KPMG 4 First Progress Report

Zixiang Yin zy2444 Xuchen Wang xw2747 Zhenyu Yuan
zy2492 Yao Xiao yx2696 Kechengjie Zhu kz2407

October 2022

1 Introduction

This Capstone project is about conducting large scale financial time series forecasting on companies from a given stock index. The research objective is to make accurate predictions to important financial indicators for companies in the given index. The business goal is to help market participants and investors make better trading decisions and manage their portfolio more suitably.

2 Methodology

2.1 Problem Formulation and Modeling Approach

Our final goal of this project is to built a model with good performance in multi-variate financial time series forecasting. To achieve this goal, we start from the simplest model and try to scale up the problem.

In the first stage(current stage) we try to build univariate classic ARIMA models for each individual features.
For each variable x of each company
Feature: $(x_1, x_2, \dots x_t)$
Target: $(x_{t+1}, \dots x_{t+k})$

In the second stage, we would want to build a multivariate regression model with some classic ensemble learning models like randomforest or xgboost.
Let the multivariate feature be $\vec{x}_t = (x_{t1}, x_{t2}, \dots x_{tm}))$ at time t, for each feature x_m we need to build an individual model, that is, for the i_{th} dimension
Feature: $(\vec{x}_1, \vec{x}_2, \dots \vec{x}_t)$
Target: $x_{(t+1)i}$

In the final stage, we would like to explore the usage of deep learning,

and see if it can be more powerful than traditional approaches.
Feature: $(\vec{x}_1, \vec{x}_2, \dots \vec{x}_t)$
Target: $(\vec{x}_{t+1}, \vec{x}_{t+2}, \dots \vec{x}_{t+k})$

2.2 Related Literature

Currently there are two papers we would like to reference from.
1, A Multi-Faceted Approach to Large Scale Financial Forecasting
The paper has a very similar approach as we do, starting from a simple baseline and gradually scale up, which can serve as a nice guideline.

2, A Transformer-based Framework for Multivariate Time Series Representation Learning
Transformer is a powerful technique which is widely used in representation learning like NLP. We would like to explore its usage in structured data regression problem.

