IAQF Annual Academic Competition: Solution by Ensemble Modeling

Abstract

In this research we attempt to bridge the connection between economic, financial factors and credit spread on US investment-grade corporate bonds, and make meaningful predictions of the credit spread movement through machine learning techniques. Our final ensemble model predicts the movement direction of credit spread with an accuracy of almost 70% out-of-sample.

1 Introduction

In this research we attempt to bridge the connection between economic, financial factors and credit spread on US investment-grade corporate bonds, and make meaningful predictions of the credit spread movement through machine learning techniques.

The corporate credit spread are closely correlated with a wide range of factors. Using default risk and interest rate risk as factors, Longstaff and Schwartz (1995)'s Two-factor model was one of the earliest attempts to value fixed and floating rate debt and credit spread. Collin-Dufresn et al. (2001) studied credit spreads on individual bond yields and found that traditional models of default risk could only explain one fourth of the credit spreads variation, while the rest should be explained by systematic factors. From an 85 year perspective, Davies (2008) found regime switching techniques could help explain credit spreads movements and checked the idea on high grade and low grade credit spreads separately. Thanks to the recent development of computer science, machine learning techniques have been widely used in financial predictions (Bose and Mahapatra (2001)). For instance, Luo et al. (2017) applied deep learning algorithms on credit scoring by using credit default swaps data set. However, few researches focused on using machine learning tools and techniques to study credit spreads. In this research, we decide to find the determinants of credit spreads and make forecast on the direction of credit spreads using machine learning techniques.

We start with a broad spectrum of data sources and narrow down our final feature list by carefully examining the inherent connections between the indicator and corporate credit spread. Variance inflation factor is applied to reduce multi-collinearity among the selected features as well as their lagged terms. We examine feature importance through algorithms such as recursive feature elimination and random forest, where only the high ranking features are retained.

After feature selections, we fit our train dataset through over ten selected machine learning algorithms with cross-validations, and use the in-sample prediction results to determine the weights of each algorithm within the final ensemble model. The whole process is totally engineer-wise replicable and is expect to shed some light on the general process of economic factor forecasting in a machine learning perspective.

2 Data

This section provides documentation on relevant data sources and selected features.

2.1 Public Dataset

The data for our analysis primarily comes from the Economic Research at the St. Louis Federal Reserve Bank, including the bond yields, treasury rates, and government interest payment data. The market data, such as S&P 500 and VIX quotes, come from Yahoo Fiance and Quandl. Unless otherwise stated, all the rest professional datasets, e.g. historical default rates of one specific company, are retrieved from Bloomberg.

2.2 Features

Our feature ideas mainly come from previous researches and our innovation.

Table 1: Basic Features.

Abbreviation	**Feature Name**
`bd`	BofAML US Corporate AAA Yield
`tr10`	US 10-Year Treasury Bond
`spr`	S&P 500 Monthly Return
`vix`	The CBOE Volatility Index
`dr`	Moody's High-Yield Bond Default Rate
`d2g`	Debt/GDP Ratio
`i2g`	Federal Gov't Interest Payments/GDP ratio
`cs`	Credit Spread
`ecs`	Credit Spread, Estimated
`dcs`	Credit Spread, One Period Diff.
`edcs`	Credit Spread, Estimated One Period Diff.
`syc`	Yield Curve Slope
`unem`	Unemployment Rate
`dunem`	Unempoyment Rate, One Period Diff.

Credit Spread. The definition of credit spread in this research is the difference between BofAML US Corporate AAA Effective Yield and 10-year Treasury Bond rate.

Estimated Credit Spread. We estimate future credit spread using spot default rates and risk-free rates: $\mathrm{E[CS]} = (r_f + \mathrm{DR})/(1 - \mathrm{DR}) - r_f$.

Stock Return. Collin-Dufresn et al. (2001) summarized the conclusions of previous researches that bonds of higher grade are similar to Treasury bonds, while bonds of lower grade are more sensitive to stock market. Norden and Weber (2009) found that stock returns lead bond spread changes by applying a vector autoregressive model. In this research we use monthly returns of S&P 500 as a measure of stock market return.

