Resume Screening System

Marios Sarrigiannis

1

Introduction

Scope and goals of the research

The most important asset of a company are the people that make up its workforce.

Therefore, recruitment plays a central role in securing a company’s future.

Scope

A common pain when recruiting is the initial screening of resumes:

- Takes a very long time to go through hundreds (or even thousands) or resumes
- People who do it do not know domain specific information (e.g. domain skills)

Goal

This undertaking's goal is to create an **automated**, **cross-domain** resume screening system in order to assist an enterprise with the first (and most cumbersome) part of recruitment

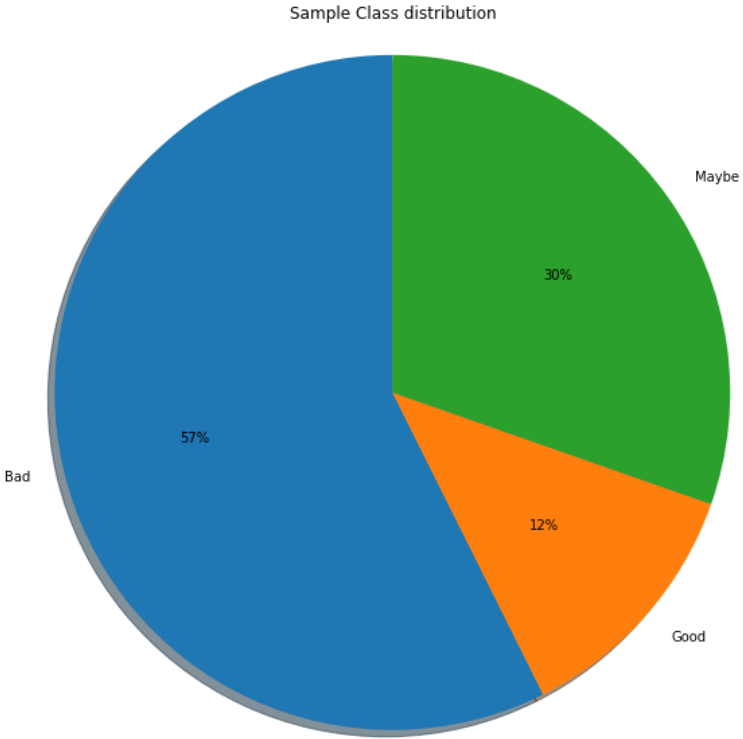
2

Methodology

Creating the system

Data

- Data was sourced from an actual recruitment process of my organization
- 327 resumes were received and broken down into 3 categories:
  - Bad
  - Maybe
  - Good



We observe a little class imbalance in the dataset

Data preparation

- Annotated resumes are placed in directories according to their labels
- A script was made to extract all text from pdf and docx files (most common file formats for CVs)
- Data was produced in a csv format (text, label)

Preprocessing

- Null records were not processed
- All text was made lowercase
- URLs were removed
- RTs and CCs were removed
- Email addresses were removed
- Most punctuation was removed
  - Notable exceptions were full stops, hashtags, the plus sign etc.

Feature Extraction

Talking to HR screening experts, they explained the main points in screening:

- Skills matching the job description
- Education level matches (or is greater than) the requirements
- Whether they have attended top universities
- Whether they have worked for top companies

Feature Extraction

In order to fulfill our requirement for the system to be cross-domain, we expect the end-user to provide the following lists:

- Required skills
- Exceptional university list
- Exceptional company list

Feature Extraction

The lists are used to produce the following:

- Skills score
- University score
- Company score

Last feature extracted is the candidate's education level, which can be: None, High School, Bachelor’s, Master’s, PhD

Feature Extraction

- Last step is to normalize the features using Min-Max Scaler
- In order to help overcome class imbalance, we tried oversampling our dataset using SMOTE, however, results were not satisfactory

