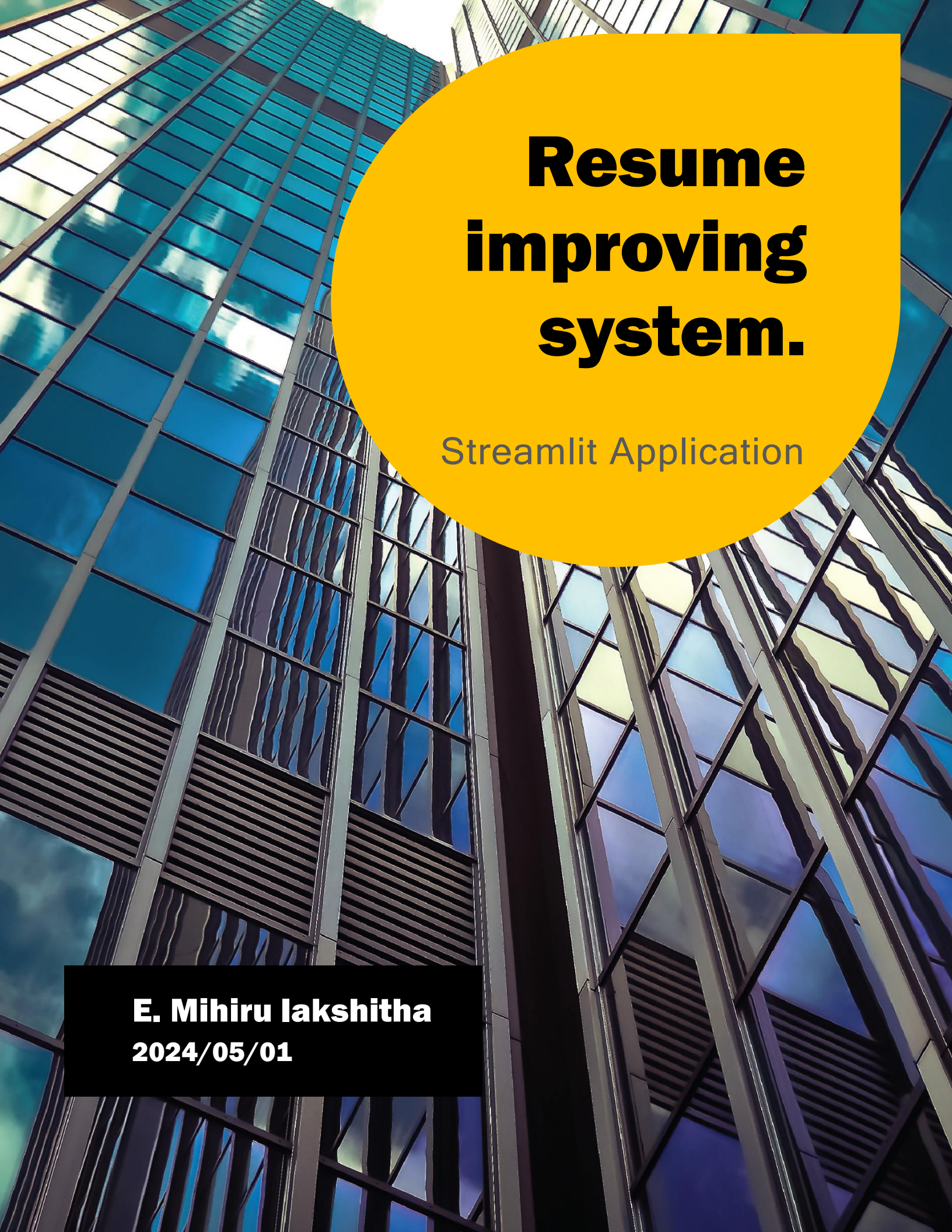


# Resume improving system.

Streamlit Application

**E. Mihiru lakshitha**

2024/05/01

# Table of content

**Real-World Text Analytics Application Assignment**

The objective of this assignment is to apply the concepts learned in the Text Analytics module to a real-world problem of your choice. Through this assignment, you will gain practical experience in preprocessing text data, applying various text analytics techniques, and visualizing and interpreting the results.




## Abstract

In today's fiercely competitive job market, the first hurdle for job seekers lies in the initial screening of resumes. With the advent of automated systems, resumes often undergo filtration processes even before human eyes get a chance to review them, resulting in numerous missed opportunities for candidates. Recognizing this challenge, this assignment proposes an innovative solution designed to empower job seekers through the application of advanced Natural Language Processing (NLP) techniques.

The primary objective is to create a comprehensive system that offers a proactive approach to resume optimization. By allowing users to seamlessly upload their resumes, the system facilitates an instant evaluation process, providing valuable feedback on how well their skills and experiences align with their targeted job roles. This feedback loop is crucial for candidates to tailor their resumes effectively and increase their chances of securing interview opportunities.