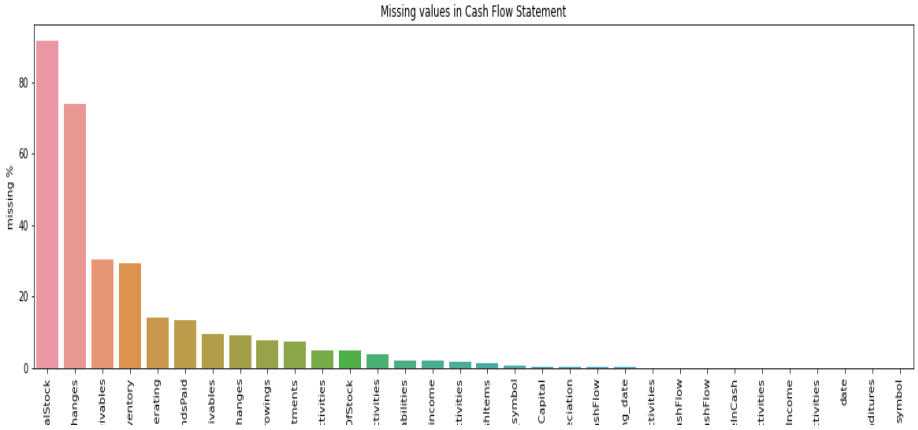
3 Results

3.1 Data Acquisition

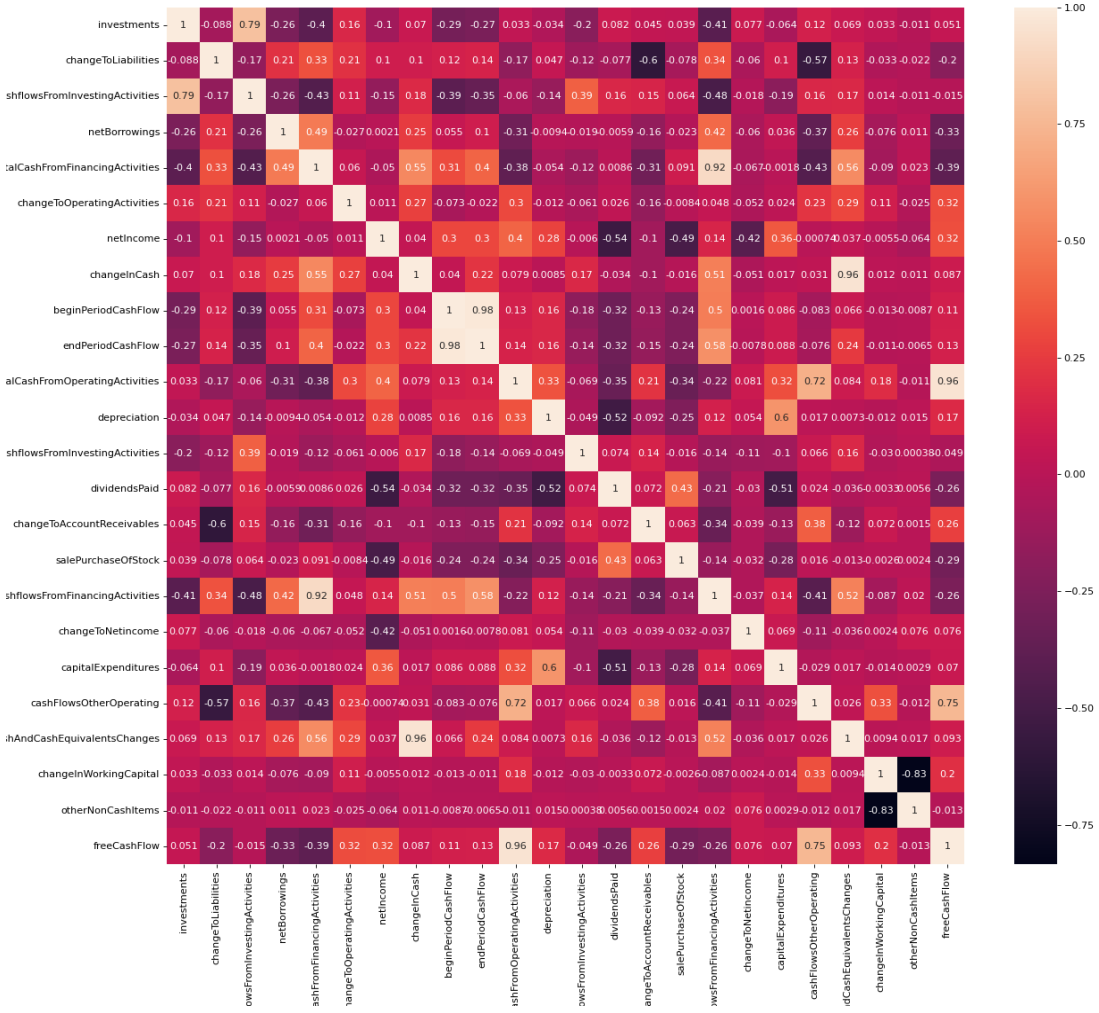
We acquired income statement, cash flow, balance sheet and stock price for index S&P500 and Nasdaq of the last 5 years with API from eodhistoricaldata.com and stored them as csv files with yearly and quarterly statistics. Currently we are mainly using quaterly data and we may consider using yearly data after more is collected.

