Predicting Funding in Indian Startups

AUTHORS

Efe Alpay, Christopher Williams, Raymond Kao

01. Introduction

The tech startup scene is fast-paced and exciting, with huge sums of money constantly being invested into businesses around the world. As students with an interest in business, we wanted a deep learning model that can predict how much funding a startup will receive, so that's what we built. While we initially wanted to use a global dataset to create a more general model for predicting startup success, we ended up limiting ourselves to an Indian startup dataset as it had more robust data for what we wanted to achieve.

02. Objective

Our objective for this project was to predict startup funding using an MLP that is trained on startup attributes and embeddings of short descriptions of the startups.

03. Methodology

Preprocessing:
A large portion of our project was finding and preprocessing the data. The best dataset we found for our task had different CSV files for each month's data. Therefore, we had to combine all of these CSV files into one data frame and remove any noise. Another significant part of our preprocessing was normalizing the funding data. Since funding values went up to millions and were heavily skewed, we needed a way to make them more stable to use with our model. In order to accomplish this, we first applied a log transformation to the funding values, which turned the data into a normal distribution. We then used a MinMaxScaler to ensure the funding values matched the output of our model (between 0 and 1).

Embedding Generation
We used Google's Universal Sentence Encoder, a transformer-based model, to generate an embedding (a 512-dimension vector) for each short company description.

Model Building
We implemented a dual-structure model consisting of an MLP and a pre-trained Transformer. We used the Transformer model to generate embeddings from short company descriptions, which we fed into the MLP alongside the company's other attributes such as investors and startup location.

Experiments
We ran a set of ablation experiments where we removed different columns from our data and measured how the model performed without these columns. We also ran our model on a set of different hyperparameters to further optimize performance. The results of these experiments are in the following sections.

04. Findings/Experiments

Results

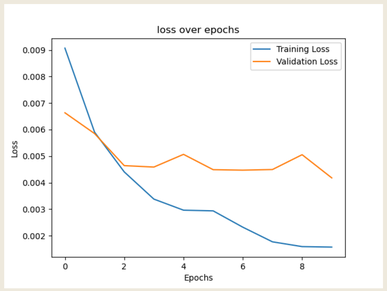

Fig. 1

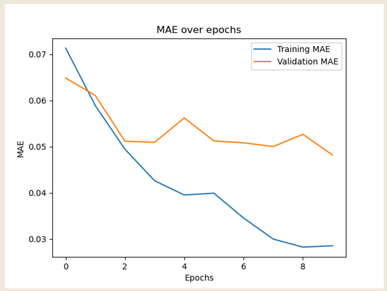

Fig. 2

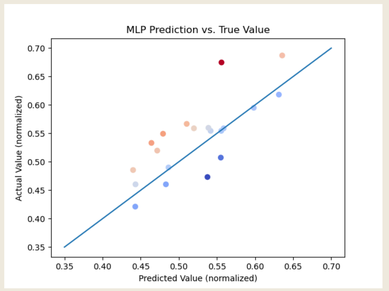

Fig. 3

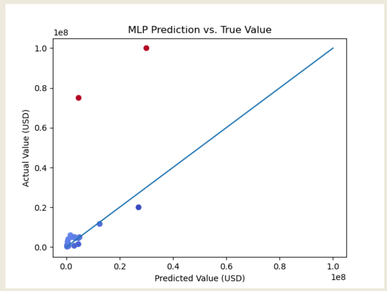

Fig. 4

Results Without Time Information

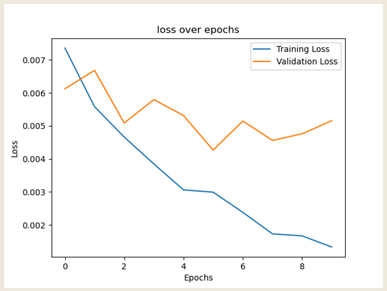

Fig. 6

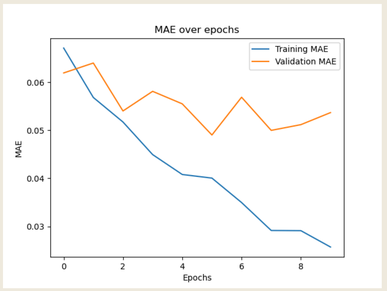

Fig. 6

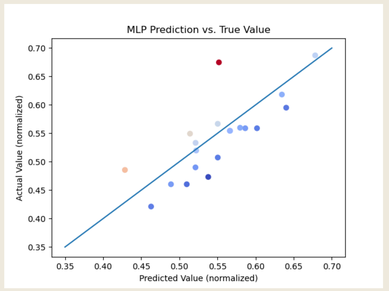

Fig. 7

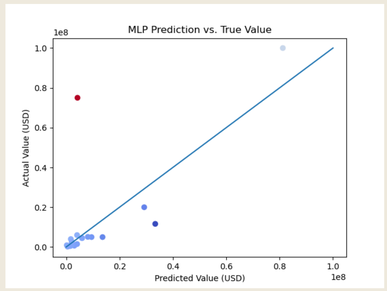

Fig. 8

05. Analysis

- Our model's training loss drops quickly within 10 epochs. As depicted in fig. 1, the rate of decrease slows down significantly as the loss approaches 0. We tried increasing the number of epochs, but found little to no increase in performance, suggesting that the model had converged to an optimal solution.
- Our model performs much worse on startups with high funding amounts – it consistently underestimates these values. This is probably because our dataset contains very few startups with these high funding amounts (> $20,000,000), so it lacks sufficient examples to learn the patterns associated with these high-value cases.
- While there appears to be an extremely small difference in validation loss and validation MAE over epochs, this is only because the metrics are calculated from normalized values. Funding values ranging from 0 to over 100 million are normalized to 0-1, which means that the "small" decrease in validation loss is actually significant.
- In our ablation experiments, we tried removing various columns from the dataset to see if and how it would affect model performance. While no single column appeared to affect the loss drastically, we found that removing the time information (month and year columns) from the inputs introduced bias (fig. 7): the model overestimated funding amounts compared to before. You can also see that the ablated model's validation loss is greater than the original model's.
- The rightmost figures (fig. 4, fig. 8) show model predictions that have been un-normalized, so the dots represent actual dollar amounts. Comparing the two figures, we can see that the ablated model's predictions stray further from the truth line.

06. Conclusion/Discussion

Our model was relatively successful in predicting the funding amount that Indian startups receive.

Real World Implications: As briefly discussed in the analysis portion, the small differences in model accuracy implies great differences in actual funding amount. Therefore, while we were able to reduce the loss of our model to a reasonable value, the model still isn't ready for real world applications. While it may give users an idea of how a startup could do, it is currently nowhere near advanced enough that people should use it to make financial decisions.

Potential Improvements: Due to the lack of data on larger funding amounts, our model performed worse with such values in testing. Given more time and resources, the model would benefit from training on a larger dataset with more metrics in order to better predict greater funding values. Additionally, while we tried some different ways of generating description embeddings, more experimentation on that front could potentially increase the embeddings' positive effect on model performance.