Motivation and Background

According to the Association of Certified Fraud Examiners, fraudulent financial statements account for 10% of white collar crimes, which pose serious threats to shareholders and investors. Typically, financial statements are hundreds of pages long with up to one thousand numeric entries, making it a daunting challenge for auditors to flag out misstated ones. Conventional methods for detecting misstated statements are horizontal and vertical comparisons (comparing among similar companies, and among different years of the same company), which require considerable domain knowledges in accounting and the businesses. Recently, several statistical tests and machine learning algorithms have been applied. For example, Dechow et al. 2011 [1] used several F-tests to find entries that are most likely associated with frauds. Our aim for this project is to incorporate state-of-the-art neural network models and interactive visualization methods to automate the process of pre-screening potentially misstated financial statements.

Problem Statement

In this project we want to answer two questions: (1) compared to other traditional classification methods, can neural network models effectively detect misstated statements; and (2) can we visualize the results of our models interactively, such that domain experts and auditors can understand the result of our model and visually explore features of interests?

Prior to this project, we had no knowledge in financial statements, therefore considerable efforts were spent on choosing and constructing relevant features. Using neural networks for anomaly detection is a rather new topic, with no common protocols or libraries. After extensive research, we explored with two types of neural network models (autoencoder and LSTM) and achieved satisfying results. Another great challenge we had was to find a plotting library to effectively and interactively convey our findings. We experimented with many tools, including Superset, Tableau, and eventually decided to use Plotly Dash for our UI product.

Data Science Pipeline

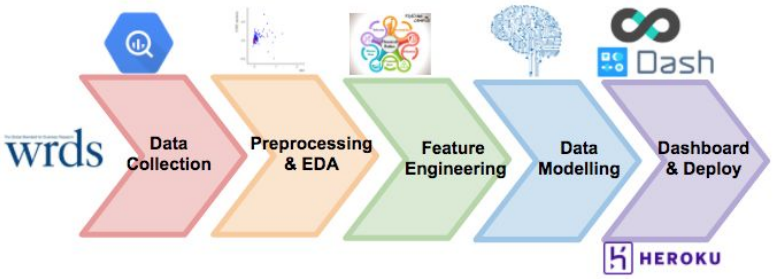

Figure 1 Our data science pipeline adapted from OSEMN

As shown in Fig. 1, we followed the OSEMN [2] process for working through a data problem. OSEMN is an acronym that rhymes with “possum” or “awesome” and stands for Obtain (data collection), Scrub (data preprocessing), Explore (EDA, feature engineering), Model, and iNterpret.

Data Collection

Our data source is COMPUSTAT, a database of financial, statistical and market information on active and inactive global companies throughout the world, which is readily accessible as SFU faculties and graduate students through Wharton Research Data Service (WRDS), a award-winning research platform and business intelligence tool.

The raw data we queried ranges from Jan 1980 to Mar 2018, with altogether 435497 observations and 981 features, which turned out to be over 1GB and took some time to read into memory using Python Pandas. To solve scalability issues in case we want to process more data in future, we managed to store the data on the Google BigQuery, which is a data warehouse enabling super-fast SQL queries using the processing power of Google's infrastructure, where new data can be appended to existing tables. It integrates perfectly with Pandas and provides convenient access to query portions of the processed data, minimizing memory usage on local computer.

Data Preprocessing and Exploratory Data Analysis

Rather than data modelling, data processing and DEA account for about 80% of the work for data scientists, according to a survey [3] published in Forbes. They require substantial data proficiency and domain knowledge.

Our dataset, COMPUSTST, has been in use for decades so the data turned out to be fairly clean. We've encountered mainly three problems during data preprocessing:

- Deal with missing values. Several reasons are stated to possibly cause missing values in this dataset: (1) various Accounting Standards interpretation of accounting rules; (2) the way Compustat databases evolve/change over time, with some legacy variables; (3) the type of company - Financial or Industrial, with Financial sector type companies usually active in the insurance sector, banking sector, etc. Thus, it makes sense for us to just fill the missing values with zeros. Although special cases may still be present, in order to streamline the preprocessing and make it part of the pipeline, those were not currently dealt with.
- Identify misstatements. Two solutions have been proposed and: (1) if variable retained earnings were non-zero, it indicates a restatement event, thereafter we exclude restatements that are caused by merger and acquisition/change in accounting rules. This approach identified 2.7% of statements to be misstated. (2) we made use of external datasets - AAERs, which is published by SEC and identifies firms for review through anonymous tips and news reports. This approach identified 0.55% of statements to be misstated. We ended up with the second approach according to expert opinion of our advisor from Business School.
- Tell whether values have been amended after being identified as misstated. Normally COMPUSTAT will backfill misstated numbers when a company files an amended 10-K. However, one literature found out that only one of the nine firms' financial data on COMPUSTAT has been backfilled with restated numbers. So we stayed with the pre-amended dataset in COMPUSTAT.

