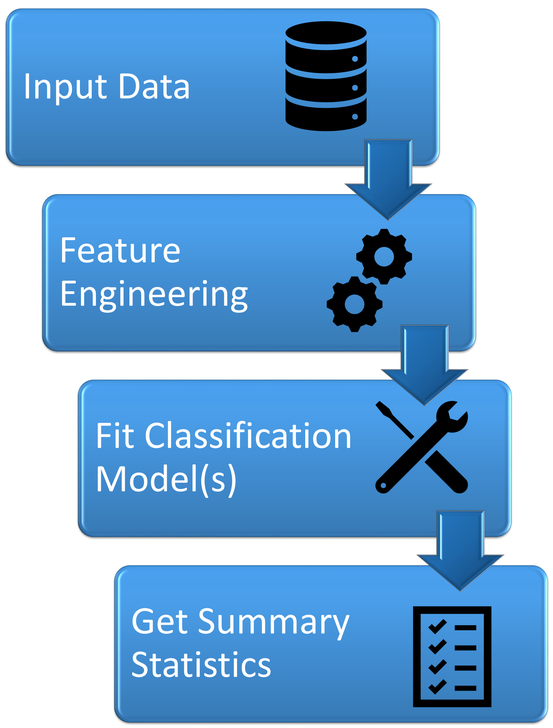
Mehul Patel
Northeastern University

INTRODUCTION

While leveraging the data, analysts usually spend a lot of time in summarizing the results of analysis to answer a business-question. So, a lot of time is spent in brainstorming the ideas about '*how to present analysis in a clear-concise manner?*'. Here, we attempt to facilitate the process of summarizing the analysis by providing highly intuitive summary statistics of some of the widely used supervised machine learning classification techniques. We have built a Python module which takes partially-processed data as input, fits the classification methods, and summarizes the results very clearly. Thus, this module is an attempt to automate the process of summarizing analysis, and to save a significant amount of human-efforts.

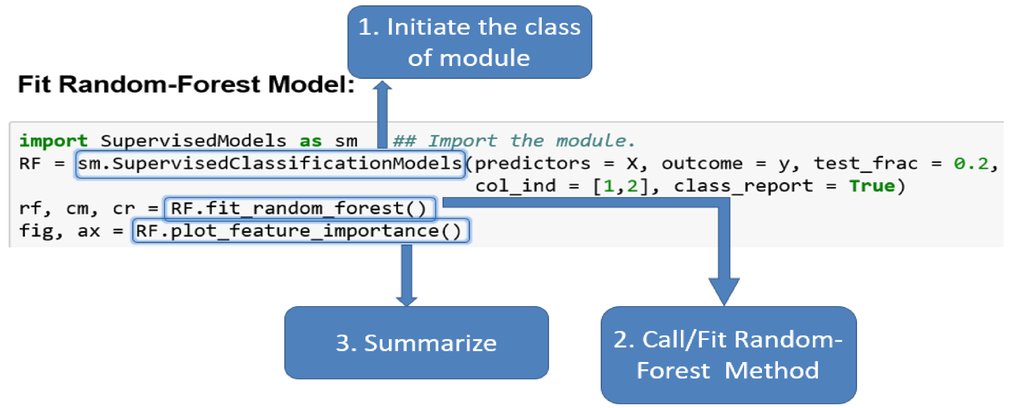
IMPLEMENTATION

Python Implementation of Module



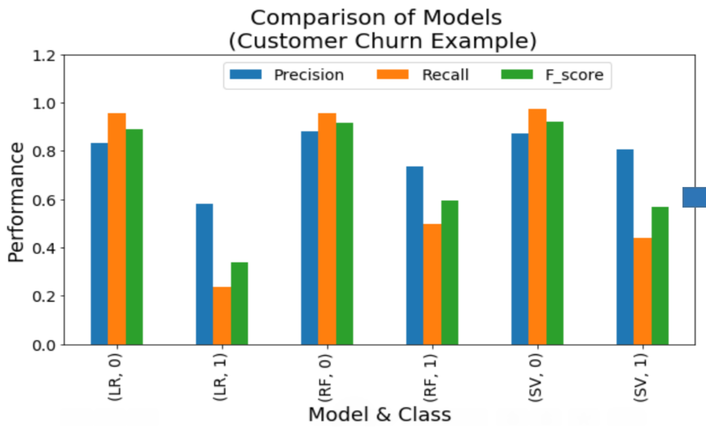
USAGE

Following code-snippet shows 'how to use the proposed module'. As one can observe, it is just a 3-step process to get summary of any classification model, in this case: Random-Forest. These steps are common for all the classification models in the module, namely Logistic-Regression (LR), Random-Forest (RF), and Support-Vector-Classifier (SV).

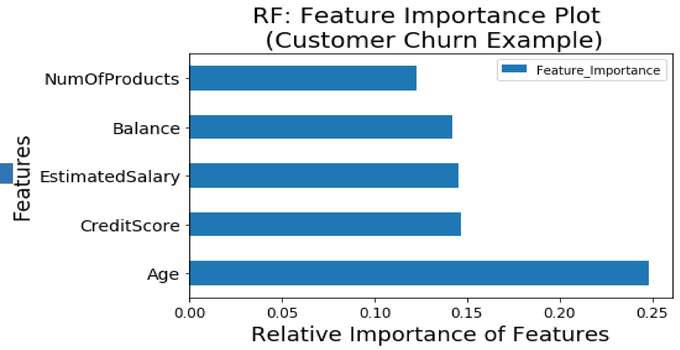


SUMMARY PLOTS

This section shows a couple of the summary plots from the module for classification models.



This summary plot compares performance of all 3 models for binary classification (Class-0/1) problem.



Module also provides summaries for individual model. Example: Random-Forest.

CONCLUSION

- This Python-module speeds-up the process of summarizing the results of several classification techniques as it just requires a couple of lines of code.
- The quick-summary-plots give a clear idea of models' performance.
- Any organization, industry, or an individual can utilize this module when it comes to fitting models and summarizing results very quickly.

FUTURE SCOPE

- Add various functionalities, in terms of parameter-tuning of models, incorporate more classification models, ensemble models etc.
- Include automation for data preprocessing such as dealing with missing values, feature transformations etc.
- Optimize the code for speed and ease-of-use
- Previous attempts to do similar thing are either commercialized or not entirely open-source. We hope to make this project entirely open-source.

ACKNOWLEDGEMENTS

- The data-set used to test the module is available on: https://goo.gl/5N46kq. We are thankful to them for making the data-set available.
- We are thankful to Dr. Md. Noor-E-Alam, Assistant Professor at Northeastern University, for reviewing our work for this project.

CONTACT

Author: Mehul Patel
Email: patel.mehu@husky.neu.edu