The solution architecture encompasses a series of crucial steps, starting with exploratory data analysis to gain insights into resume data patterns. This is followed by meticulous data preprocessing, which involves techniques such as text cleaning, encoding, and tokenization to prepare the data for model training. The heart of the system lies in the machine learning model, leveraging advanced algorithms like Bidirectional Long Short-Term Memory (LSTM) to analyze and interpret resume content. Upon successful model training, the system becomes adept at predicting suitable job roles based on the content extracted from uploaded resumes. This predictive capability is instrumental in providing job seekers with actionable insights and recommendations, enabling them to iteratively refine their resumes until they align perfectly with their desired positions.

Furthermore, the system incorporates visualization tools to enhance user understanding and evaluation of the generated recommendations. These visualizations serve as intuitive aids, allowing job seekers to grasp the underlying trends and patterns in their resume data more effectively.

# Introduction

In today's fast-paced and highly competitive job market, the initial screening of resumes has emerged as a critical bottleneck for job seekers worldwide. Automated systems are being used more and more to screen resumes before humans even see them. This means a lot of good candidates might not even get a chance. We need new ideas to help job seekers get past these systems and find the right jobs.

**Problem**:

The old way of applying for jobs and sending resumes is getting outdated as companies change how they hire. Automated systems for screening resumes can handle lots of applications quickly, but they might not understand candidates well enough to judge if they're right for the job. This means good candidates might not get noticed, which can be really frustrating and discouraging. So, ways to guide candidates to improve their resumes are essential.

**Aim**:

This study focuses on tackling those challenges by using Natural Language Processing (NLP) techniques related to text classification. With NLP, we want to create a strong system that helps job seekers improve their resumes and boosts their chances of getting interviews. **This system will give immediate feedback on what job role is suitable for applying with their resume** to avoid automated filtration using included candidate's details.  This way, job seekers can apply more strategically and target the right positions. This system should have ability to classify the matching job roles to given Resume,

**Process:**

Users can log into the web application, upload their resume, and click the "Analyze" button. Then, based on the provided CV data, a trained deep learning model will predict the suitable job role for the given information. **If the predicted job role is not what the user wants to apply for, they should fine-tune and improve their resume, and upload it again**. This process should be repeated until they receive the desired job role as prediction.

**Background**:

While using technology to improve recruitment isn't new, recent progress in NLP and machine learning is changing the game. These advancements offer exciting opportunities to transform how resumes are assessed and how candidates find jobs that fit them well. We want to use this technology to create a solution that not only helps with current challenges but also stays ahead of future trends in hiring. By using data, smart algorithms, and designs focused on users, we hope to give job seekers more control over their careers in a tough job market.

# Data acquisition & EDA

**Dataset**:

Dataset under consideration is sourced from Kaggle, a renowned platform for datasets and machine learning competitions. This dataset is unique in that it comprises labeled data, with each entry containing a "Category" column representing target job roles and a "Resume" column containing the textual content of resumes. Notably, this dataset encompasses an extensive array of job roles, numbering around 24 distinct categories, with each category containing more than 1000 rows of data.

| | Category | Resume |
|---|---|---|
| 0 | Data Science | Skills * Programming Languages: Python (pandas... |
| 1 | Data Science | Education Details \r\nMay 2013 to May 2017 B.E... |
| 2 | Data Science | Areas of Interest Deep Learning, Control Syste... |
| 3 | Data Science | Skills â□¢ R â□¢ Python â□¢ SAP HANA â□¢ Table... |
| 4 | Data Science | Education Details \r\n MCA YMCAUST, Faridab... |

