AIR-mazing Predictions: Text-Based Stock Market Prediction for Business Ideas

Kai Kainbacher, Maximilian Legat, Jonathan Maier, Michael Sickl
Group 09

November 2024

Abstract

In this document we present a neural network model that aims to predict the stock performance of a company based on to textual description of its business idea. The prediction is relying on how companies with similar business ideas performed on the stock market in the past. In order to achieve that, our model is being trained with business ideas and the corresponding stock performance enriched with some relevant features such as market size, investment, or team strength, This improves comparability and provides deeper insights into the factors contributing to success. Overall, by combining text-based insights and static attributes, our model aims to provide a realistic prediction of potential market success for new business ventures.

1 Introduction

A very important aspect of starting a business, if not the most important one, is the business idea. It contains the purpose or the goal of the company. And although it is not the only relevant aspect of success, it provides the foundation for all other business procedures.

Because of that it is crucial for an upcoming entrepreneur to know, whether the business idea is likely to be successful. Furthermore, if the company is present on the stock market, potential investors also do have an interest in acquiring knowledge about the stock performance of a company since a more promising idea will most likely end in a bigger return on their investment.

Therefore, our team has decided to construct a neural network model to evaluate a business idea. Based on how existing companies, with similar ideas have performed on the stock market, our model predicts how the new idea will perform.

To accomplish this, we first need to gain insight on how current companies perform and what their business ideas are. Once we have that information it is being enriched with further static parameters such as market size, investments and the size of the team to make it easier for the model to put their success into perspective. That allows the model to give a prediction of the stock performance based on how similar businesses performed in the past. For a more accurate prediction it is possible to also provide those static parameters. This document outlines how the dataset and model are constructed in order to suggest an accurate outlook.

2 Division of Work

The work will be split up as follows:

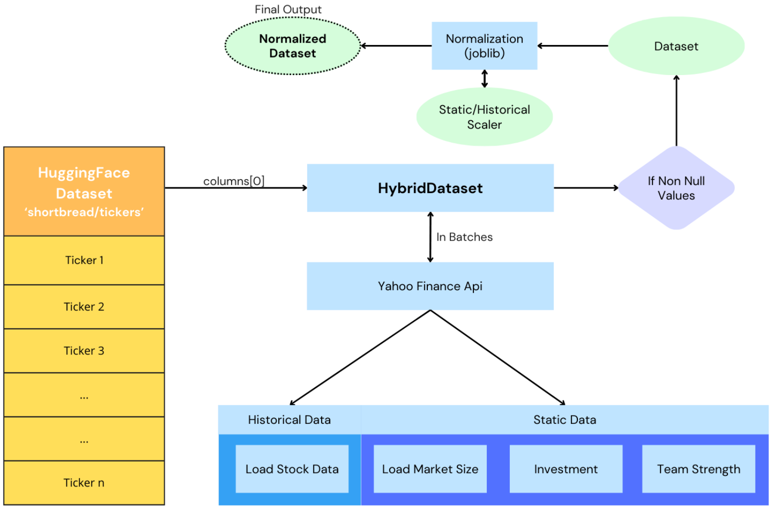
People	Work
Kai Kainbacher	Training Model
Maximilian Legat	Evaluation and Illustration
Jonathan Maier	Creating Dataset and Model Architecture
Michael Sickl	Evaluation and Document

3 Dataset

The dataset used for this project was created by combining a dataset from Hugging Face with financial information retrieved from the Yahoo Finance API. This combination results in a hybrid dataset that integrates both static and dynamic components and therefore can be effectively utilized to train our model.

3.1 Dataset Creation

The dynamic data in our dataset is the historical stock performance in monthly intervals. Here, we are looking to get the closing prices for the most recent months for each ticker, which can be dynamically adapted but is set to 24 months for now. The static attributes of the dataset involve the **market size, investment level** and **team strength**. However, additional attributes, such as the date the company entered the market and other relevant factors, are likely to be included in the future. The **market size** refers to the total revenue of the company, showing the overall scale of the company. The **investment level** represents the total assets of the company and is used as an indication of the financial capacity of the company. The **team strength** simply represents the number of employees. If a company is missing any of these attributes, then they are excluded from the dataset to ensure data quality and no distortion. This step makes sure that the final dataset remains correct and does not include incomplete data. In addition to the static and historical data, there is the **business description**, which is the most important part of our dataset and that should contain the core idea and area of activity for each company. The Hugging Face dataset already provides this for us but we can replace or enrich this description with the yahoo finance API, if needed.



