ixisiv

1. Set environment, import all necessary libraries
2. Data Ingestion¶

2.1. Import data into Python Pandas DataFrame

3. Explorative Data Analysis¶

3.1. - check missing and duplicated values

3.1.1. No missing and duplicated values

3.2. Deal with numeric and categorical features separately

3.2.1. Take out numeric features

3.2.2. Take out categorical features

3.3. Check data distributions

3.3.1. Check the distribution of the response variable by its value counts in barplot The "yes" to "no" ratio is roughly 1:8, which will be handled in the later classification processing

3.3.2 Check the distributions of numeric features by histograms¶

3.3.3 Check the distributions of Categorical features by value counts in barplots¶

4. Filtering Feature Selection¶

4.1 Filtering Feature Selection on numeric features by 2-sample t-tests For every individual numeric feature, make a t-test between the two separate groups, one group contains values with the label y = "yes", and the other group contains values with the label y = "no"

4.2 Filtering Feature Selection on categorical features by Chi-square tests¶ For every individual categorical feature, make a Chi-square test by its cross-table with the label y

4.3 Filter in the significant numeric and categorical features Set alpha level to be 0.05 and use Bonferroni method to adjust multiple tests

5. Feature Engineering¶

5.1 Map "yes"/"no" to 1/0 for binary variables Binay variables will be converted to 1/0 to aviod redendancy, since if use one-hot enconding, the two new features will be perfectly correlated

5.2 Convert categorical variable to numerical by one-hot encoding¶

6. Classification Modeling¶

6.1 Prpare training and testing data by random splitting 80% data for training; 20% data for testing

6.2 Use SMOTE to resample the TRAIN DATA ONLY¶ Leave the test data AS-IS to best represent the "Reality"

6.3 Define multiple classification algorithms Including Random Forest, LASSO, Support Vector Machine, k-Nearst Neighbor. All chosen algorithms have the "class_weight" parameter to address the imbalance of label y. Set the "class_weight" parameter to be "balanced" will bring the imbalanced training data into balanced situation. Put the defined algorithms and their names into 2 corresponding lists

6.4 Make a function to perform training, testing and evaluation It displays confusion matrix, and returns the f1-score, precision, recall, accuracy as well as the trained classifier itself.

6.5 Make function to scan all defined classification algorithm For each algorithm, train a classifier on training data and test/evaluate it by the test data Returen all trained classifier and their evaluation results

6.5.1 Run the function to loop all defined algorithms to check performance 6.5.2 Display the aggregated evaluation result of all classifiers Based on the F-1 score, ElasticNet, with which L1 Ratio set to 0.6, has the best performance

6.6 Feature importance analysis

6.6.1 Feature importance analysis for LASSO

6.6.1.1 Use coefficients to indicate the importance of corresponding features

6.6.1.2 Plot all features by their coefficients¶

6.6.1.3 Pick up features with |coefficients| >100

6.6.1.4.Plot features with |coefficients| >100

6.6.2 Analyze feature importance for the Random Forest classifier

7. Conclusion

7.1. The ElasticNet is the best model based on the training/testing and evaluation by F-1 Score

7.2. The "housing" feature is the most importance feature in ElasticNet classifier

7.3. The "duration" feature is the most importance feature in Random Forest classifier

======== The main analysis/modeling has finished, thank you! ========

8. Additional Tries¶

8.1 Additional Try No. 1: Try Recursive Feature Elimination (RFE) Slightly improved LASSO, but NO improvement on Random Forest

8.1.1 Use the Random Forest classifier to do RFE feature selection Didnot use ElasticNet because it is an embedded feature selection which will be redundant

8.1.2 The best peformed classifier is a little bit better than what before RFE Decided to keep using ElasticNet as the final model

8.1.3 The feature importance (coefficient) on the LASSO model after RFE

8.2 Additional Try No. 2: Try AutoML using H2O NO Improvement according to the F-1 Score

8.3. Additional Try No. 3: Try deep neural network through tensorflow/keras No improvement Converged to precision: 0.1247; recall: 1.0000