Capstone Proposal
Numer.ai: binary classification on financial market data

Domain Background

A long standing problem in financial markets have been pricing of assets and price prediction into the future.[1] The efficient market hypothesis states that all new information about a stock is immediately reflected in the price.[1] Numerai is trying to get the wisdom of the crowds at the same time as not giving out information about the market.[2] [3] Numerai is a hedge fund that does this by doing two things. Getting the public to make predictions and uses homomorphic encryption. By using homomorphic encryption the data has the same structure but it is not possible to know what the source data actually looks like.[4] Which means that information cannot leak back to the market. I find this task particularly interesting because it tests the raw data science and machine learning skills up against other data scientists.

Problem Statement

The dataset is homomorphic encrypted. This means that any new information from prediction can only be understood by Numerai I.e. we do not know if this is time series or not, or what the data represents. Only that Numerai has encrypted information about the market and transformed it into a machine learning classification task. The data is a set of features where the values range from 0 to 1. And the goal is to classify the rows 0 or 1, i.e a binary classification task.[2] The submission format is the probability that the class of the row is 1 or 0. By providing the prediction you are immediately scored against a holdout set and get rewarded based on your accuracy.[2]

The predictions for the test set is scored and a score is given after submitting online on the numer.ai website, scored against a hidden holdout set. In addition you receive a ranking of how much your prediction improve the meta model created from all models submitted. Along with the data a benchmark is given based on logistic regression on the data. Also score should be higher than -log(0.5) in order to show that you do better than chance.

Datasets and Inputs

The dataset is comprised of 50 features and one target feature.[5] The 50 features are float values in range from 0 to 1. The target is binary 1 or 0. The task is a binary classification task. A new dataset is given out every week where the leaderboard is reset and payment given out to the top 100 predictions.

Solution Statement

The solution is to use pre-processing and feature engineering along with classifiers to create a classifier that can classify the test data. The solution is scored by doing cross validation in

addition to submitting the results online. It should be compared to the baseline logistic regressor provided by numer.ai and score better than –log(0.5). The goal should be to get in the money, meaning becoming one of the top 100 models. I plan to do this is several steps.

1. Explore the data.
2. Process data.
3. Create model.
4. Evaluate model.
5. Go to step one.

Where the steps are iterative. For pre processing I plan to do standard normalisation and check the distribution and balance of classes. Also I am planning to consider dimensionality reduction to reduce the number of features and visualising in 2D. Using PCA and t-sne. And in some cases clustering of the reduced feature to get additional features.

For modelling the most common classifiers used in competitions like these are xgboost, GBM, Extremely random trees and k-nearest[6]. But I will also use a simple neural net as deep learning is proven to be versatile and possibly relevant for this competition. In addition to a logistic classifier.

In the end I intend to combine these model by stacking them together.[7]

Benchmark Model

In financial predictions doing better than chance means that you can stand to win money (if you disregard the cost around trading and possible slippages in timing). The model should therefore do better than chance which is > -log(0.5). Numer.ai has also provided a baseline classifier, which is a log loss classifier with no preprocessing. Any solution should also beat this baseline. The ultimate goal is to be among the top 100, to get paid.

Evaluation Metrics

As the classification is to be binary, where the model provides probability of a binary class the evaluation metric of choice is log loss. This is also the evaluation metric used by numerai. The second metric should be the position on the leaderboard.

Project Design

The project would follow the following process.

1. Explore the data.
2. Process data.
3. Create model.
4. Evaluate model.

1. Explore the data

Visualise the data.
Do descriptive statistics on the data.
Evaluate the different features, importance, correlation.
Evaluate the difference between the test and train dataset.

Consider the balance of classes and distribution of the dataset.

2. Do feature engineering by normalising the data and using PCA, and T-SNE with different parameters. PCA has number of dimensions and T-SNE has a set of parameters that can be varied.

3. Use data preprocessed in the different ways. I.e raw, PCA with different parameters, T-SNE with different parameters, and run the following models. Extremely random trees, GBM, XGboost, k-nearest, logistic classification and a neural net in keras. All the models have varying hyperparameters.

4. Evaluate the different models.
Before evaluating the models I randomly select 10% of the data as holdout.

Using hyperparameter search find the best model and use the results combined in stacking the models. The models should be separately evaluated using stratified kfold validation [8][7]

When doing data preprocessing and model prediction the data is always stored between steps, in order to avoid re running the models.

The optimised metamodel created in stacking should be evaluated compared to the holdout set.

I am considering doing hyper parameter optimalization, where I change the different parameters and check the performance, but this could take a long time, as the time grows exponentially with the number of parameters. [9]

Finally, I submit the prediction to numer.ai and check the results and compare with baseline and results in training to consider over fit.

References

1. https://en.wikipedia.org/wiki/Stock_market_prediction
2. https://numer.ai/rules
3. https://medium.com/numerai/invisible-super-intelligence-for-the-stock-market-3c64b57b244c#.ia4y8pocu
4. https://en.wikipedia.org/wiki/Homomorphic_encryption
5. https://numer.ai/rules
6. http://mlwave.com/kaggle-ensembling-guide/
7. https://github.com/ikki407/stacking
8. http://scikit-learn.org/0.15/modules/generated/sklearn.cross_validation.StratifiedKFold.html
9. https://github.com/hyperopt/hyperopt