Treasury Rate. The level of treasury rate can also affect the size the spread, as shown by Collin-Dufresn et al. (2001) that in an environment where risk-free rate is higher, the credit spreads of all bonds tend to be lower. Collin-Dufresn et al. (2001) explains that an increase in drift theoretically decreases the risk-neutral probability of defaults.

VIX. VIX as a measure of future volatility expectation has been proved in numerous researches, as shown in Collin-Dufresn et al. (2001), Guo and Newton (2013), Chun et al. (2014) and etc., that it has significant correlation with credit spread. Thus, we add VIX into our features.

Default Rate. In the research of Collin-Dufresn et al. (2001), it was found that traditional default rate models can explain one-fourth of the variation of credit spread. Another interesting fact is that credit spread is much higher than expected default loss, Amato and Remolona (2003) explained that it is due to the undiversified character of default risk. In this research we use Moody's high-yield bond default rate as a measure of default risk.

Debt-to-GDP. Min (1999) fount that strong macroeconomic fundamentals like low inflation rate have association with low yield spread, while weak liquidity indicators like high Debt-to-GDP ratio are associate with high yield spread, so we can take Debt-to-GDP into account in our model.

Interest Payments/GDP ratio. Due to the correlation between Debt-to-GDP ratio and credit spread, we reason that the ratio between Interest Payment and GDP could also be helpful for predicting credit spread. Hence it is added into our features.

Yield Curve Slope. According to Litterman and Scheinkman (1991), the increase of Yield Curve Slope would lead to the decrease of credit spread. We use the difference between 10-year Treasury rate and 2-year Treasury as as a measure of yield curve slop.

Unemployment Rate. We believe that unemployment rate is indicative of both the overall health of economy and the stability of corporations.

3 Methods

This section illustrates how we tackle the problems by incorporating different algorithms on different stages of the entire project. First, based on the features we already gathered, we expanded the feature list by generating lagged and polynomial terms. Then, we reduce the size of our features pool, by first using Variance Inflation Factor test, and then further reduce the list by ranking the importance of individual variables through different means.

Once we are settled with the 15 features we decide to proceed with, we run 10 different classifiers, and tunes the models with GridSearchCV, which gives us a list of the 10 models with the most effective parameters. Finally, we use Stacking to ensemble all the base models, and produced a new meta-classifier which in turn gives us decent result, which will be discussed at the end of this section.

3.1 Feature Engineering

Starting from the 14 features, `bd`, `tr10`, `spr`, `vix`, `dr`, `d2g`, `i2g`, `cs`, `ecs`, `dcs`, `edcs`, `syc`, `unem`, `dunem` selected above, we first add lagged values up to 3 days, which gives 56 features in total. Then squares of these 56 features are added, giving 112 features in total. We use different algorithms to reduce the number of features down to 15, based on the correlation and importance of these features. We use a trailing `_L` to denote the lag values and a trailing `_2` to denote the squared values. And we use a binary variable `y`, which is our target class, to denote the upward or downward movement of the credit spread one day ahead.

VIF. After expanding the feature lists to 112 with lagged and squared features, we suppose that multicollinearity will be a big problem, especially considering autocorrelation within different factors. Thus, we choose to use VIF reduction method to deal with the problem in an effort to reduce the pool of the features to a more manageable size before we go to more sophisticated feature selections. During implementation, instead of calculating individual VIFs for different variables and arbitrarily dropping large ones, we choose to consider the the biggest VIF among all features to be the models VIF, and keep dropping the feature with the largest

VIF, until the model VIF fall below the threshold of 10. In this way, we try to keep more variables, by considering the fact that removing high VIF variables will at the same reduce the VIF of others. Finally, we ended up with 38 features, after dropping 74, and the max VIF of our model is 9.1671.

Random Forest. A Random Forest is an ensemble of Decision Trees. For a single tree, important features are likely to appear closer to the root of the tree, while unimportant ones will often appear closer to the leaves. Based on this idea, we calculate a features importance by averaging the depth at which it appears across all trees in the forest. The 15 most important features given by random forest are spr, spr_L1_2, spr_2, spr_L2_2, spr_L3, vix_L3_2, spr_L3_2, tr10_2, syc_L3_2, dr_2, ecs_L3_2, cs_L3_2, edcs_L1, edcs and edcs_L2.