During EDA, as there are controlling variables such as industry group (sic), countries and years that have significant effects on the result, we started the data exploration focusing on one specific group by plotting their misstating frequency (Fig. 2). As it turned out, industry group with sic 73 - Business Services, has the most misstating frequency.

We started with some of the most important variables in financial statement identified by our advisor from Business School, such as those related to assets, liabilities and equities. We found out that most of such variables are strongly correlated, presenting opportunities for further processing with dimensionality reduction (Fig. 3(a)), which will be discussed in more detail in Methodology section.

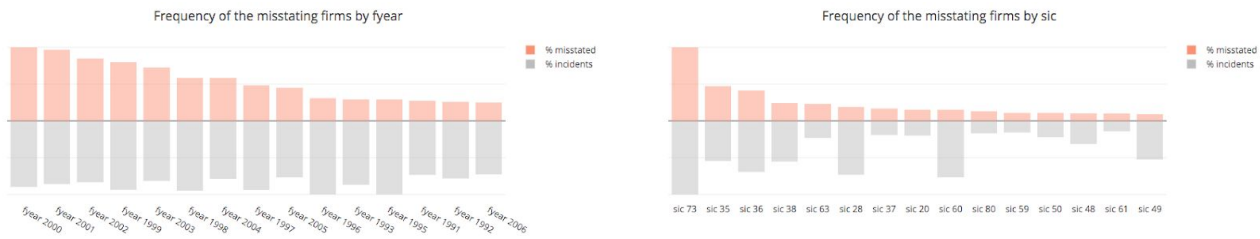

Figure 2 Frequency of missing firms by financial year (left) and industry group (right)

Feature Engineering

The raw dataset contains 981 features, among which are possible noise that will affect modelling performances and visualization clarity, thus we need to perform more extensive feature engineering. Other than extracting those important features given by the professor, we resort to Ratio Analysis that is adopted and most commonly used in the industry. Ratio analysis involves evaluating the performance and financial health of a company by using data from the current and historical financial statements. We calculated 24 such ratios covering all six main groups categorized as Liquidity Ratios, Solvency Ratios, Profitability Ratios, Valuation Ratios, Activity ratios and Coverage Ratios.

Also, in the paper Dechow published [1], they engineered a set of ratios, including accrual quality such as % change in soft assets, nonfinancial measures such abnormal reductions in the number of employees, and off-balance-sheet activities. These ratios were found to indicate that managers have been hiding diminishing performance during misstatement years, or raising more financing. According to that, we calculated 13 such variables to be included in our feature set. Altogether, we ended up with 250 raw features and 37 calculated ratio variables.

Data modelling will be discussed extensively in the Methodology and Evaluation sections and interactive visualization with dashboard deployment will be discussed in the Data Product section.

Methodology

Principal Component Analysis (PCA)

Using PCA (Fig. 3(b)), we extracted two components that respectively explained 63.6%, 10.6% of the total variance. By eyeballing the distribution, we can see most of the misstatements (~90%) are hiding among the cluster, while a few of them can be identified as outliers, among other outliers that can potentially be misstatements. After some initial digging, this project, as a matter of fact, turns out to be quite challenging.

Here we want to emphasize that if only using modelling to detect misstatement, one unavoidable issue is that the revelation of a misstatement by the SEC is a rare event, and there are likely many cases where a misstatement goes undetected or is at least not subject to an SEC enforcement action, which leads to analysis that can identify only misstatements actually identified by the SEC. Therefore, we decided to employ two approaches in solving this problem, one is through data modelling, the other is to design a dashboard for domain experts to detect misstatements through interactive visualization, both of which will be discussed in more detail in the following sections.