3.2 Feature Extraction

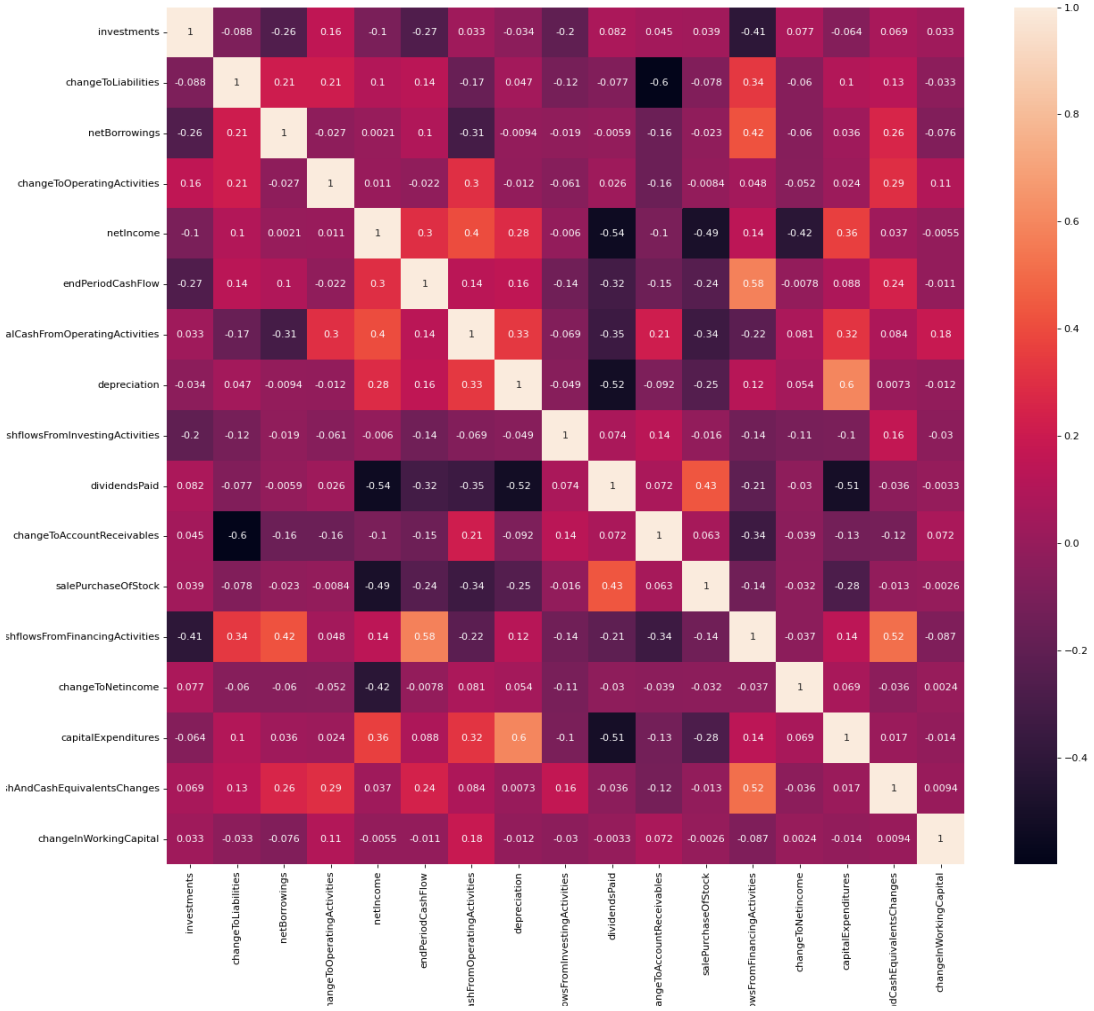
We want to build prediction models on features that can represent the financial condition of the company and we do feature selection in two phases
1, automatically remove features with too many missing columns and highly correlated features.
2, manually pick features from the rest that we think are crucial.
Take cash flow statement as an example



We remove features with more than 20% missing



Correlation pattern



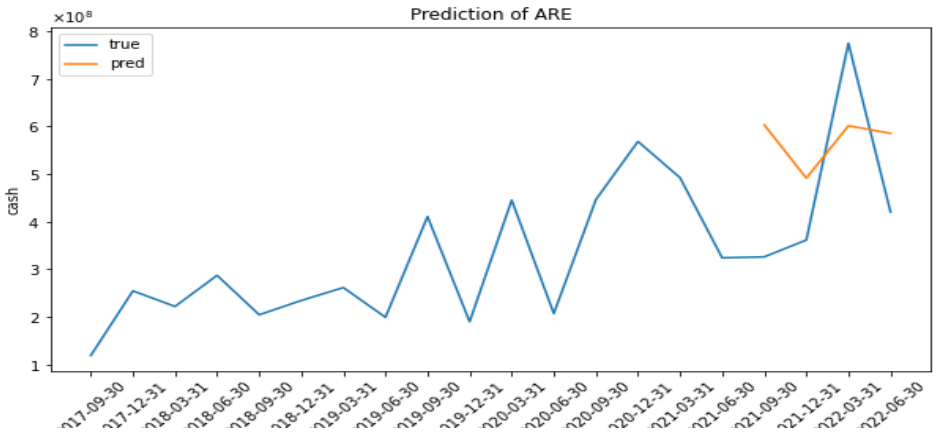
Correlation pattern after setting threshold of removal to 0.7

