PHASE 2: EDA AND FEATURE EXTRACTION

1. BASIC EDA (EXPLORATORY DATA ANALYSIS)

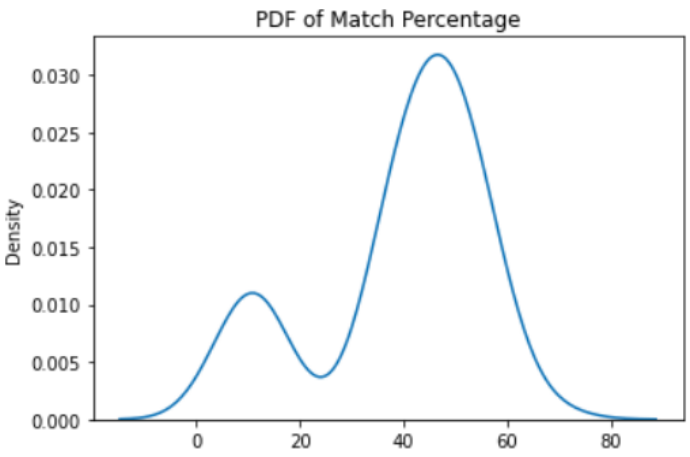
In basic eda, we will explore the given data only. That means we check the high-level stats of the given data.

1.1. Job Description

- The job description pdf is of 2 pages.
- It has required work experience, education qualification and must have and nice to have skills mentioned.
- This is a typical job description for machine learning engineer role.

1.2. Resumes

- We have 90 data points.
- All the resumes are in pdf format.
- We have the resumes and their corresponding match percentage.
- There are two columns in the csv file,
    - CandidateID : which same as the resume file name.
    - Match Percentage : it is between 0 to 100.
- There is no missing data.
- There is no duplicate data.
- Minimum match percentage is 4.81
- Maximum match percentage is 69.21
- Mean of match percentage is 39.645
- Median of match percentage is 44.65
- There are no irregularities or outliers with match percentages.
- Match percentage is a bimodal distribution.
- There is no resume with Match percentage between 15 and 35.



- There are some spaced words like “D A T A S C I E N C E”, which we need to take care.
- Also there are short forms mostly of education qualifications like B.Tech.

1.3. Univariate and Bi-variate Analysis

- With the PDF of resumes's word length and character length, we can be certain that there are no irregularities in the resumes.
- There are minimum of 529 and maximum of 1543 characters in a resume.
- Character length is approximately gaussian distributed.
- There are minimum of 73 and maximum of 213 words in a resume.
- Word length is approximately gaussian distributed.
- There is very weak correlation between Resume character length and Match Percentage.
- There is very weak correlation between Resume word length and Match Percentage but it is better than character length.

2. DATA CLEANING

In real-world, we do not find data in a clean format. So we do various data cleaning operations to make sure that we have good quality data. Because if we do not have good quality data then whatever model we make, it will perform poorly.

- I have converted everything to lower case so that words like “Data”, “data” and “DATA” are treated the same.
- I have replaced unusual quotes like ’ with ' (quote).
- I have replaced the new lines with spaces.
- I have concatenated words like "D A T A S C I E N C E" to get "DATA SCIENCE".
- I have also removed hyperlinks.
- I have converted education degrees like B.Tech or BTech to a specified form.
- I have converted skills with special symbols to words like C++ to words cplusplus
- I have replaced non-alphanumeric characters with space.
- I have removed the stop words of english.
- I have removed inflections with the word lemmatizer.

3. FEATURE EXTRACTION

Feature extraction is the art part of data science and it is the most useful skill. If we do good feature extraction then a simple model might outperform a complex model. I have created 12 interesting features.

Since I have only one job description, I did not create a few more features that could have been related to the job description and resume. Like difference of word length between job description and resume.