3.2 Data Processing

The data processing was carefully designed and further updated to handle large volumes of data efficiently. It consists of many stages to ensure correctness and prevent issues between the dataset and the model.

3.2.1 Batch Processing and Data Collection Stage

To handle large datasets, we use batch processing. We start with around 7,000 stock tickers from the Hugging Face dataset and split them into smaller groups of 50 tickers each. This method helps us process the data efficiently while staying within API limits.

3.2.2 Validation and filtering

The validation and filtering is performed during data collecting. This ensures that stocks with insufficient data or missing attributes will be excluded. Missing values in the stock history are padded with zero, to keep the same data structure.

3.2.3 Normalization

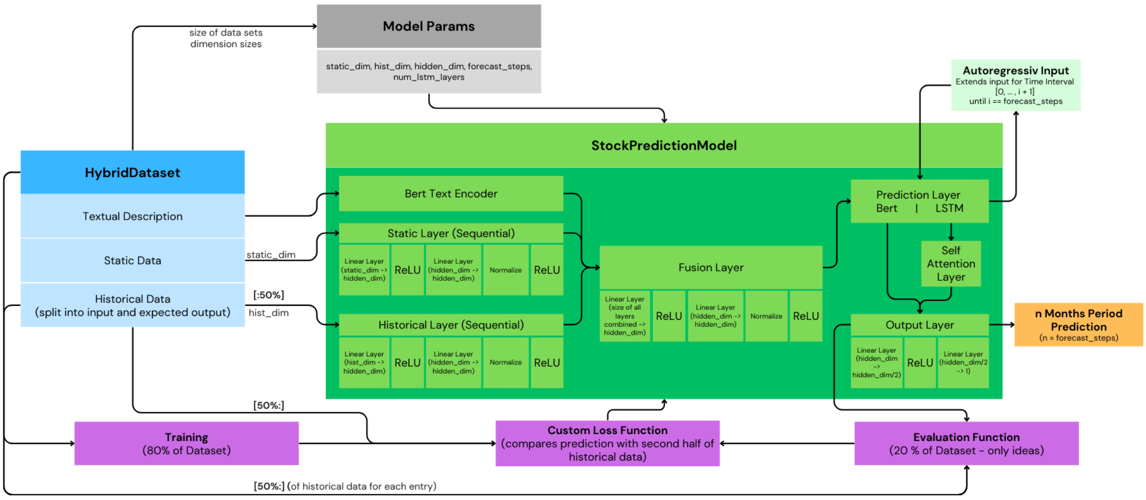
After identifying the static and historical data, it has to be normalized in order to allow for a smooth training process. Since our data is of very different nature, we apply different normalization methods. For the static features, we use the **StandardScaler**, which standardizes the variables by removing the mean and scaling them to unit variance. For the dynamic variables, we use the **MinMaxScaler**, which rescales the data to a fixed range (0,1) so that the variance is not too large.

$$\begin{array}{c|c} \text{For static features:} & \text{Normalization for dynamic features:} \\ z = \frac{x-\mu}{\sigma} & z = \frac{x-\min(x)}{\max(x)-\min(x)} \end{array}$$

After the dataset has been normalized, we save both the processed data and the scaler objects for static and dynamic features. By storing the scaler objects, we can ensure consistent scaling across future dataset extensions and also enable denormalization for our predictions.

4 Base Architecture

The Stock Performance Prediction Model includes several layers that work together to handle our task of stock predictions effectively.


Model Params
static_dim, hist_dim, hidden_dim, forecast_steps, num_lstm_layers
size of data sets
dimension sizes
HybridDataset
Textual Description
Static Data
Historical Data
(split into input and expected output)
static_dim
[:50%]
hist_dim
[50%:]
StockPredictionModel
Bert Text Encoder
Static Layer (Sequential)
Linear Layer (static_dim -> hidden_dim)
ReLU
Linear Layer (hidden_dim -> hidden_dim)
Normalize
ReLU
Historical Layer (Sequential)
Linear Layer (hist_dim -> hidden_dim)
ReLU
Linear Layer (hidden_dim -> hidden_dim)
Normalize
ReLU
Fusion Layer
Linear Layer (size of all layers combined -> hidden_dim)
ReLU
Linear Layer (hidden_dim -> hidden_dim)
Normalize
ReLU
Prediction Layer
Bert | LSTM
Self Attention Layer
Output Layer
Linear Layer (hidden_dim -> hidden_dim/2)
ReLU
Linear Layer (hidden_dim/2 -> 1)
Autoregressiv Input
Extends input for Time Interval [0, ... , i + 1]
until i == forecast_steps
n Months Period Prediction
(n = forecast_steps)
Training
(80% of Dataset)
Custom Loss Function
(compares prediction with second half of historical data)
Evaluation Function
(20 % of Dataset - only ideas)
[50%:] (of historical data for each entry)