Finally we filtered manually from these features and picked 'investments', 'changeToLiabilities', 'netBorrowings', 'changeToOperatingActivities', 'endPeriodCashFlow', 'depreciation', 'changeToNetincome', 'cashAndCashEquivalentsChanges', 'capitalExpenditures', 'changeInWorkingCapital'.
Finally, We joined the 3 statements and kept the features we are interested in. An extra correlation analysis was done but it seems high correlation doesn't exist for features from different statements, and our

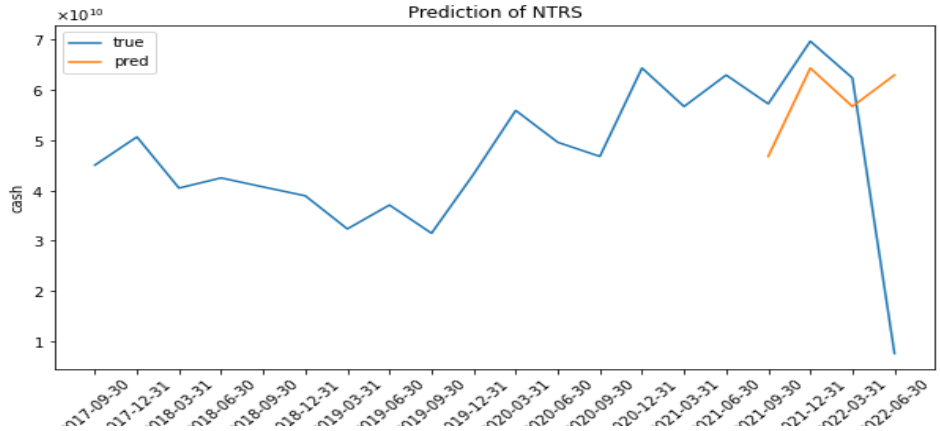
final selected features are: 'commonStock', 'retainedEarnings', 'goodWill', 'cash', 'shortTermDebt', 'propertyPlantEquipment', 'netReceivables', 'accountsPayable', 'investments', 'changeToLiabilities', 'netBorrowings', 'changeToOperatingActivities', 'endPeriodCashFlow', 'depreciation', 'changeToNetincome', 'cashAndCashEquivalentsChanges', 'capitalExpenditures', 'changeInWorkingCapital', 'incomeBeforeTax', 'sellingGeneralAdministrative', 'grossProfit', 'operatingIncome', 'otherOperatingExpenses', 'interestExpense', 'incomeTaxExpense', 'totalOperatingExpenses', 'totalOtherIncomeExpenseNet'.

