ReadMe:

Dear Human Resource Manager,
First of all, thank you for giving me this opportunity to present my project. This project is about a recommend system for social media application. You can read the details below to get a rough view about this project.

1 A rough view of my thinking process:
The link prediction problem is a combination of Graph Theory and Machine Learning algorithms. Firstly, I reviewed related knowledge about both of them and I found that this problem could be mapped into supervised learning problem which label "1" and "0" indicate the positive and negative links.

To start with the implementation, I do the degrees analysis which contains indegree, outdegree and the combination first. The purpose to do this is to get a rough view about the mean number, maximal number and minimal number of followers and followees for most of the users.

Then, I implement some similarity features algorithms such as Jaccard Distance, Cosine Distance, Page Rank, etc. I add them as different columns in the train and test dataset, which are very important for Machine Learning models to process and train data.

2 Models picking:
As I have mentioned above, the task could be mapped to a supervised learning problem. So I tried different supervised learning models such as Decision Trees, KNN, Naive-bayes and Randon Forests.

The first three models all did not perform well at avoiding over-fitting. The scores for test results are much lower than that for train results. Random Forest has best performance taking advantage in its special data gathering way. All in all, ensemble algorithms usually have better performance than simple models.

3 Results:
For RF model, I tried different estimators and max-depths to find the best parameters.
My result interpretation is based on estimators = 500 and max-depths in [5,7,9,11,12,13,15].

Precision on train dataset and test dataset are nearly 77% and 89%, which indicates that the model have acceptable performance for recognizing negative links.
Since Precision = TP/(TP+FP), the smaller FP is, the bigger Precision is.

Recall on train dataset and test dataset are nearly 84% and 82%, which indicates that the model performs well at recognizing positive links.
Since Recall = TP/(TP+FN) = TP/P, the bigger TP is, the bigger Recall is.

F1 scores on train dataset and test dataset are nearly 80% and 85%, which indicates that the model has a stable performance.
Since F1 score = 2TP/(2TP + FN + FP) = 2 * Precision * Recall/(Precision + Recall), the bigger F1 score is, the smaller the difference between Precision and Recall is.
So a high score for F1 score is a strong proof for a stable model.