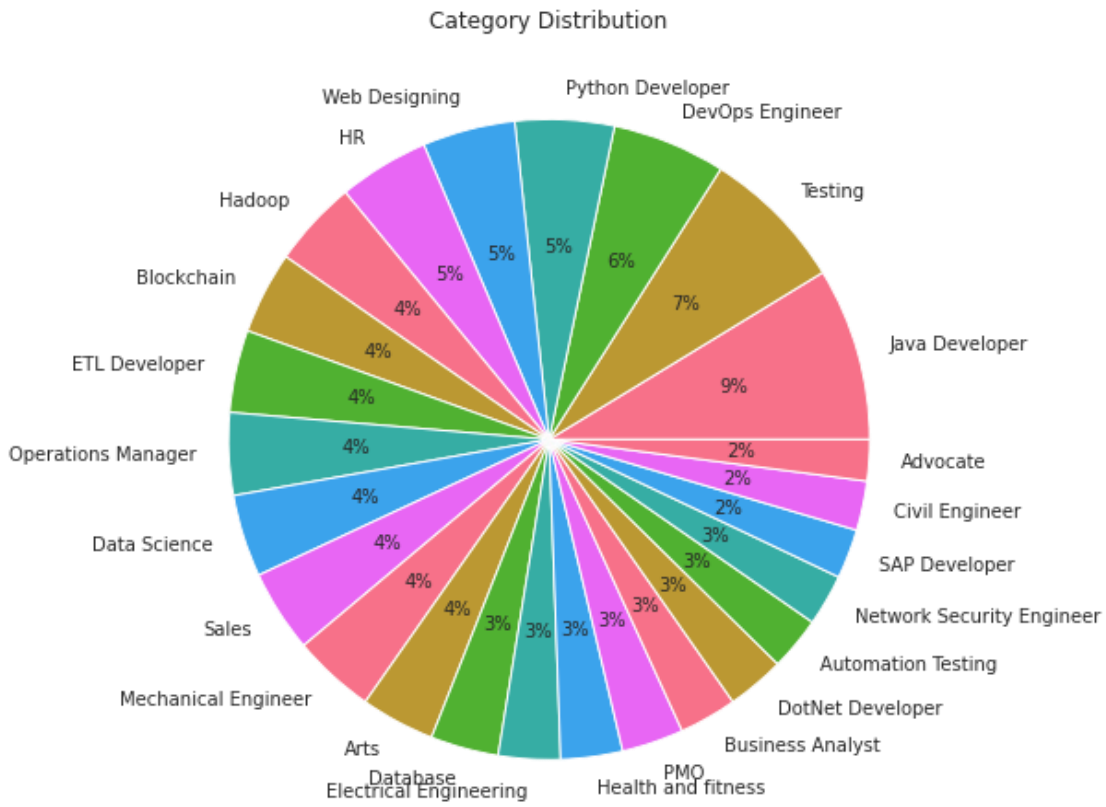
**Exploratory data analysis**

Exploratory Data Analysis (EDA) serves as a crucial step in understanding the structure and characteristics of the dataset, providing valuable insights into the distribution of categories and the most frequent words within the textual data. One of the primary objectives of EDA is to examine the distribution of categories within the dataset.

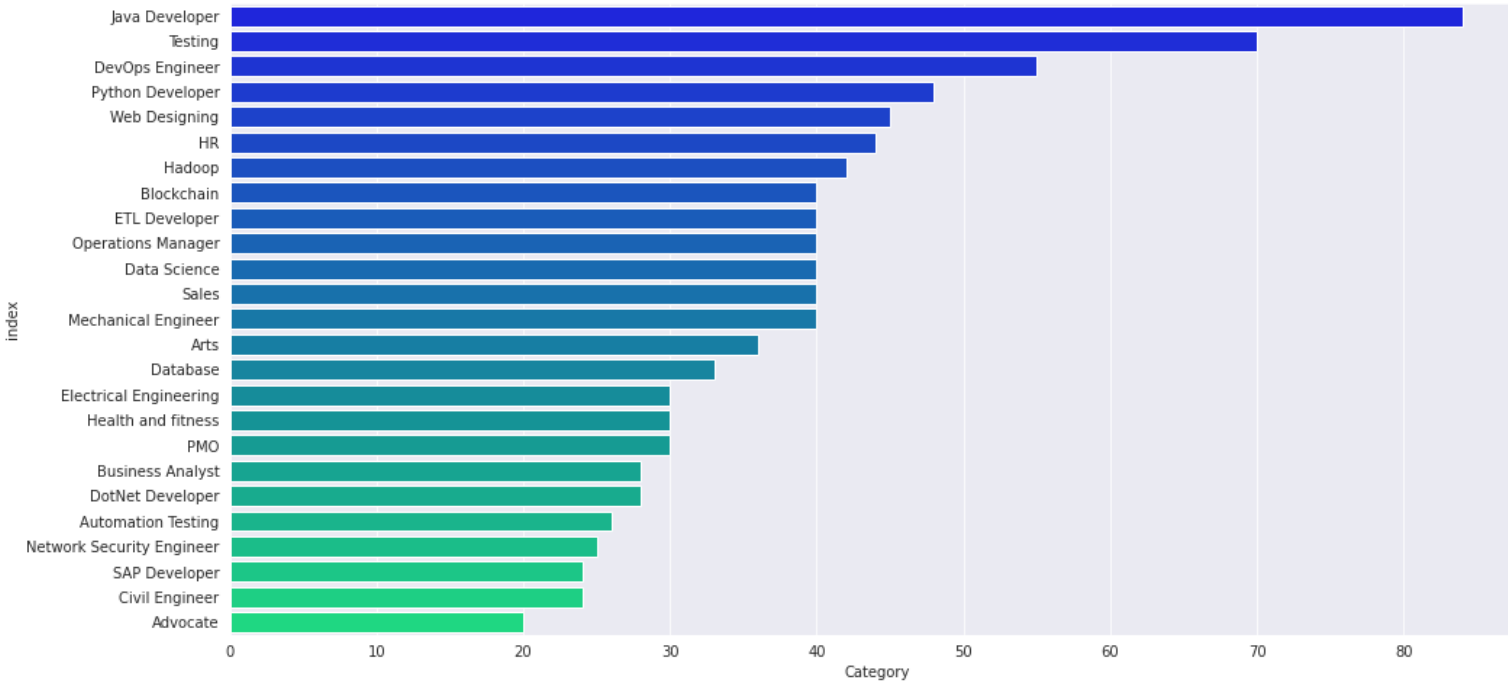
In this case, the dataset contains a diverse array of job roles, including Java Developer, Testing, DevOps Engineer, Python Developer, Web Designing, and more. Notably, the distribution appears to be relatively even, with each category represented by a substantial number of entries. This balanced distribution ensures that the dataset is not skewed towards any specific job role, allowing for a comprehensive analysis of different categories. Below is top 4 job roles with count sum.