χ^2. Chi-square test measures dependence between stochastic variables. We compute chi-squared statistics between each (non-negative) feature and our target class and select the 15 features with the highest values, eliminating the rest of the features which are the most likely to be independent of the class and therefore relevant for classification. The 15 features selected are dunem_L3_2, edcs_L2_2, spr_L3_2, dr_2, syc_L3_2, dunem_L2_2, dcs_L3_2, dunem_L1_2, dcs_L1_2, dcs_2, spr_2, unem_L3_2, edcs_2, vix_L3_2 and dunem_2.

RFE. We use Recursive Feature Elimination with Logistic Regression, pruning one feature with the lowest coefficient on each step, until we finally get 15 most significant features. The final 15 features left are spr, spr_L3, dcs_L2, dcs_L3, edcs_L1, edcs_L2, dr_2, spr_L2_2, spr_L3_2, ecs_L3_2, dcs_L3_2, edcs_L1_2, edcs_L2_2, syc_L3_2 and dunem_L3_2.

After three individual selection procedures, we sum the rankings in the three algorithms of each feature (for unranked ones, fill in 15) and sort the result ascendingly. We keep the 15 features with the lowest sum rankings, which are dcs_L2, dcs_L3, dcs_L3_2, dr_2, dunem_L3_2, edcs_L1, edcs_L2_2, spr, spr_2, spr_L1_2, spr_L2_2, spr_L3, spr_L3_2, syc_L3_2 and vix_L3_2.

3.2 Sampling

First, we see a big reason to re-sample our dataset. As shown in Figure 1, about 72% of our training set y values are classified as 1. As we use accuracy score as the benchmark to evaluate our models, we undergo a substantial risk of obtaining a highly overfitted model while achieving high predictive accuracy, because the outcome is fairly less sensitive to the error produced when predicting negative values.

When conducting samplings with time-series data, we try to be more careful with the means and algorithms we use. As most algorithms assume that observations are conditionally independent, we have reduce the extent to which this assumption is violated by dropping most of our features based on the VIF benchmark. On the other hand, in Chawla et al. (2002), Chawla points out with rich empirical evidence that SMOTE performs desirably with under-sampling algorithms. Therefore, we use SMOTETomek as the algorithm for resampling, which combines SMOTE with Tomek-Link, a well-known under-sampling technique. By using this ensemble sampling algorithm we implicit assume that the time structure, except for those captured by lagged variables, is missing in our data. This is actually a pivotal drawback and we'll come back to it by the end of Section 3.3.

As a result, we have balanced both of our training and testing dataset, with negative (denoted as 0) and positive (denoted as 1) data points evenly distributed, as shown in Figure 1. Zeros movements are trivial enough to be neglected in this case.

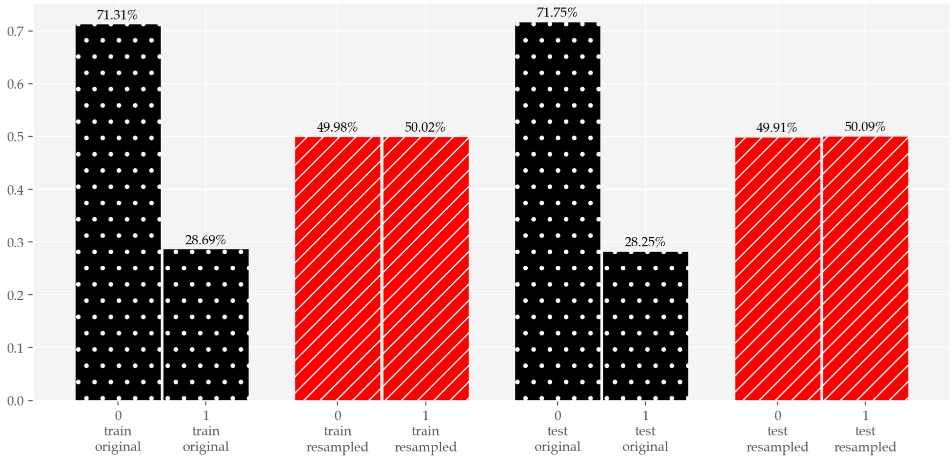

Figure 1: Target class distribution before and after sampling.

