Predicting Bond Liquidity Using Ridge Regression

- Senior_Undergra1 (GS-Quantify Team)

Idea

- Given

  Three month history of trading of bonds starting 16th March and ending 9th June, 2016.

- To Predict

  Buying and selling volumes for immediate next 3 business days, i.e., 10th(**Fri**), 13th(**Mon**) and 14th(**Tue**) June, 2016.

- Assumption/Strategy

  The nature of trading of bonds on weekend(Friday) and early week days(Monday, Tuesday) follows a pattern over the weeks.

Approach

- Training Input

  Train regression model using one month history and static characteristics data.

- Training Output

  Buying and selling volumes of each bond for immediate next three business days which are Fri, Mon, and Tue.

- Move the 1 month window by 1 week for 8 weeks(15th April till 3rd June 2016) - **Training data size ~ 8 x (No. of bonds)**

Implementation

- **Python 2.7.12**
- **SciKit Learn** - Machine Learning Library
  - Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
- **Pandas** - Library for Data Manipulation & Analysis

Steps :

Feature Extraction

Feature Selection

Regression Model

Feature Extraction

From the given raw data:

- Get features in proper formats/datatypes
- Pass the features to a feature-selection alogorithm.

Static Features:

- Features ignored -
  1. *issuer*
- Features extracted -
  - ➔ **Real -** *amtIssued, amtOutstanding*, *coupon, couponFrequency*
  - ➔ **Categorical to Integers -** *Market, paymentRank, ratingAgency1Rating, ratingAgency1Rating*
    - ◆ ASSUMPTION - A linear order of importance in the categories of these four characteristic variables.
  - ➔ **Categorical to dummy encoding -** All the remaining categorical static variables were coded as binary features, one for each possible category.

Feature Extraction

➔ **Datetype features -**

For *maturity, raingAgency1EffectiveDate, ratingAgency2EffectiveDate,* the difference in no. of months between *maturity* data and the last date of recorded three month history(9th June, 2016) was taken.

Historical Price & Volume Features:

- For each bond in a one month training window, the following features were extracted:
    - Sum of buyVolume
    - Sum of sellVolume
    - Standard deviation in price
    - Mean price

  REFERENCE: http://www.finra.org/investors/alerts/bond-liquidity-factors-questions

- *time* and *timeofday* were ignored.

Missing Values

- For missing values in *couponFrequency*, **mode** of the available coupon frequencies was used.

- For missing values in *maturity*, *ratingAgency1EffectiveDate*, *ratingAgency2EffectiveDate*, the **mean** number of months of difference between these dates and the last date of recorded history(9th June, 2016) was used.

- For bonds which did not appear in the last one month window, the **sum** of buyVolume and sellVolume, the mean of price and the standard deviation in price across the entire three month window was used to generate input test data for predicting the final ouput for submission.

Feature Selection

Criterion :

- P-values for the predictors

  In regression, low p-values indicate terms that are statistically significant.

  REFERENCE:

  http://blog.minitab.com/blog/adventures-in-statistics/how-to-interpret-regression-analysis-results-p-values-and-coefficients

- `Sklearn.feature_selectiof_regression` was used to get the p-values for the extracted features.
- The features with P-values greater than 0.001 were removed from the training input.

Regression Model

- We used **Ridge Regression** predicting the Bond Liquidity in the terms of buy and sell volume.
- Ridge regression addresses overfitting problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares

$$\min_{w} ||Xw - y||_2^2 + \alpha ||w||_2^2$$

- $\alpha \geq 0$ is a complexity parameter that controls the amount of shrinkage
- The highest score(0.651981) was achieved with **alpha = 0.1,** after testing in the range of [0.01, 10.00].
- The outputs of buyVolume and sellVolume were rounded off to the nearest 10,000 before submission.

Unimplemented Ideas

- Including ***count*** of total transactions for a bond in one month training window.
- Utilizing the '*issue date*' feature in the model in terms of difference with '*maturity*' or with the last day of the 3 month recorded history.
- More testing with regression models like Elastic Net Regression.