I have created the following features,

- **resume_word_num :** total number of words in resume
- **total_unique_word_num** : total number of unique words in job description and resumes
- **common_word_num :** total number of common words in job description and resumes
- **common_word_ratio :** total number of common words divided by total number of unique words combined in both job description and resumes

- **common_word_ratio_min :** total number of common words divided by minimum number of unique words between job description and resumes
- **common_word_ratio_max :** total number of common words divided by maximum number of unique words between job description and resumes
- **fuzz_ratio :** fuzz.WRatio from fuzzy wuzzy library
- **fuzz_partial_ratio :** fuzz.partial_ratio from fuzzy wuzzy library
- **fuzz_token_set_ratio :** fuzz.token_set_ratio from fuzzy wuzzy library
- **fuzz_token_sort_ratio :** fuzz.token_sort_ratio from fuzzy wuzzy library
- **is_fresher** : wheather a candidate is fresher or experienced
- **from_reputed_college :** wheather a candidate is fresher from reputed college

3.1. Univariate Analysis

For univariate analysis, we have various options like visualization of pdf, boxplot, violin plot, histogram. I have used pdf to get insight of each features. Along with the pdf I have also printed some stats like min, max and mean value to understand better about the distribution.

- resume_word_num is approximately log normally distributed with mean 102.789
- Minimum resume_word_num is 63 and maximum resume_word_num is 168
- total_unique_word_num is distributed widely with mean 194.067
- Minimum total_unique_word_num is 172 and maximum total_unique_word_num is 227
- common_word_num is approximately gaussian distributed with mean 19.71
- Minimum common_word_num is 9 and maximum common_word_num is 36
- common_word_ratio is approximately gaussian distributed with mean 0.10
- Minimum common_word_ratio is 0.045 and maximum common_word_ratio is 0.17
- common_word_ratio_min is approximately gaussian distributed with mean 0.245

- Minimum common_word_ratio_min is 0.118 and maximum common_word_ratio_min is 0.245
- common_word_ratio_max is approximately gaussian distributed with mean 0.148
- Minimum common_word_ratio_max is 0.068 and maximum common_word_ratio_max is 0.27
- fuzz_ratio has bi-modal distribution with mean 74.356
- Minimum fuzz_ratio is 48 and maximum fuzz_ratio is 86
- PDF of fuzz_partial_ratio is falling slowly and has mean 44.956
- Minimum fuzz_partial_ratio is 43 and maximum fuzz_partial_ratio is 48
- fuzz_token_set_ratio is approximately gaussian distributed with mean 52.867
- Minimum fuzz_token_set_ratio is 41 and maximum fuzz_token_set_ratio is 61
- fuzz_token_sort_ratio is approximately gaussian distributed with mean 50.878
- Minimum fuzz_token_sort_ratio is 40 and maximum fuzz_token_sort_ratio is 59
- There are about 28 freshers out of 90 candidates.
- For both freshers and experienced the pdf of match percentage peak at same points.
- That means the match percentage is not affected by wheather a candidate is fresher or experienced.
- There are about 14 candidates are from reputed colleges out of 90 candidates.
- For both type of candidates the pdf of match percentage peak at same points.
- That means the match percentage is not affected by wheather a candidate is from reputed or non reputed college.
- This can be because candidates from different domain might have also applied for the job. Otherwise, we could have seen some high match percentage for experienced and/or candidates from reputed colleges.

3.2. Bi-variate Analysis

For bi-variate analysis I have plotted scatter plot between each feature and the output. And also I have used the SRCC (Spearman Rank Correlation Coefficient) to check for correlation between the features and the output. The advantage of the scatter plot is that we can see the correlations clearly. Also, the SRCC help in quantizing the correlation. Since this is a regression task there are not many options for multi-variate analysis.

- There is very weak correlation between resume_word_num and Match Percentage. And the SRCC (Spearman Rank Correlation Coefficient) is 0.285
- The SRCC between total_unique_word_num and Match Percentage is -0.02. Since it is very close to zero we can say that there is not any correlation between total_unique_word_num and Match Percentage.
- There is some correlation between common_word_num and Match Percentage and the SRCC (Spearman Rank Correlation Coefficient) is 0.512
- The SRCC between common_word_ratio and Match Percentage is 0.541 which is a good.
- The SRCC between common_word_ratio_min and Match Percentage is 0.491
- The SRCC between common_word_ratio_max and Match Percentage is 0.512
- The SRCC between fuzz_ratio and Match Percentage is -0.19. There isn't much correlation. But if we ignore the fuzz ratios above 80 then we can see a good correlation.
- The SRCC between fuzz_partial_ratio and Match Percentage is 0.344 which is again a very weak correlation.
- The SRCC between fuzz_token_set_ratio and Match Percentage is also very small and is 0.354.
- The SRCC between fuzz_token_sort_ratio and Match Percentage is 0.501, which is good enough.
- The features related to common words have a good correlation with the output.
- But the common word features are also correlated with each other.
- Specially the common_word_num, common_word_ratio and common_word_ratio_max have very strong correlation.