3.3 Individual Classifiers

Logistic Regression. Logistic Regression uses a linear model assigning a weight to each feature, and uses logistic function to convert the weighted average to a probability. If the predicted probability is larger than 0.5, we conclude the class to be positive.

Linear Discriminant Analysis (LDA). Linear Discriminant Analysis assumes that the conditional probability density functions $P(\mathbf{X} \mid y = 0)$ and $P(\mathbf{X} \mid y = 1)$ are both normally distributed. Its form is very similar with Logistic Regression which predicts a point as being positive ($y = 1$) if the dot product between the learned weights and input features vectors is above a certain threshold.

Support vector machine (SVM). Support vector machine tries to draw a $n-1$-dimensional hyperplane in a n-dimensional feature space to separate two classes with the widest "street" between boundaries. Though essentially SVM is still a linear model, because we have added polynomial (square) terms before, the hyperplane can be curved to model nonlinear separations.

K-Nearest-Neighbors (KNN). KNN is an instance-based machine learning algorithm, making predictions by finding the K nearest neighbors (measured by a norm) of a new instance in the training sample and giving probabilities according to frequencies of the K instances. The class with the highest probability is the final prediction.

Decision Tree (DT). Decision Tree makes predictions using a tree, with each split comparing one feature with a threshold. The minimization of cost function maximizes the purity of each node and leave. But if we do not set constraints on the usage of features and maximum depth, it is easily exposed to overfitting problems.

Random Forest (RF). Random Forest addresses the problem of overfitting in a single decision tree by ensembling decision trees, generally trained via the bagging method (or sometimes pasting). It introduces extra randomness by limiting features to a random subset to grow diversified trees, which trades a higher bias for a lower variance, generally yielding an overall better model.

Extra Trees (ET). Extra Trees is a forest of extremely random trees which also uses random thresholds for each feature when growing trees, thus trades more bias for a lower variance. It is even less prone to overfitting and is usually trained faster.

AdaBoost (ADA). AdaBoost is a boosting algorithm which iteratively trains predictors, with a new predictor correcting its predecessor by giving more weights to the training instances that the predecessor underfitted. The predicted class of AdaBoost algorithm is the one that receives the majority of weighted votes of the predictors.

Gradient Boosting (GB). Gradient Boosting is another boosting algorithm which instead of tweaking instance weights at every iteration like AdaBoost does, tries to fit the new predictor to the residual errors made by the previous predictor. The prediction is basically the sum of predictions given by all predictors.

XGBoost (XGB). XGBoost is a scalable (the most important factor of XGBoost) machine learning system for tree boosting brought up in Chen and Guestrin (2016), which is widely used by data scientists and provides state-of-the-art results on many problems.

The optimized hyper-parameters found using grid search with cross-validation and accuracy obtained with these optimized classifiers are listed in Table 2.

Table 2: Optimized hyper-parameters and accuracy of individual classifiers.

Algorithm	Optimized Hyper-Parameters	Accuracy
LR	`C:0.001, penalty:l1`	0.4997
LDA	`n_components:None, solver:svd`	0.2578
SVM	`C:10, gamma:1`	0.5905
KNN	`leaf_size:2, n_neighbors:5, p:2`	0.5062
DT	`criterion:entropy, max_depth:25, max_features:7`	0.6163
RF	`bootstrap:True, criterion:gini, max_depth:25, max_features:None, n_estimators:200`	0.6146
ET	`bootstrap:True, criterion:gini, max_depth:25, max_features:None, n_estimators:100`	0.5005
ADA	`algorithm:SAMME, learning_rate:0.001, n_estimators:100`	0.4683
GB	`learning_rate:0.01, max_depth:25, max_features:7, n_estimators:200`	0.5965
XGB	`gamma:0.1, learning_rate:0.01, max_depth:25, min_child_weight:1, n_estimators:200`	0.6133

LSTM. LSTM networks are a type of RNN that uses special units in addition to standard units. LSTM units include a "memory cell" that can maintain information in memory for long periods of time, and use a set of gates to control the flow of information; this architecture lets them learn longer-term dependencies. This vastly differs LSTM from the previous classifiers, which all can only handle stateless inputs and neglect all time structures in data. Instead of a classifier, here LSTM serves more similar to an ARIMA model, which unfortunately cannot describe nonlinear relationships. The training process of our LSTM model is shown in Figure 2. Specifically, considering the network structure, we used in total 2337 parameters and 4 hidden layers, respectively an LSTM input layer (2048 parameters), a dropout layer, a tanh activation layer (272 parameters) and finally a linear output layer (17 parameters).