3.3 Uni-variate Modeling

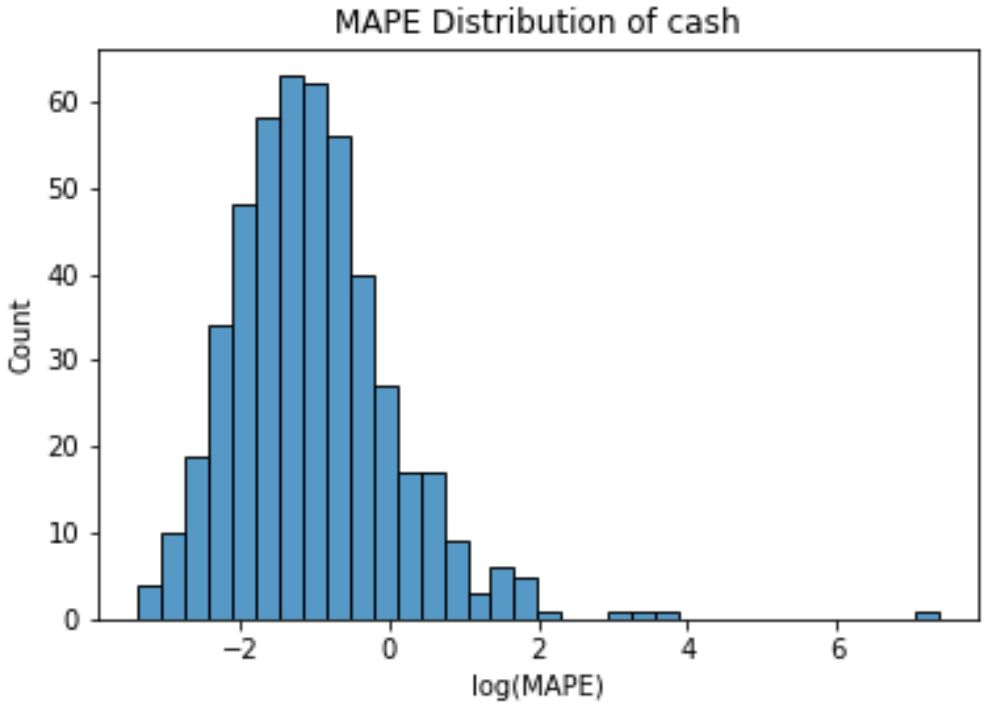
We used the auto_arima model in python to build our baseline models. We filtered the dataset and kept only those companies with full(5 years 4 quarters so 20 records). For each feature, we find companies with no missing values in that feature and build arima models for every such company. Since we haven't come up with a good metric that combines all the features, we measure the mean absolute percentage error(MAPE) for each feature and use it as our guidline. Here are some prediction results.



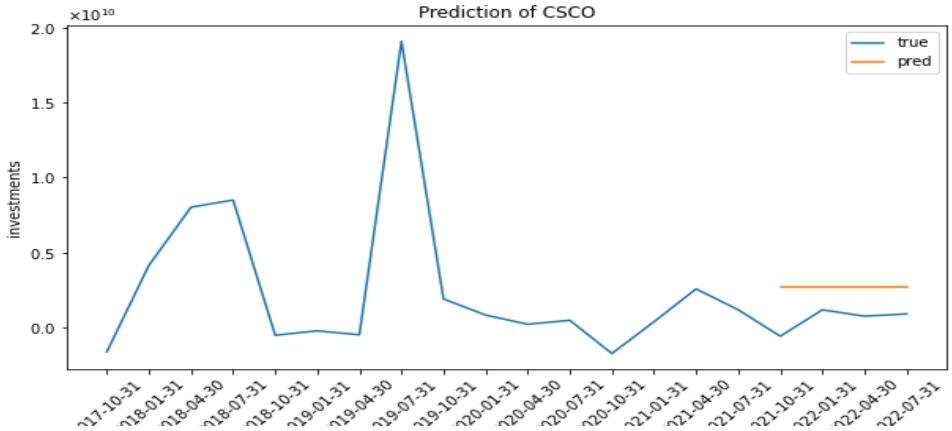
We can see that the predicted cash has similar trend with the real cash but different in scale.



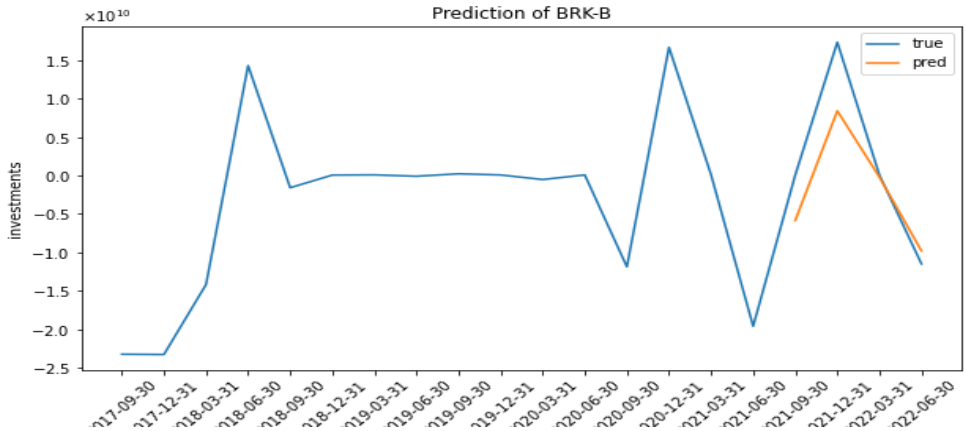
The model is doing a descent job except failing to predict the final sudden drop.