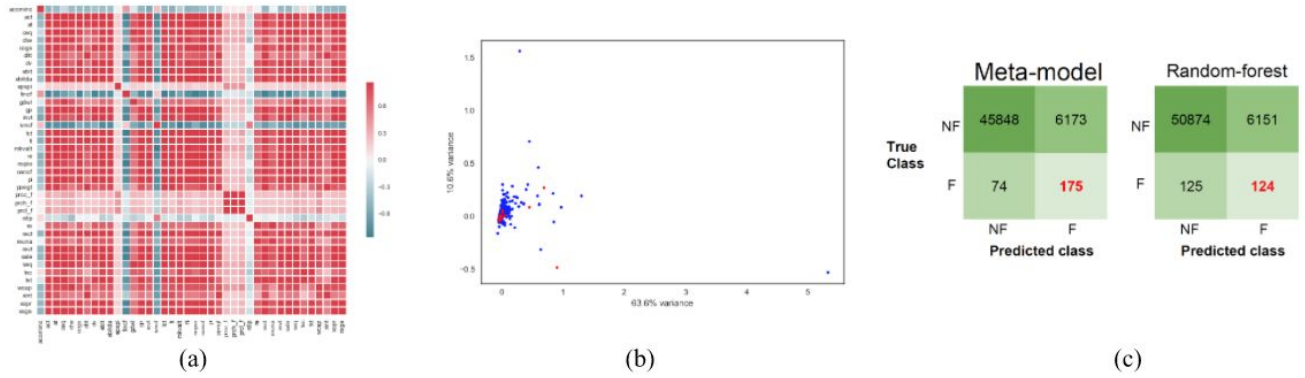

Figure 3 (a) Correlation plot among variables with level one importance (b) Scatter plot of top two principal components (c) Confusion matrices of our meta-model and a simple random-forest classifier. NF means non-fraudulent (correct) statements; F means fraudulent (misstated) statements.

Neural Network

We used two types of neural network models (autoencoder and LSTM) to detect anomalies in financial statements with Keras library in Python. 80% of data was used during training, and 20% was set aside as test data. As a benchmark for model performance, we also trained and evaluated a random forest classifier on the same dataset.

An autoencoder is an artificial neural network used for unsupervised learning which aims to learn a representation (encoding) for a set of a data. It consists of two parts, the encoder and the decoder, where the encoder transforms data to a small hidden layer (also called bottleneck) and decoder transforms the compressed data back to original data. During training, the encoders and decoders are chosen to minimize reconstruction errors. Autoencoders aim at reducing feature

space (i.e. dimensionality reduction) in order to distill the essential aspects of the data to capture non-linearities and subtle interactions within the data. Recently, autoencoder has been used in various cases of anomaly detection, including credit card transaction fraud detection [4].

We decided to use autoencoder in our project, because we believed there to be underlying structures in entries of financial statements representing accounting rules and/or common economic patterns of companies. To train the model, we only used all correct statements in the training data to extract the underlying structure of correct statements. Then we ran the model on all testing data to calculate and visualize the reconstruction error (root mean squared error, RMSE). Lastly, a threshold was chosen, above which an entry would be labeled as "fraud". The idea is that the further a statement deviates from the underlying structure of correct statements, the more likely that statement is misstated.

Long short-term memory (LSTM) is a type of recurrent neural network (RNN), which is designed to recognize patterns in sequences of data therefore suitable for learning trends in time. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate, the combination of which can "remember" the cell content along a sequence. LSTM model can be used to predict data at time T based on data at time T-1. Similarly to above, during training, we only used all correct statements to extract the underlying time-series trend of correct statements. The model was then used on all testing data and RMSE was calculated for each entry. A threshold was chosen, above which an entry would be labeled as "fraud".

Because we had two sets of data (the raw financial statement and the ratios) and two types of models, we ran a total of 4 models: autoencoder with raw data, autoencoder with ratios, LSTM with raw data, LSTM with ratios. The autoencoder with raw data model had five hidden layers (3 encoders of 32, 16, 8 neurons each, and 2 decoders of 16, 32 neurons each), whereas the autoencoder with ratio model contained three hidden layers (2 encoders of 8, 4 neurons each and 1 decoder of 8 neurons). The LSTM model for raw data had one hidden LSTM layer with 50 neurons, while the model for ratio data had one hidden layer of 8 neurons. For autoencoders, we trained the model for 100 epochs, whereas LSTM models were trained for 50 epochs. For all models we used the ADAM optimizer.

To ensemble the four models together, we took reconstruction error from each model as features and the label as target in a random forest classifier. The parameters of the ensembled models were selected via cross-validation on the training dataset. We then compared the performance our meta-model with a benchmark random forest classifier on raw data. From cross-validation, the random forest classifier had 60 trees of maximum depth 6.

Evaluation

Because the data is highly unbalanced (only 0.5% were misstated), we could not use overall accuracy to evaluate our model efficacy. Instead, we focused on precision and recall of the misstated class, which measure our ability to select potential problematic statements. More specifically, precision is the fraction of true positives among all cases that were labeled positive, while recall is the fraction of true positives that have been retrieved over the total amount of all positives.

Here we showed the confusion matrix of our meta-model and the benchmark random forest classifier. As can be seen from Fig. 3(c), compared to the random forest classifier, our meta-model had about the same precision; however, our recall (0.7) is more than 40% higher than the that of random forest classifier (0.5), suggesting our model has significantly increased our ability to detect potential misstated statements.

We believed by capturing the underlying structures and time-series trends from two angles (raw features and ratios), our meta-model had better grasped the intricacies of the dependencies and non-linearities among the variables, therefore is more supreme than simple classification methods.