In the image above an overview of the overall structure is given. In the following we will have a closer look at the layers individually.

4.1 Layers

The **Text Representation Layer** is an important part of the model that uses the Sentence-BERT model (”all-MiniLM-L6-v2”) to turn textual business idea descriptions into vector embeddings. This layer effectively understands the semantic meaning and works with fixed settings, which keeps the strength of the pre-trained model without needing extra processing power.

The model also possesses a **Static Feature Layer** that handles our static data as described above. Using a sequential layer, these features were transformed into a compact vector form. This process enriches the model’s understanding by providing the context of the business environment.

The next layer is the **Historical Feature Layer** to handle data about how stocks did in the past. This layer is similar to the static feature layer, applying deep transformations to historical inputs. It helps the model to get a better understanding of our sequential data.

The **Fusion Layer** combines the information from the text, static, and historical layers. This ensures that the model connects our inputs properly and can effectively continue with the next operations. The fusion design combines the text data (which has a fixed size of 384) with information from static and historical layers, scales it down to a specified number of dimensions, and then applies deep transformations to this combined data.

The **Prediction Layer** predicts a sequence of expected stock performance for a defined number of months. For this task LSTM and BERT were utilized and compared. The prediction process is an iterative process, where previous predictions are used as input for the next one. This enables the model to make coherent predictions on the stock performance. To extend the capabilities of the LSTM an additional layer was introduced, the **Self Attention Layer**. This layer helps with identifying crucial data to make better predictions.

At last, the **Output Layer** is our final sequential Layer, that takes the processed features from the attention mechanism or the prediction layer and reduces their dimensionality to create a single forecasted value.

4.2 Future Extension and Improvements

As we continuously identify new problems and potential improvements our model is constantly adapting and evolving. Therefore, its very likely that the above described architecture will change and parameters such as the number of dimensions for different layers will be adapted.

5 Methods

In order to make our stock performance predictions as good as possible, our project consists not only of our model, but also additional logic that ensures an effective training. We integrated a custom **training loop**, which processes batches of the dataset for a specified number of epochs. The training data comprises 80% of the dataset, with 12 out of 24 months designated as the target. The predictions are based on the remaining 12 months. This target data is used to evaluate the model's prediction accuracy.

5.1 Loss and Evaluation Function

We employ a **CustomLoss** class to calculate the loss between the prediction and the target data. Our CustomLoss is based on the TemporalLoss function, which computes temporal differences in the data and applies them as a penalty. Additionally, we incorporate the mean squared error, a penalty for negative predictions, and a penalty for insufficient diversity.

After each epoch, the model undergoes evaluation. The **evaluation function** takes a batch of ideas as input, which comprises the remaining 20% of our dataset. During this phase, the model is not trained but rather assessed by comparing its predictions with the target data using our custom loss function. To provide **visual feedback**, we plot the predictions alongside the target data. This graphical representation helps us better understand the model's performance and identify areas for improvement.

Overall, this approach allows us to train, evaluate, and visualize our model's performance effectively, ensuring we can refine our predictions and optimize our results.

6 Evaluation

As mentioned earlier, the training is performed using 80% of the dataset, with the remaining 20% held back for evaluation purposes. During the evaluation phase, the model is provided with a business idea and tasked with predicting the company's performance over the next 12 months. The predictions are then assessed using the CustomLoss function and visualized through a plot that compares the actual data with the model's predictions. In addition, we will do an intensive analysis on the model by selecting a few new companies from the stock market, defining their core business ideas, and conducting a detailed comparison between our predictions and their actual stock performance. Of course, huge economic crises such as Covid-19 that can largely impact companies assets are not predictable, therefore the predictions should be used with caution and not be taken as granted.