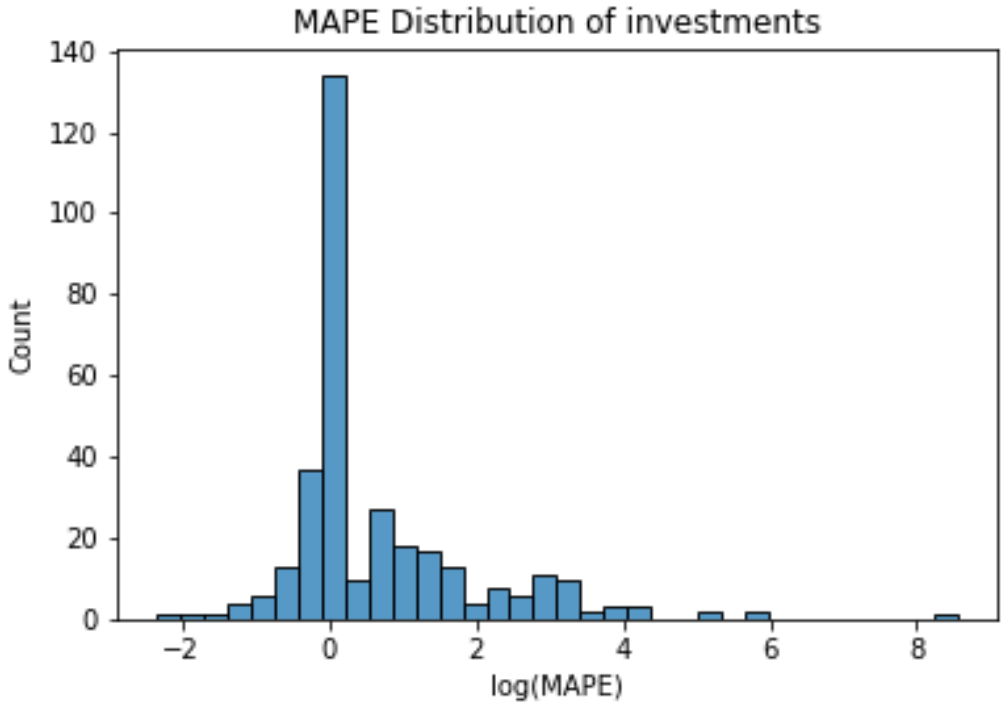
The distribution of log MAPE is close to normal but slightly right-skewed. On log scale we would expect $ln(0.1) \approx -2.3$ to be a threshold for a good prediction and we can see not many models have achieved that.



The model just gives a constant.



The model predicts well.



We can see the vast majority of models have log(MAE) close to zero meaning the predicted value can differ as much as 2 times from the actual value.

From those examples we can see that the baseline models may be able to make good predictions to some series, the general performance is not as good as what we would want to achieve.

4 Future Works

Our next step would be trying to build our first multivariate models with tree based methods. Besides, we want to define a more advanced metric other than MAPE which can hopefully combine the prediction outcome of all features and reveal whether the model is making a good prediction to the trending of series. Our mentor suggests that we could take the investor's estimate into consideration and we will try to come up with something related. We would also want to look closer into the features with high missing ratio and see if there exists a default imputation so that we can include them. For example if something is missing because the company does not have it, we may consider to impute with 0.

We would also like to impute for the ones with low missing ratio. Although in the univariate case missing values won't severely affect the model quality since we are making prediction on single columns and the largest missing ratio is 10%, about a half rows have at least one value missing and this will be problematic in the multivariate case.