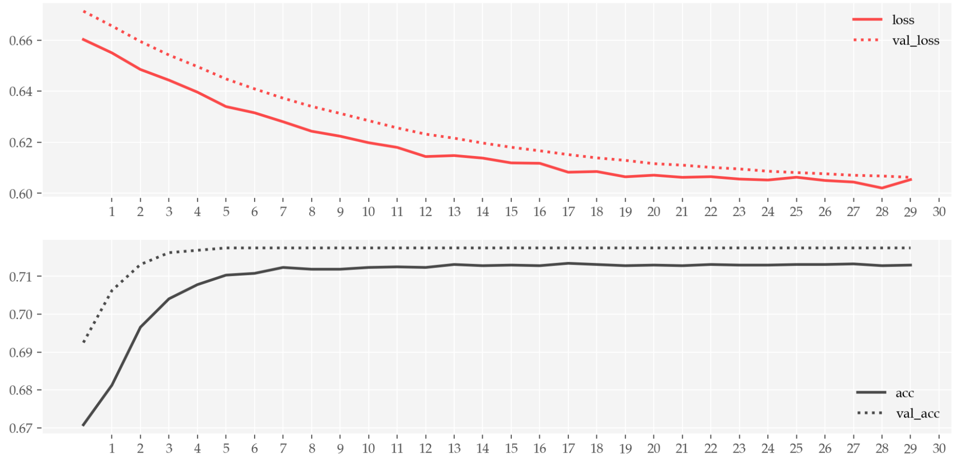

Figure 2: Training process of LSTM model.

There are several potentially important things to take care of whenever one adopts RNN algorithms like LSTM, in time series forecasting or classification. First, the huge amount of parameters in these RNN networks suggests, which is well awared of during our research, a high possibility of over-fitting, and this can be avoided by proper cross-validation. Second, it's critical to note that when running an LSTM algorithm, we use original training set without sampling because shuffling is not permitted in RNN. This directly leads to an unavoidable overfit which is very clear by comparing Figure 1 and the final accuracy. The essence of RNN is to capture the sequential manners in data and thus in fact, not only suffling is forbidden, neither is general cross-validation. Only piece-wise cross-validation is performed in our analysis.

3.4 Ensemble

Stacking. Finally, we use stacking in an effort to ensemble all the models we have as weak learners, and to achieve higher predictive accuracy. In our approach, we first predict the credit spread moves with all the tuned models we have, and generate a new dataset which includes all the predictions we have generated as independent value, while keeping the dependent values unchanged. Subsequently, we will train a Logistic regression as a meta-classifier based on the new dataset, which gives us the final prediction. The regression coefficients of this meta model are summarized below.

Table 3: Meta model. Scores from in-sample prediction accuracy (cross-validated). Coefficients from logistic regression of in-sample labels on in-sample predictions from all individual classifiers.

Model	**LR**	**DA**	**DT**	**KNN**	**RF**	**EXT**	**ADA**	**GB**	**XGB**	**SVM**	**LSTM**
Score	0.5	0.258	0.616	0.506	0.615	0.500	0.468	0.596	0.613	0.591	0.718
Coef	0.0	−0.778	1.509	−0.024	3.609	1.677	−0.991	4.477	2.522	0.306	0.000

As shown in Figure 3, we saw that the final ensembled model gives us a prediction result, which accurately predicts 84.7% of down moves, and 55.2% of the up moves. The corresponding accuracy is 0.6995, which means our model can successfully classify up and down moves upon the balanced testing dataset. The F1 score of our final model in the test set is 0.7381.