| | index | Category |
|---|---|---|
| 0 | Java Developer | 84 |
| 1 | Testing | 70 |
| 2 | DevOps Engineer | 55 |
| 3 | Python Developer | 48 |

Below pie chart will be helpful for get idea of percentage distribution of the categories.



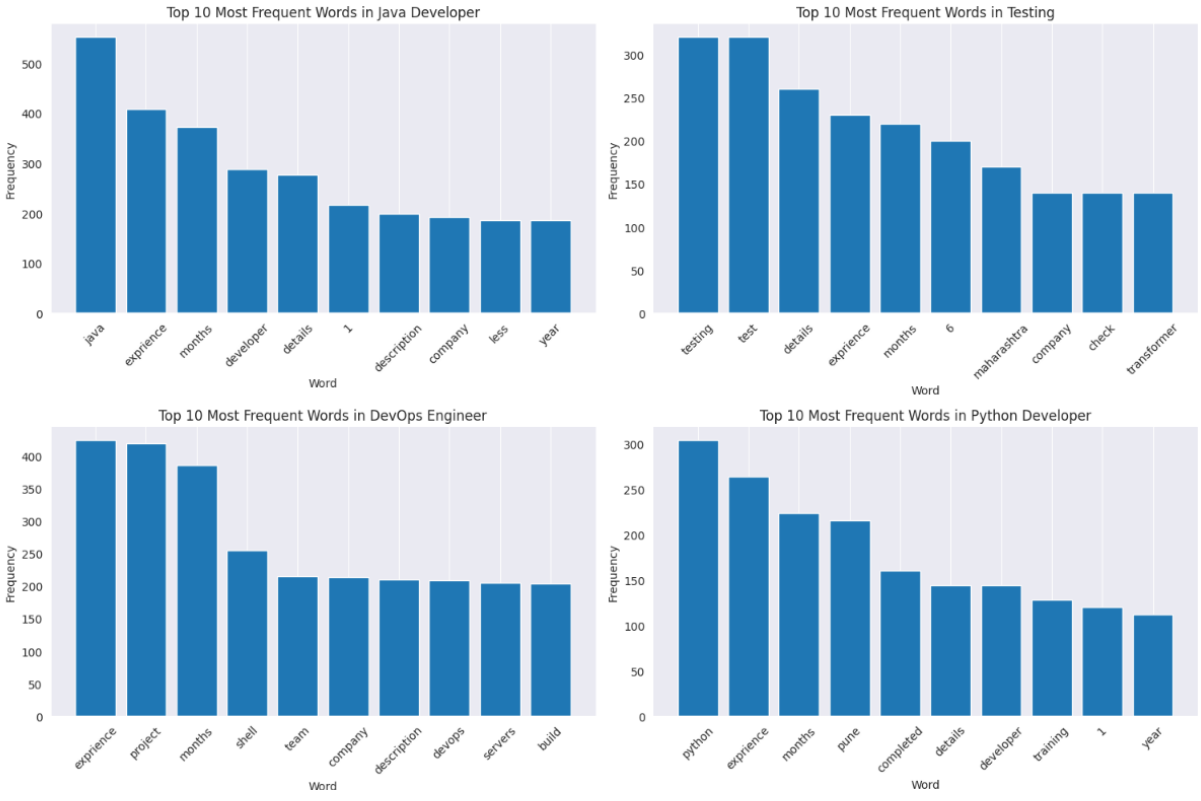

The above pie chart suggests that Dataset has a bit of imbalance distribution of target categories ranging from 2% to 9%. So, we can conclude that the dataset has class-imbalance issue. Below graph also provides evidence on above claim.




So, according to this bar graph we can observe that, java developer, testing, Devops categories have some kind of high frequency in the dataset compared to others. So, we have to take necessary actions to avoid bias of the model to most frequency classes.

Another key aspect of EDA is the exploration of textual data, particularly the identification of the most frequent words. By visualizing word frequencies in a word cloud, we can gain insights into the recurring themes and topics present in the resume texts. In this dataset, common words such as "experience," "months," "skills," "company," and "details" emerge as the most frequent. These words provide valuable cues about the content and structure of resumes,


And below Diagrams show the top 10 most frequent words in top four categories. Which helps us to understand most likely words in each category resumes.



