Write-up for Machine Learning Homework 3

Tommy Yu

- **Improvements on Homework 2:**
  - For x_dist, adopted violin plot instead of box plot, and applied to the middle 99% of the data to get rid of the outliers.
  - Made sure that labels and titles are complete for all charts.
  - Added methods of filling missing data: mode, median, particular value.
  - For creating categorical variables, made use of pandas' get_dummies function.
  - Most importantly, rewrote codes for classifiers and metrics based on Rayid's magic loop.
    - Modified classifiers and parameters to test
    - Recreate a simpler version of plot precision recall curve
    - Added metrics: AUC of PR curve, recall, f1
    - Tracked running time of each loop

- **From Homework 2 – summary of findings from data description:**
  - Those who are in financial distress are on average 7 years younger, earn $1,000 less per month, and support 0.2 more dependents than those who are not.
  - It is substantially more likely for those in financial distress to have their bill payments past due.
  - They also have a lower debt ratio, fewer open loans & credit lines, and lower credit balance on average.
  - Those with over 8 dependents are never in financial distress.
  - The more often "past due" occurs, the more likely it's the case for a person in financial distress.

- **Summaries and new findings based on the loop:**
  - Features adopted: (n=11)
    RevolvingUtilizationOfUnsecuredLines, Age, DebtRatio, Quartile of MonthlyIncome, NumberOfOpenCreditLinesAndLoans, NumberOfTimes90DaysLate, NumberRealEstateLoansOrLines, NumberOfDependents

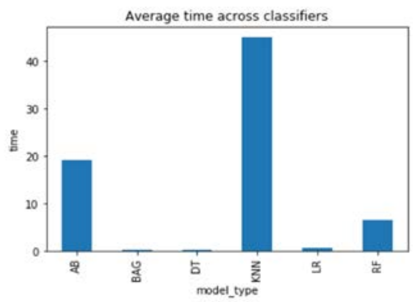
  - Best parameters among the tested for each model according to accuracy; if multiple selections share the same level of accuracy, the one takes the least time to execute is chosen:

    - Random Forest:
      {'max_depth': 50, 'max_features': 'sqrt', 'min_samples_split': 10, 'n_estimators': 100}
    - Boosting:
      {'algorithm': 'SAMME', 'n_estimators': 100}
    - KNN:
      {'algorithm': 'auto', 'n_neighbors': 50, 'weights': 'uniform'}
    - Logit:
      {'C': 0.01, 'penalty': 'l1'}
    - Decision Tree:
      {'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'min_samples_split': 2}

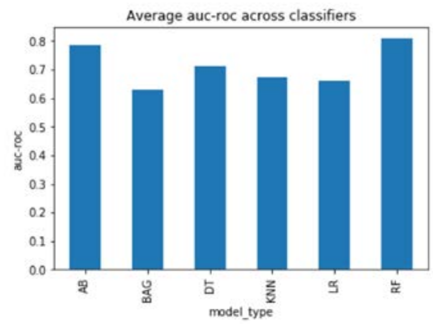
- Bagging:
  {'max_features': 5, 'max_samples': 5, 'n_estimators': 1}
- SVM:
  N/A

o Comparison of metrics across classifiers:

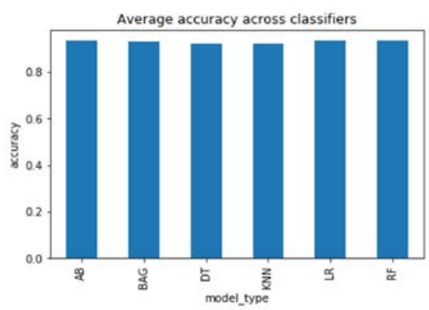
- Time:



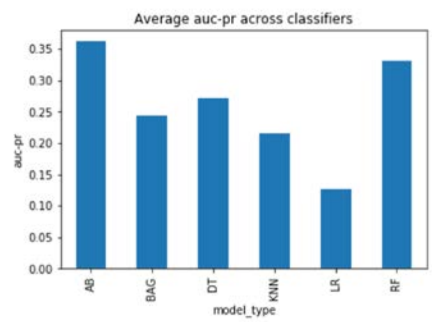
- AUC of ROC



- Accuracy:



- AUC of PR-curve



o Takeaway & Recommendations
- SVM takes a great amount of time (over 20 minutes to execute one try). One explanation is that there is not enough margin to fit a (n - 1) hyperplane between the two classes of dependent variable. “Linear” may not be a suitable kernel. Whether the credit data fit the assumptions of SVM classifier should be further studied before running another try.
- KNN is also a very slow option, mainly because it is very slow at scoring/prediction time.
- Ensemble methods – random forest and boosting – have better performance in terms of AUC of ROC and AUC of PR-curve (aka average precision score).
- AUC of ROC and AUC of PR yield similar results, that is to say class imbalance is less of an issue here.
- Random forest is very efficient on large data.
- Accuracy remains on a similarly high level for all classifiers, thus should not be used as a key metrics to differentiate models here.
- It requires much additional work to justify why the parameters listed above result in higher accuracy for a certain classifier.