Predicting The Success of Crowdfunding
Yan Chen, Yiwen Guo, Chenchen Pan
Department of Statistics, ICME, MS&E, Stanford University

Abstract
Crowdfunding platform like Kickstarter, where entrepreneurs and artists seek for support from a huge number of contributors, has become prominent over the past decade. Better understanding and more accurate prediction of the success of a project can help both contributors and creators make better use of their resources. Previous works applied traditional machine learning methods, such as SVM and Random Forest, with only categorical and numerical features, such as goal and duration, or only textual features, such as project description and keywords. To our knowledge, we are the first to apply deep neural networks with all three types of features. Using a dataset of 100k+ crowdfunding projects, our model achieves 72.78% accuracy on test set, which is significantly better than the baselines and the previous works. We also show that the trained model can help us understand the success and failure of crowdfunding projects better.

Data Preprocessing
- Turn country, currency, and double_communiation column into categorical variables having values from 0 to the number of class minus one.
- Add a new column called duration, which is computed by subtracting the date when the project is launched from the deadline date.
- Split into 90% training set, 5% dev set, and 5% test set

Baseline Models
- Random forest classifier with 100 trees
- Shallow neural network
  - On-chip vector word embedding with 10k vocabularies
  - Adam optimization
  - L2 regularization with the penalizing parameter \( \lambda = 0.01 \)
  - binary cross-entropy cost

RNN Model
- Pre-trained word embedding with 300-dimensional GloVe vectors
- Loss function: \[ J = -\frac{1}{N} \sum_{i=1}^{N} \left( y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right) \]
- Adam optimization
- Regularization: early stopping + dropout
- Our best hyperparameters are as follows:
  - LSTM state size is 64.
  - Dropout rate is 0.45.
  - FC layer output size is 64.
  - Learning rate is 0.000145
  - Epoch is 50

Discussion
- About 26% of errors are due to dataset issues, including label error (6%) and incomplete information (20%), there are some missing content with quotation marks in project’s description.
- 29% of the errors are due to the fact that the model sometimes only uses partial information for predictions (‘game’, ‘documentary’ and ‘film’ for success and ‘app’, ‘food’ and ‘music’ for failure).
- The rest 45% of errors are model errors. This is a quite challenging task for human as well because a human can only get around 30% correct (no better than random).

Future Work
- Address the mistrusted data issue. Include more features of the projects (images, videos).
- Try bi-directional LSTM. Apply batch normalization. Adjust the loss function to give different weights to the losses regarding different misclassifications.

About The Dataset
- Use a dataset from Kaggle [1], which has 188,129 examples in total. 34,561 of them are successful, and the rest are failed.

RNN Model - Hyperparameter Tuning

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropout rate</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Number of LSTM layers</td>
<td>2-7</td>
</tr>
<tr>
<td># of hidden units in one LSTM layer</td>
<td>64,128,256</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.0001-0.01</td>
</tr>
</tbody>
</table>

Result
Our result is summarized in the following table. We can see that the RNN model gives the best result.

<table>
<thead>
<tr>
<th>Models</th>
<th>Dev Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest Classification</td>
<td>76.20</td>
</tr>
<tr>
<td>Shallow Neural Network</td>
<td>69.55</td>
</tr>
<tr>
<td>RNN Model</td>
<td>72.42</td>
</tr>
</tbody>
</table>

Figure 1: Shallow neural network model
Figure 2: RNN model
Figure 3: Accuracy and cost for RNN model
Figure 4: Confusion matrix

Reference
[1] https://www.kaggle.com/annaajibhagat/kickstarter/data