# King Fahd University of Petroleum and Minerals College of Computing and Mathematics Computer Engineering Department

COE 292 Class Project

|  |  |  |  |
| --- | --- | --- | --- |
| Title: Resume Parser | | | |
| *Group Members* | | | |
| **Name** | ***ID#*** | **Department** | **Signature** |
| *Shabib Aldawsari* | *201844020* | *ICS* | *SA* |
| *Tariq Madkhali* | *201920030* | *ICS* | *TM* |
| *Sultan Mashyakhi* | *201778630* | *ICS* | *SM* |
| *Yaman Shullar* | *201954350* | *COE* | *YS* |

By signing this document, the authors affirm that they comply fully with the published rules and regulations of the collage of computing and mathematics policies and the work that has been presented is their own work. We also grant the university the permission to share this document, or any material submitted hereafter with third parties including external evaluators and observers. In addition, we grand permission to King Fahad University of Petroleum and Minerals to reproduce the submitted material in part or whole in any form or type (paper or electronic) and use it for the purpose of advertising and showing the impact on student education.

*(11/13/2021)*

# Goals and Objectives

**The goal of this project is to identify the most suited candidates for the job. This will be achieved by detecting certain keywords that are desired for the position in a resume and scaling candidates according to their favorability from 0 to 10. However, some candidates will abuse this functionality by stuffing their resumes with these keywords.**

**The program will use the SVM algorithm to filter the candidates that are abusing it.**

# 

# Expected Achievement List

# Filter candidates that are stuffing their resume with keywords.

# Eliminate the unqualified candidates.

# Help the HR department to identify the potential candidates.

# Assign a final weightage score to each resume from 0 (least favorable) to 10 (most favorable).

# Used AI algorithm

# SVM 🡪 To distinguish valid and invalid resumes

# K-NN 🡪 To group keywords for each domain

# Neural Networks (Potentially) 🡪 We may use NN to get a better result if the results of K-NN were not satisfactory (Specifically, to separate candidates based on their favorability)

# Project Plan

# Phase 1: Clean the data

# Phase 2: Use SVM to determine which resume is invalid

# Phase 3: Use K-NN to cluster keywords for each domain

# Phase 4: split candidates based on their favorability from 0 to 10

|  |  |
| --- | --- |
| Name | Contribution for each phase |
| Shabib Aldawsari | %25 |
| Tariq Madkhali | %25 |
| Sultan Mashyakhi | %25 |
| Yaman Shullar | %25 |

# After we gathered, we concluded that it is better for the team if we all contributed to each task equally.

1. **Raw Data set**

Identification: We took our data from Kaggle (provided in the project statement), and it is stored in an excel file and consists of 2 columns:

Column 1 🡪 Category:

* Job title.

Column 2 🡪 Resume:

* A description of each candidate, containing his/her education, previous experiences, fields of interests and skills.

Limitations:

As stated in the project statement, this data is in a text form meaning that it is not easy to pre-process, hence, it is very difficult to recognize a pattern. This will affect the ability to find outliers (if any), categorizing data, and cleaning it from any other impurities. This is because of the large size of a considerable number of the resumes and lots of diverse and odd symbols appearing in mostly every resume.

Visualization:

Text

Description automatically generated

Figure 1: General information about the data set

Chart, pie chart

Description automatically generated

Figure 2: Category distribution

(Source: <https://www.kaggle.com/gauravduttakiit/resume-screening-using-machine-learning>)

Text

Description automatically generated

Figure 3: The contents of the second column (Resume)

1. **Cleaned Data set**

Cleaning: we used regular expressions to remove any odd symbol. After a thoughtful discussion and analysis, we concluded that there are no outliers (excluding keywords stuffers) in this data set.

Transformations: We tried to reformat the data to make it clearer using normal statements and functions (split, for loop, isalnum etc.….), but we saw that there is a better methodology to do that from Kaggle. We also encoded the first column (category). Finally, we normalized each word (the second column: Resume) in the resumes.