# Data Pre-processing

Data preprocessing is a critical step in preparing textual data for analysis and modeling. In this context, various techniques such as tokenization, lowercasing, removing punctuation and stop words, lemmatization or stemming, and spelling correction are applied to ensure that the data is clean, consistent, and ready for further processing.

### Tokenization

Tokenization is important because it breaks down the text into individual tokens, such as words or phrases, which serve as the basic building blocks for analysis. By tokenizing the text, we can better understand its structure and extract meaningful insights. In the provided data, tokenization is applied to split the resumes into individual words, enabling subsequent processing steps.

### Lowercasing

Lowercasing is essential for standardizing the text by converting all characters to lowercase. This ensures uniformity and consistency in the text data, preventing issues such as case sensitivity during analysis and modeling. In the preprocessing pipeline, lowercasing is performed to normalize the text and facilitate easier comparison and matching of words.

### Removing punctuation

Stop words, and special characters help in cleaning the text and removing noise that may interfere with analysis. Punctuation marks and special characters are often irrelevant for text analysis tasks and can be safely discarded. Stop words, such as "and" "the," and "is," are common words that occur frequently in text but typically carry little semantic meaning. By removing them, we can focus on the most informative words and improve the quality of analysis results. In the provided data, these steps are executed to enhance the clarity and relevance of the text content.

### Lemmatization or stemming

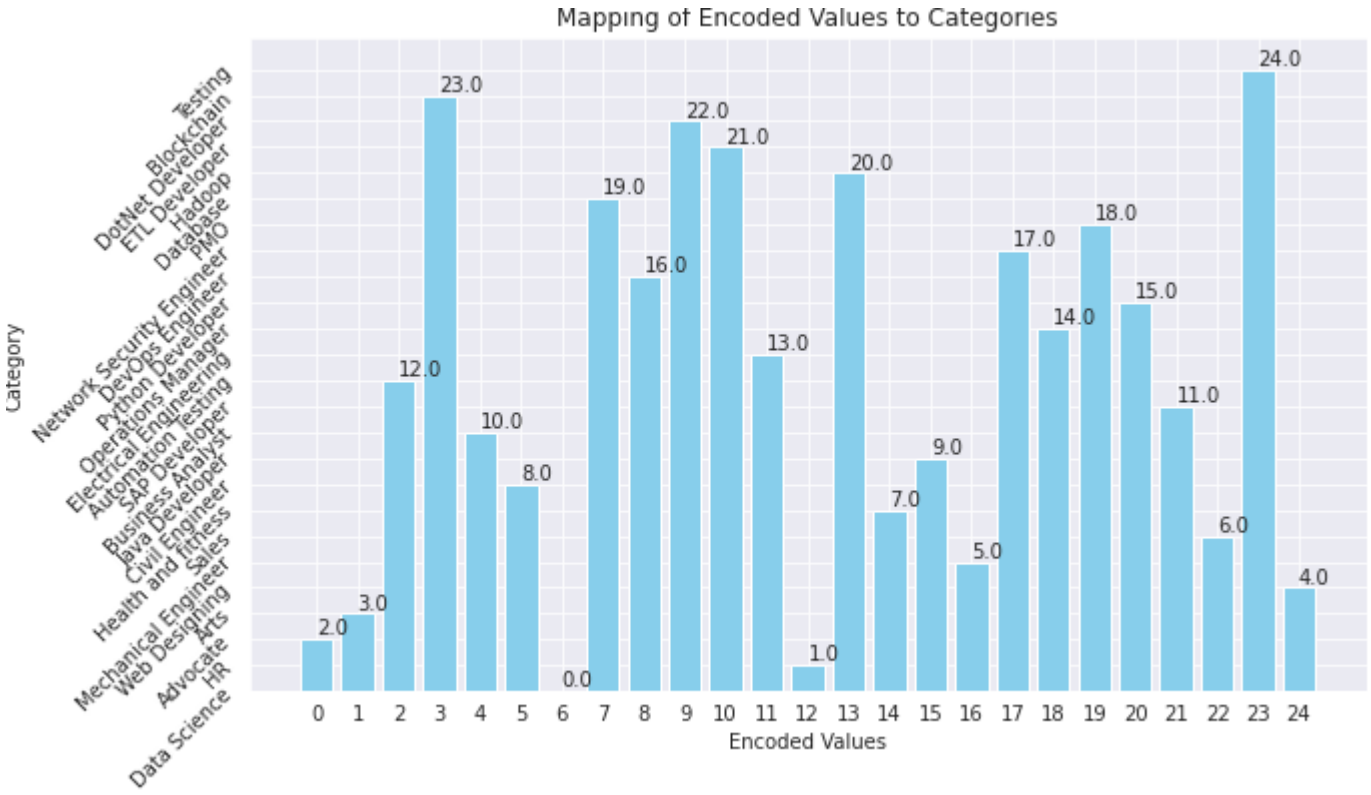
This is crucial for reducing words to their base or root forms, which helps in standardizing the vocabulary and reducing redundancy. By converting words to their base forms, we can consolidate variations of the same word and simplify the text representation. This makes it easier to identify patterns and extract meaningful insights from the data. In the preprocessing pipeline, lemmatization is applied to ensure that words are represented consistently across different forms.