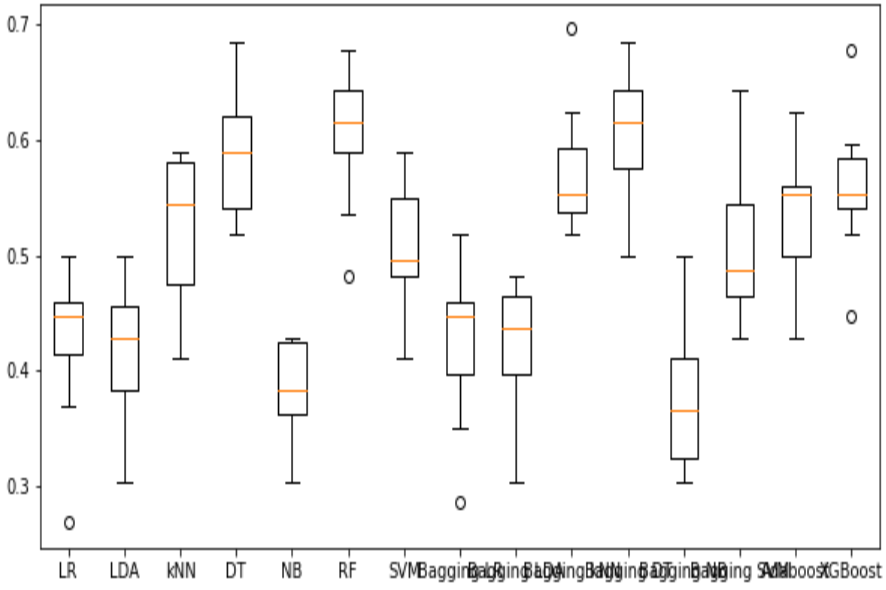
Algorithm Selection

- A plethora of machine learning algorithms were run and evaluated based on the following metrics:
  - Accuracy
  - F1-score

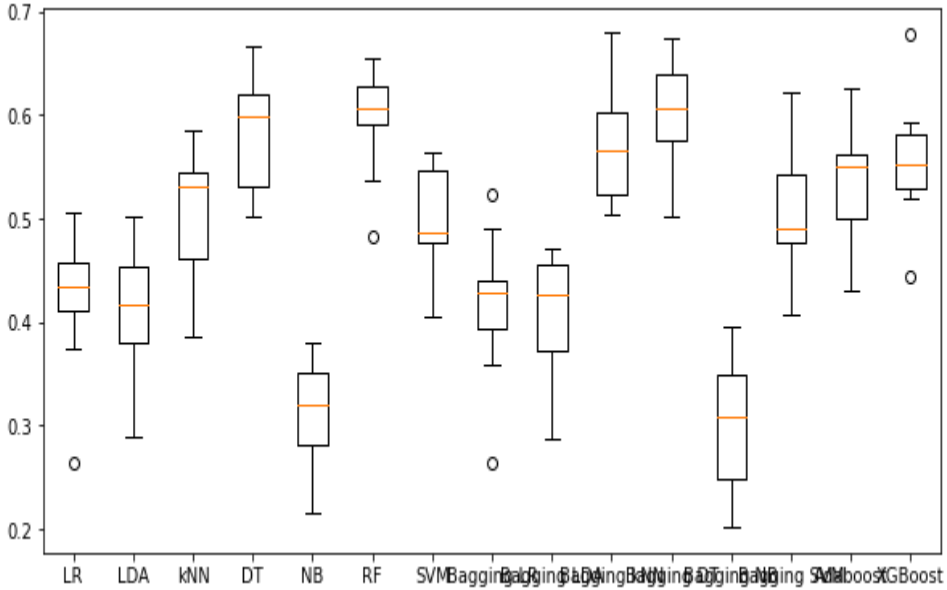
Algorithms Used

- Logistic Regression
- Linear Discriminant Analysis
- K Neighbors
- Decision Trees
- Naïve Bayes
- Random Forest
- SVM
- Bagging versions of the above
- Adaboost
- Xgboost

Accuracy



F1



Results

Algorithms Picked

- K Neighbors
- Random Forest
- SVM
- Adaboost

- All algorithm's parameters were tuned and scored against their f1 score

Algorithm	F1-score	Parameters
Knn	52%	algorithm='brute', leaf_size=25, metric='euclidean', n_neighbors=15, p=4
Random Forest	51%	max_features='log2', min_samples_leaf=4, min_samples_split=5, n_estimators=10
SVM	48%	Kernel=rbf, C=1000, gamma=0.01
ADA Boost	51%	algorithm='SAMME', learning_rate=0.97, n_estimators=8

Results

3

Conclusion & Further Work

Expanding our research

Conclusions

- This project has been an attempt to extract meaningful and useful features from text
- Performance was not great due to:
  - Small amount of data
  - Annotation mistakes
  - Personal knowledge was used to extract features

Further Work

- Investigate NLP techniques for feature extraction
- Investigate tools (such as NER) to better help with feature accuracy

THANKS FOR YOUR TIME!

Any questions?