Techniques used:

1. TfidVectorizer 🡪 to normalize the data and extract features from it (This will help K-NN, as it can only deal with scaled numbers rather than features in the text form)
2. Words processing techniques from NLTK library (Will help in general)
3. Regular expressions 🡪 to remove odd symbols and unwanted data like e-mails, mentions etc.…. (This will help SVM and K-NN to work properly by providing only letters and normal symbols in the text)

Graphical user interface, text, application, email

Description automatically generated

Figure : Resumes before and after cleaning

1. **Technical task distribution**

As we mentioned in report 1, we decided that all team members should be included in each phase of the project. Although some of the members will have a greater contribution than the others in some phases, all members are included in each phase. As we concluded that diverse ways of thinking may benefit the project. The following table shows the tasks:

|  |
| --- |
| Cleaning data (including removing outliers, impurities etc.….) |
| Transformations (including reformatting, scaling, encoding etc.….) |
| Applying SVM (including training, testing etc.…) |
| Applying K-NN (including training, testing etc.…) |
| Applying NN (including training, testing etc.…) (if needed) |
| Final stage (testing the completed program) |

Expected enhancements: The use of SVM to eliminate stuffers and recommend the most qualified candidates for a particular job. We may also apply NN to determine the favorability of candidates.

1. **Short background on the AI algorithm**

Algorithms selected:

1. SVM
2. K-NN
3. NN (Not needed)

SVM: Originally used for problems in binary classification, where it is required to distinguish an object belonging to a category from others. (As in the slides). As stated, we decided to chose SVM as it has an impressive ability to classify under several consequences. Mainly, we will use it to distinguish valid and invalid resumes (stuffers and non-stuffers) by identifying a certain range of words occurrences in positions in each resume that we ASSUMED to be the most important positions. Further details will be provided subsequently.

K-NN: It can perform splendidly when it comes to cluster problems by comparing test data with the training set and then classify similar data in one cluster based on the nearest data (most similar in the set based on attributes). Hence, we selected this algorithm to help us to identify the domain the candidates are suited for. By making a group of keywords for each domain and then compare the test resume of domain (x) with the group of keywords in the corresponding domain.

1. **Result and discussion**

SVM results:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

In the above snapshots: We had first to clean the resumes from odd symbols so that we can classify stuffers from non-stuffers (symbols affected this process). After cleaning, we trained SVM to distinguish stuffers from non-stuffers by choosing the following range of words (number of words after description (in the resume), which contains somehow important information, is less than 600 and the number of words before the skills details, which contains the skills of the candidate, is less than 125). Please note that 125 and 600 are not magic numbers, they are ASSUMED by the members after estimating the average of each case on the internet (that is, different sources stated different numbers of skills and general description. Our estimation for the situation went to 125 and 600). As can be seen from the second snapshot, most of the candidates in the data set are not stuffers. It can be noted that this result may change depending on the range selected. In our case, these are the results.

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

The above snapshots simulate how SVM would work (classify) on newly entered data.

Table

Description automatically generated

The snapshot above shows how accurate K-NN in estimating the category (job title) of a newly inserted data (candidate). K-NN will utilize TfidfVectorizer class to extract features from each resume. It will be trained based on relating certain number of features with each job title and then it compares a newly inserted resume’s features with the training set to determine to which cluster it belongs. This high percentage is understood to be normal as this kind of problems (direct comparison and count) require nothing special (like NN) to be well done.

A picture containing calendar

Description automatically generated

Table

Description automatically generated

The snapshot above shows how a number is assigned to a resume (which indicates it is favorability among other candidates). This will be done by comparing the most frequent words in resumes of domain (x) with words in the resume.

1. **What could have been done better**

* We may assign a weight for every keyword, which will enhance the process of determining the favorability of candidates. This can be done by using Neural Networks. We could also use a window instead of a keyword of variable length and assign a weight for it.
* We could use gradient decent to optimize the number of keywords and skills for SVM.