**Spelling correction** is important for rectifying any spelling errors in the text, which can improve the accuracy and reliability of analysis results. Misspelled words may lead to incorrect interpretations or hinder the performance of natural language processing models. By applying spelling correction, we can mitigate the impact of spelling errors and ensure the integrity of the text data.

## Model building and feature engineering

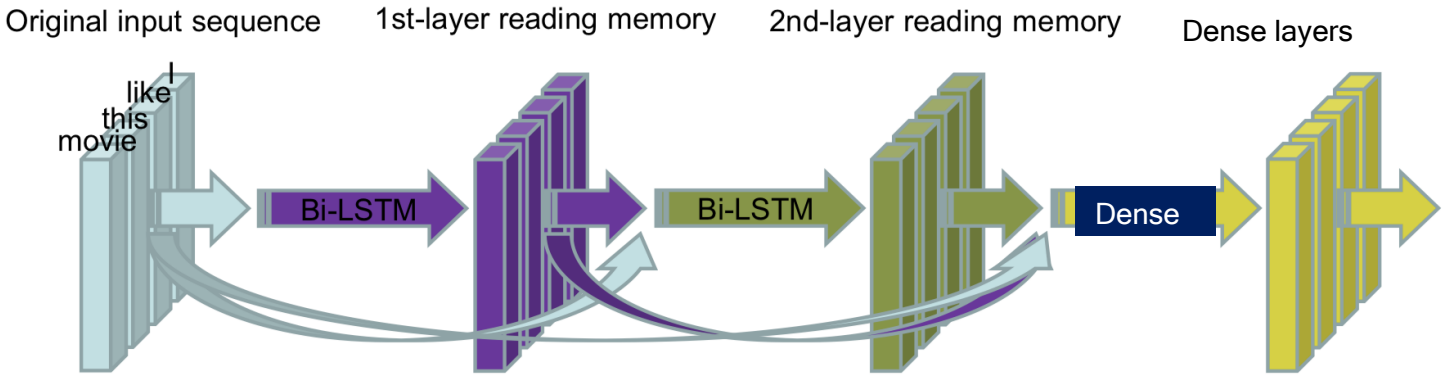
This step plays a crucial role in developing effective machine learning models for classification tasks. In the provided context, several techniques are applied, including label encoding, model architecture design, and evaluation, to build a robust classification model for predicting job categories based on resume text data.

**Label Encoding**: Before training the model, the categorical target labels (job categories) are encoded into numerical values using the LabelEncoder from the scikit-learn library. This step is essential as machine learning algorithms typically require numerical inputs. The encoded labels serve as the target variable for model training.




After the Label encoding, Label mapping has done in order to identify the original categories correctly. This will be helpful when we are predicting the actual categories.

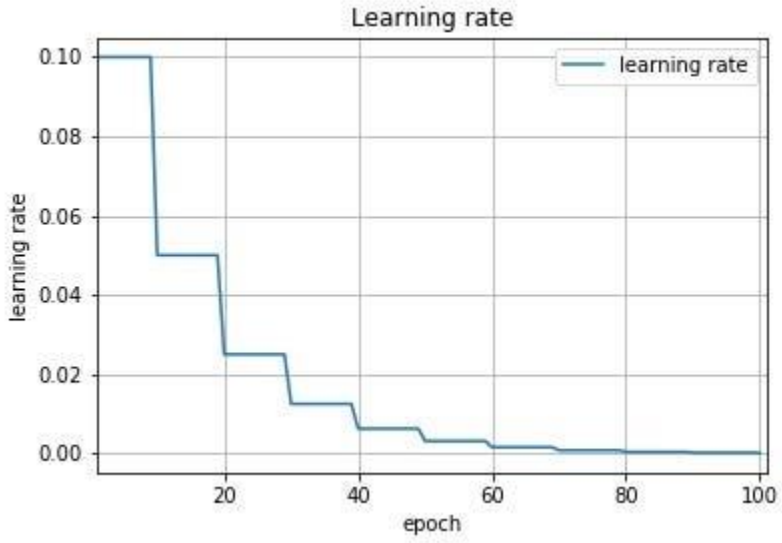
**Model Architecture**: The model architecture is designed using a deep learning framework with the Sequential API from TensorFlow. The architecture consists of an embedding layer, bidirectional LSTM layers, and dense layers. The embedding layer converts the input text data into dense vectors of fixed size, which are then fed into bidirectional LSTM layers to capture sequential information from the text bidirectionally. Additionally, dense layers with ReLU activation functions are added to learn non-linear relationships in the data. The final dense layer uses the softmax activation function to output probabilities for each job category.