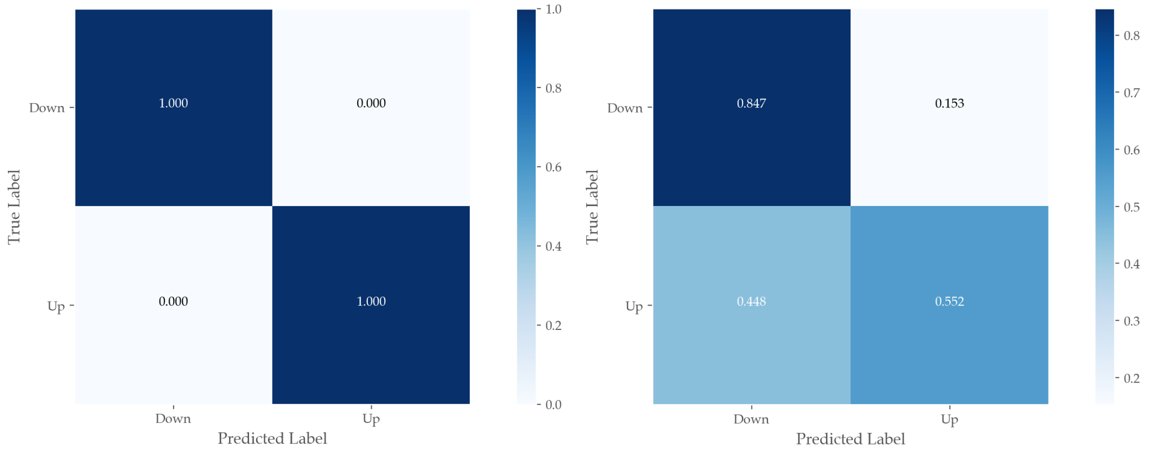

Figure 3: Confusion matrix. This is the final prediction result with our stacking model. The left subplot corresponds to in-sample predictions and the right is out-of-sample.

4 Conclusion

In this research, we have combined economic and finance insights together with state-of-the-art data mining techniques to numerically determine important factors that determine the direction of the next day's credit spread movement. The problem is well defined as a binary classification, which pertains to a much larger and sophisticated field of machine learning, regression analysis, etc. By proper ensemble modeling we eventually achieved significant predictability based on a balanced dataset. This implies an unnegligible economic vision, with which policy makers could make potentially better decisions concerning corporate capital behaviors and credit risk management.

In the meantime, more advanced sampling algorithms (e.g. over-sampling for sequential data) are expected, together with their joint performance with LSTM networks. This is a field beyond the reach of most sampling theorems. Also, more industry-based features should have been included in our research so that the model can capture more subtle movements. These relationships are unfortunately nowhere to be found but resorting to actual experience of years in the respective field.

References

Amato, Jeffery D and Eli M Remolona (2003). The credit spread puzzle. *BIS Quarterly Review, December*.

Bose, Indranil and Radha K Mahapatra (2001). Business data mininga machine learning perspective. *Information & management 39*(3), 211–225.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16*, 321357.

Chen, Tianqi and Carlos Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Chun, Olfa Maalaoui, Georges Dionne, and Pascal François (2014). Credit spread changes within switching regimes. *Journal of Banking & Finance 49*, 41–55.

Collin-Dufresn, Pierre, Robert S Goldstein, and J Spencer Martin (2001). The determinants of credit spread changes. *The Journal of Finance 56*(6), 2177–2207.

Davies, Andrew (2008). Credit spread determinants: An 85 year perspective. *Journal of Financial Markets 11*(2), 180–197.

Guo, Biao and David Newton (2013). Regime-dependent liquidity determinants of credit default swap spread changes. *Journal of Financial Research 36*(2), 279–298.

Litterman, Robert and Jose Scheinkman (1991). Common factors affecting bond returns. *Journal of fixed income 1*(1), 54–61.

Longstaff, Francis A and Eduardo S Schwartz (1995). A simple approach to valuing risky fixed and floating rate debt. *The Journal of Finance 50*(3), 789–819.

Luo, Cuicui, Desheng Wu, and Dexiang Wu (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence 65*, 465–470.

Min, Hong G (1999). *Determinants of emerging market bond spread: do economic fundamentals matter?* The World Bank.

Norden, Lars and Martin Weber (2009). The co-movement of credit default swap, bond and stock markets: An empirical analysis. *European financial management 15*(3), 529–562.