- So I have removed common_word_ratio and common_word_ratio_max columns.
- So we have 10 features.
- The fuzzy wuzzy features except fuzz_token_sort_ratio does not show any promising correlation.

4. FEATURE ENCODING

For feature encoding we have various options like BoW, TF-IDF, Word2Vec, Bert based, etc. For the initial phase I have opted binary bag of words and average word2vec.

We will do advanced feature encoding like sentence bert later in this project. To check if it can show a better result than the base model.

What is Binary BoW?

- In Binary BoW, we create vector, based on presence or absence of a word. If the word is present then the corresponding cell will be 1 and if not then it will be 0. The encodings will be very sparse and high dimensional. The count BoW has counts of occurrences of a word in the document.

Why Binary BoW?

- I think the Binary BoW is a very good option because the count or the frequencies are not as important as weather a word (which could be skill) is present or not. I believe that it may outperform even bert based encodings for this task.

4.1. Binary BoW

- I have used uni-gram, bi-gram and tri-gram to get some sequence information as well.
- And I have used both job description and resume text to create the vocabulary.
- The minimum document frequency is 4. So it help in removing some non-useful words like names of candidates.

- The maximum document frequency is 99%. That means word which are very frequent will be ignored.
- The vocab size is 716. So we will get 716 dimensional output for both job description and resumes.
- We have created two new features cosine_similarity and euclidean_distance.
- **cosine_similarity** : It represents cosine similarity score between sentence embeddings (based on BoW) of job description and resume.
- **euclidean_distance** : It represents euclidean distance between sentence embeddings (based on BoW) of job description and resume.
- Now we have total of **12** extracted features.
- And the total feature dimension is 12+716+716 = **1444**

4.1.1. Univariate and Bi-variate Analysis

- cosine_similarity is approximately gaussian distributed with mean 0.230
- Minimum cosine_similarity is 0.117 and maximum cosine_similarity is 0.355
- PDF of euclidean_distance is falling very sharply on both side and has mean of 12.724
- Minimum euclidean_distance is 11.532 and maximum euclidean_distance is 13.928
- The SRCC between cosine_similarity and Match Percentage is 0.506, which is good.
- But the SRCC between euclidean_distance and Match Percentage is only -0.135. So we can say that there is no correlation.

4.2. Average Word2Vec

What is Average Word2Vec?

- The word2vec produces dense word embedding usually of small size like 300 dimensions. I have used the pre-trained model on google news data. I did not have much data that's why I did not train my own word2vec model. After getting the word embeddings of each word in the document. I have summed them and divided by the number of total words to get average word2vec representation.

Why Average Word2Vec?

- The Word2Vec is used to be the state of the art for word embeddings. I have used it to compare with the BoW. It would have performed better if it was trained on similar dataset

- We have created two new features cosine_similarity and euclidean_distance.
- **cosine_similarity** : It represents cosine similarity score between word embeddings (based on w2v) of job description and resume.
- **euclidean_distance** : It represents euclidean distance between word embeddings (based on w2v) of job description and resume.
- Now again we have total of **12** extracted features.
- And the total feature dimension is 14+300+300 = **612**

4.2.1. Univariate and Bi-variate Analysis

- cosine_similarity is approximately gaussian distributed with mean 0.171
- Minimum cosine_similarity is 0.083 and maximum cosine_similarity is 0.291
- PDF of euclidean_distance is falling very sharply on both side and has mean of 15.578
- Minimum euclidean_distance is 14.036 and maximum euclidean_distance is 17.493
- The SRCC between cosine_similarity and Match Percentage is 0.495, which is good.
- But the SRCC between euclidean_distance and Match Percentage is only -0.03. So we can say that there is no correlation.
- cosine_similarity and euclidean_distance features are very similar for both feature encodings.

5. HIGH-DIMENSIONAL DATA VISUALIZATION

Since this is a regression task. The high-dimensional data visualization does not make much sense.

We could have used PCA or t-SNE to reduce the dimensionality to visualize. But I have ignored it because it is not relevant with this problem.

6. REFERENCES

- Chouhan, Sanjay. Towards data science. The Quora Question Pair Similarity Problem. 2021.
- FuzzyWuzzy. SeatGeek. [https://github.com/seatgeek/fuzzywuzzy]
- AppliedRoots. [https://www.appliedroots.com/]