1 Machine Learning : Demographics Prediction for Online Customer data by Lesara

Random Forests are trained with random sample of data (even more randomized cases available like feature randomization) and it trusts randomization to have better generalization performance on out of train set.Ensemble learning methods tend to show better performance for classification. SVM models perform better on sparse data than does trees in general.Builds Support vector machines and determine best fitting line.

1.1 The Algorithm: Random forest

- The data in non linear
- The data is not highly sparsed
- large data (overfit less likely with RF)
- compute time (can parallelize RF)
- imbalanced data (can stratify sampling in RF)
- Random Forests are better to strain on overfitting

1.2 Metrics

To evaluate model we need to select two things a Validation scheme and metrics:

- validation scheme
  - k-fold validation
  - validation with train/validation
- metric for regression: MAE, RMSE, MSE
- metric for classification:AUC, Accuracy
  - Compute Area Under the Curve (AUC) using the trapezoidal rule

1.3 Evaluation of trained model

Following are the results acquired by modeling. Random forest with 1000 trees. Gives accuracy score of 0.78. With very less Mean square error and acurracy of 0.70. In principle a good machine learning model performing well on training set should also perform on similar test set.

Table 1.1: Monte Carlo Simulation for Average

Forest Size	Accuracy Score	MSE	AUC
1000	0.788	0.211	0.703
500	0.789	0.210	0.705
100	0.791	0.203	0.701
50	0.786	0.213	0.7004

As we can see , the precision and AUC changes with number of estimators, representing over fitting when estimators are too high.

1.4 Pros for Using random forest

- Random Forest are applicable to a wide variety of modeling tasks, they work well for regression tasks, work very well for classification tasks(and even produce decently calibrated probability scores)
- Random Forests can be easily grown in parallel.
- Since random forests are not very sensitive to the specific hyper-parameters used, they don't require a lot of tweaking and fiddling to get a decent model.
- Random Forests perform implicit feature selection and provide a pretty good indicator of feature importance.

1.5 Cons for Using random forest

- The main drawback of Random Forests is the model size. You could easily end up with a forest that takes hundreds of megabytes of memory and is slow to evaluate.
- Another point that some might find a concern is that random forest models are black boxes that are very hard to interpret.
- Over fitting can occur

1.6 Performance and Scalability

- To compute a machine learning model for a very large data set is unfeasible. But still we need to classify users quickly.For that purpose the Machine learning models are persisted as binary files called Pickled files using joblib from sklearn.externals or available in Model persistence as pickle.dump.

- Exposing you machine learning model as a web service is a good idea. In python flask library can be used to fulfill the requirement.
- To answer large number of requests coming , it is good idea to use multi-threading over your web service.

1.7 Improvements

- Accuracy of model can be increased, by fine tuning the model, for example changing the criteria , enabling boot strapping, njobs, out-of-bag cross validation etc.
- 10- fold cross validation can be used to improve the performance of the model
- Optimal number of estimator can be found by plotting steps curves . Instead of randomly experimenting with estimators.