Data Product

Our Dashboard, https://financial-dashboard-app.herokuapp.com/, is built with Plotly Dash, is designed for domain experts such as auditors to interactively visualize, explore and possibly spot suspicious financial statements by identifying outliers through plotting data points in two dimensions and visualizing historical trend of specific variables for each data point. This dashboard makes it possible to (1) incorporate data modelling results using size of the data point; (2) target a specific industry group in a specific year to identify outliers; (3) use different set of features either form Dechow or Ratio Analysis; (4) present time series for any interesting companies users discovered.

Note that consistent with what we found in the modelling result, financial statements that are misstated are usually well hidden and rarely identified among the outliers. That explains why in AAERs the misstatements were usually recognized by anonymous tips. Please see Fig. 4 and our video for more details on how to use the Dashboard.

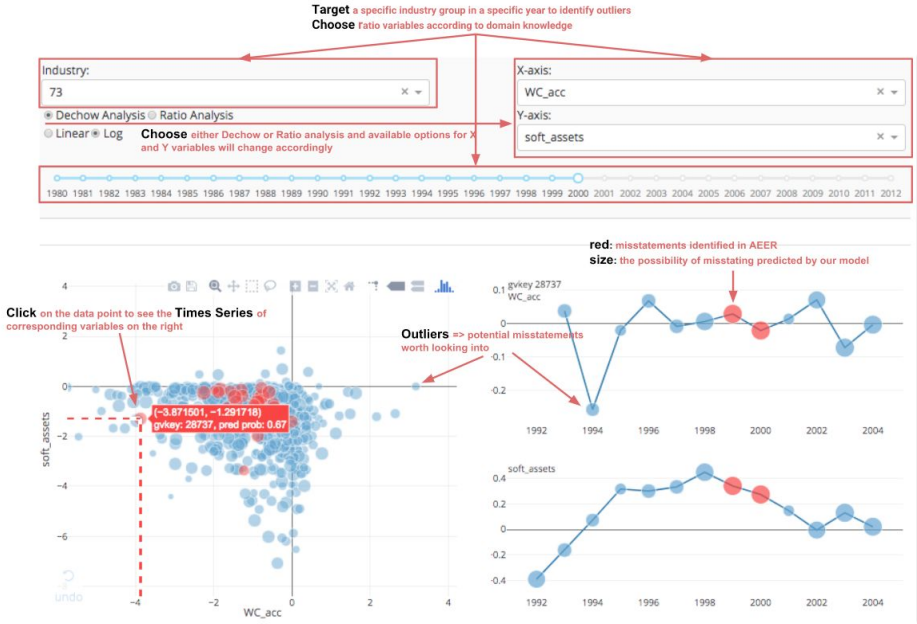

Figure 4 Instructions on how to use the Dashboard

Lessons Learnt

Our most important lesson is that misstated financial statements are difficult to identify because large amount of human resources are usually devoted if the company intentionally plan to forge a seemingly reasonable financial statement. Though our meta-model outperformed random forest, it also generated a high frequency of false positives, due to the fact that most misstatements likely went undirected by SEC, resulting in a highly unbalanced dataset. However, we managed to come up with the Dashboard, therefore instead of buried in and crunching tremendous amounts of numbers, domain experts can explore the financial statement in a more intuitive way through visualization. The end product is far from complete and we plan to continue to incorporate new features, for instance, linking the data point to the actual financial statement.

Regarding the data science pipeline, we learned that uncertainties abound in any dataset, and a lot of decisions need to be made, sometimes without even knowing if it is good decision or a bad one. We only have to keep trying, realizing it is an iterative process - there is no best, only better.

Summary

In this project, our goal is to automate the process of pre-screening potential misstated financial statements. We constructed a complete data pipeline to process and clean financial statement data, engineered relevant features for two neural network models (autoencoder and LSTM), and visualized model output interactively. Based on half a million financial statements from 1980-2018, our model was able to reach a recall score of 0.7 for misstated statements, a 40% of improvement compared to a random forest classifier while retaining the same precision score. Written in Plotly Dash, our final product is a web UI of a interactive dashboard incorporating yearly trends of each accounting term and the model output, which is designed for domain experts to understand the results of our model and visually explore potential features of interests.

References

(1) Dechow, Patricia M., et al. "Predicting material accounting misstatements." *Contemporary accounting research* 28.1 (2011): 17-82.

(2) A taxonomy of Data Science. http://www.dataists.com/tag/osemn/

(3) Cleaning Big Data. https://www.forbes.com/sites/gilpress/2016/03/23//#4fdc77246f63

(4) Aytekin, Caglar, et al. "Clustering and Unsupervised Anomaly Detection with L2 Normalized Deep Auto-Encoder Representations." *arXiv preprint arXiv:1802.00187* (2018).