*Visual Architecture of the model*

**Techniques Applied**: In addition to model architecture design, several techniques are applied during the model building process. Tokenization and padding are performed to convert the raw text data into sequences of numerical tokens, ensuring uniform input shapes for the model. Furthermore, early stopping and learning rate scheduling callbacks are implemented to prevent overfitting and optimize the learning process during training.

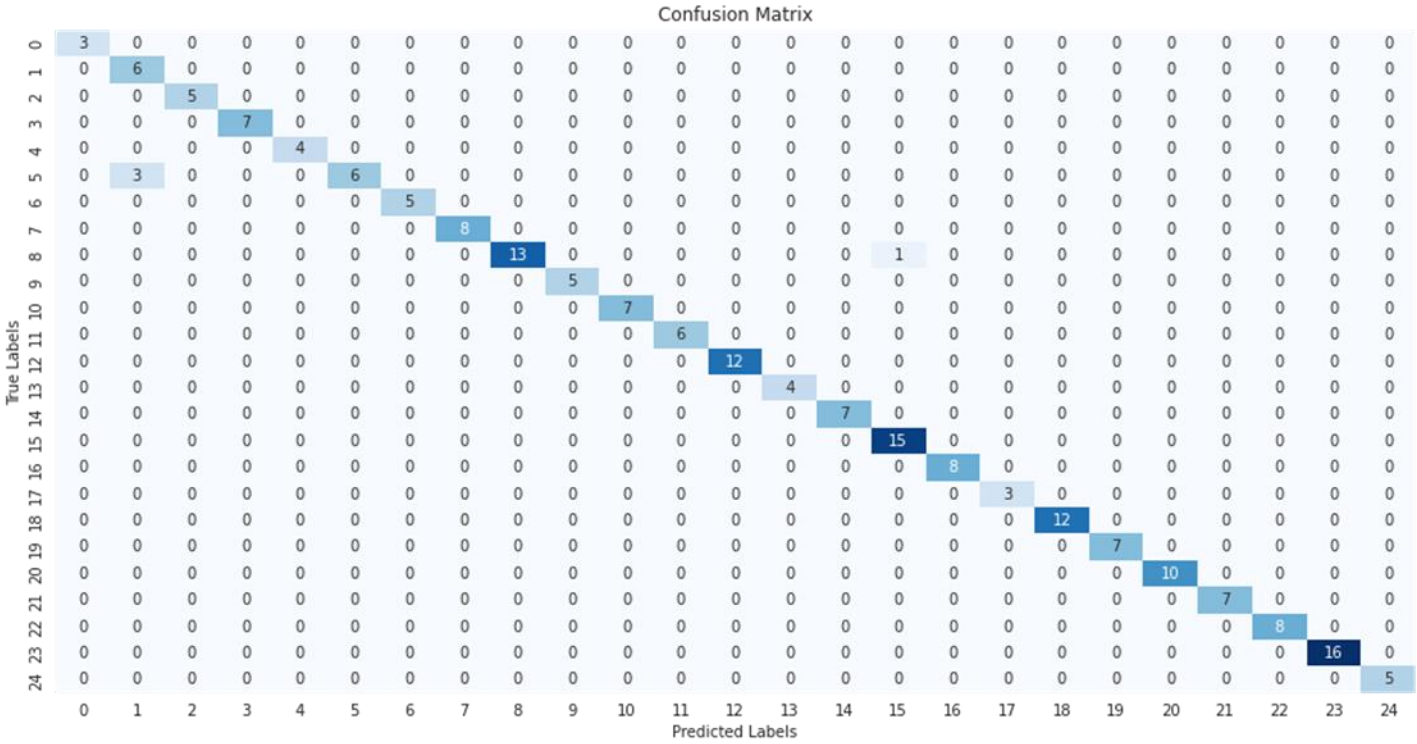
Under Learning rate scheduler , step based decay applied with 10% reduction of rate at each epoch.



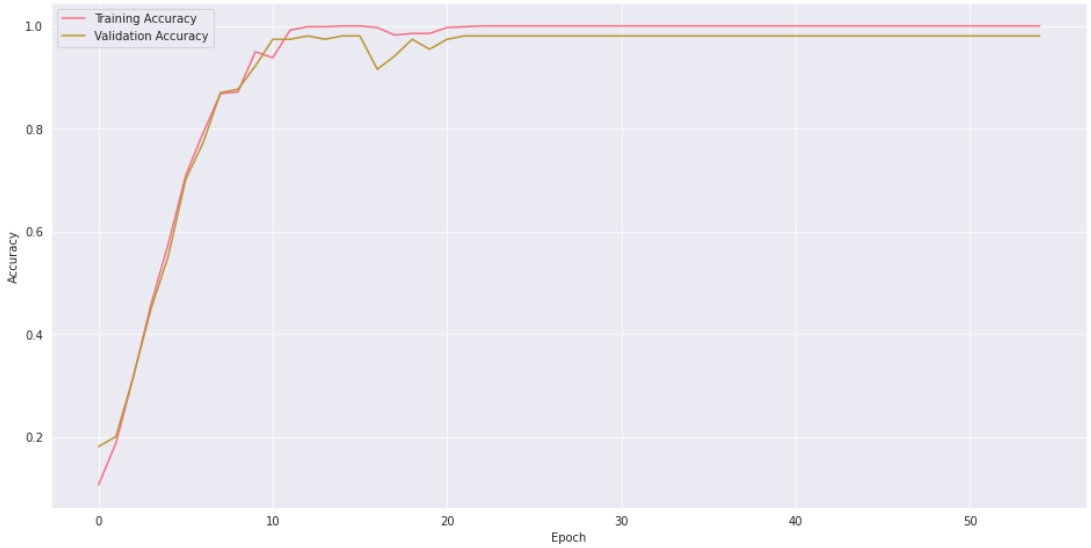
# Model Evaluation and results

**Confusion Matrix**:

The confusion matrix provides a detailed summary of the model's predictions compared to the actual labels. It shows the number of true positives, true negatives, false positives, and false negatives for each class. In this case, the confusion matrix indicates that the model's predictions align well with the actual job categories, with minimal misclassifications. The majority of the predictions fall along the diagonal, indicating correct classifications.




**Accuracy**: The overall accuracy of the model is 0.98, or 98%, which is a high accuracy rate. This metric measures the proportion of correct predictions out of the total number of predictions made by the model. A high accuracy score suggests that the model is effective in accurately predicting job categories from resume text data.



So the above graph shows us how the model accuracy converges to the 98% level.

**Macro Average**: The macro average is calculated by averaging the precision, recall, and F1-score across all classes. In this case, the macro average for precision, recall, and F1-score is also 0.98, indicating strong performance across all classes. This metric provides a balanced assessment of the model's performance across different categories, considering both false positives and false negatives.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 3 |
| 1 | 0.67 | 1.00 | 0.80 | 6 |
| 2 | 1.00 | 1.00 | 1.00 | 5 |
| 3 | 1.00 | 1.00 | 1.00 | 7 |
| 4 | 1.00 | 1.00 | 1.00 | 4 |
| 5 | 1.00 | 0.67 | 0.80 | 9 |
| 6 | 1.00 | 1.00 | 1.00 | 5 |
| 7 | 1.00 | 1.00 | 1.00 | 8 |
| 8 | 1.00 | 0.93 | 0.96 | 14 |
| 9 | 1.00 | 1.00 | 1.00 | 5 |
| 10 | 1.00 | 1.00 | 1.00 | 7 |
| 11 | 1.00 | 1.00 | 1.00 | 6 |
| 12 | 1.00 | 1.00 | 1.00 | 12 |
| 13 | 1.00 | 1.00 | 1.00 | 4 |
| 14 | 1.00 | 1.00 | 1.00 | 7 |
| 15 | 0.94 | 1.00 | 0.97 | 15 |
| 16 | 1.00 | 1.00 | 1.00 | 8 |
| 17 | 1.00 | 1.00 | 1.00 | 3 |
| 18 | 1.00 | 1.00 | 1.00 | 12 |
| 19 | 1.00 | 1.00 | 1.00 | 7 |
| 20 | 1.00 | 1.00 | 1.00 | 10 |
| 21 | 1.00 | 1.00 | 1.00 | 7 |
| 22 | 1.00 | 1.00 | 1.00 | 8 |
| 23 | 1.00 | 1.00 | 1.00 | 16 |
| 24 | 1.00 | 1.00 | 1.00 | 5 |
| accuracy | | | 0.98 | 193 |
| macro avg | 0.98 | 0.98 | 0.98 | 193 |
| weighted avg | 0.98 | 0.98 | 0.98 | 193 |

## Deployment (Web-application )

This marks the transition of a machine learning model from development to practical use, allowing users to interact with the model and benefit from its predictions. In this case, the model built for predicting job categories from resume text data has been deployed using a web application developed with Streamlit.

**User Interaction**: The web application provides a user-friendly interface where users can upload their resumes (CVs) directly. Upon uploading their CVs, users can trigger the analysis process by hitting the "Analyze" button.

**Prediction Process**: Behind the scenes, the deployed model utilizes the saved weights and architecture to predict the relevant job category based on the content of the uploaded CV. The model leverages Natural Language Processing (NLP) techniques to extract features from the resume text and make predictions.

**Output Interpretation**: Once the prediction is made, the web application presents the predicted job category to the user. Users can then review the predicted category and receive instructions on how to fine-tune their CVs to better align with the desired job roles.

### Web-Application