SCHOOL OF BUSINESS

OPIM 5671 – Data Mining and Business Intelligence

Prof - Sudip Bhattacharjee

RESUME CLASSIFICATION

GROUP-5

Team members:

Manas Joshi

Anand Shiv Sharma

Sri Varshini Chava

Anuja Sunil Kamble

Mitul Krishnaswamy Sivakumar

TABLE OF CONTENTS:

1. Summary:

Our Resume Classification project mainly aims to develop a model for categorizing the resumes into any of the labels defined in the dataset. Resume Classification dataset consists of data with 24 subcategories and 2484 rows and 4 columns with no missing values. The project involved collecting the data and preprocessing the data to remove inconsistencies in the data and building the best model. We created different classification models (Logistic regression, Decision Tree Classifier, Support Vector Machine, Naïve bayes, and Gradient Boosting) and compared their evaluation metrics(accuracy, precision, recall, and F1 score) and came to a conclusion that Gradient boosting model is the best model that can be used for resume classification.. From the above developed model, Recruiters can have a standardized recruitment process by screening a large number of resumes based on education, skills, certification etc. and also this model makes it easier for the recruiters to match the candidates with the relevant job opportunities.

2. Problem Statement

Hiring is a crucial part of a company's functioning. Getting the right people often helps the company move in the right direction towards achieving its mission and objectives. Choosing and filtering the right candidates could be difficult due to the high volume of applicants for each listing. Nowadays, there are many candidates applying to different roles in companies. It is becoming difficult for recruiters to divide these resumes into different labels. Our main aim is parsing unstructured resume data into structured format and categorizing those defined resumes into different labels defined in a dataset so that it will be easy for recruiters while going through resumes.

3. Data Description

The dataset (resume.csv) consists of 2484 rows and 4 columns, with category columns divided into 24 subcategories.

The columns are as follows:

ID:

Unique identifier and file name for the respective pdf

Resume_str:

Contains the resume text only in string format.

Resume_html:

Contains the resume data in html format as present while web scraping.

Category:

Category of the job the resume was used to apply.

Present categories used for the resume dataset are (HR, Designer, Information-Technology, Teacher, Advocate, Business-Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital-Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public-Relations, Banking, Arts, Aviation).

4. Exploratory Data Analysis

- The first step in the Exploratory data analysis is loading the dataset. There are 2484 rows and 4 columns with 24 subcategories in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2484 entries, 0 to 2483
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   ID           2484 non-null   int64 
 1   Resume_str   2484 non-null   object
 2   Resume_html  2484 non-null   object
 3   Category     2484 non-null   object
dtypes: int64(1), object(3)
memory usage: 77.8+ KB
```

- Now, rename the independent variable Resume_str as text and the target variable Category as label.
- Drop the columns ID and Resume_html and remove the three categories agriculture, automobile, and BPO as they do not have any contribution to the prediction.
- Now the filtered data consists of 2363 rows with 2 columns and 21 subcategories with zero missing values.

Data Visualization

- The below graph represents the histogram for showing the string lengths in the text column and got to learned that the maximum text length of a string is 250(**figure:1)**.

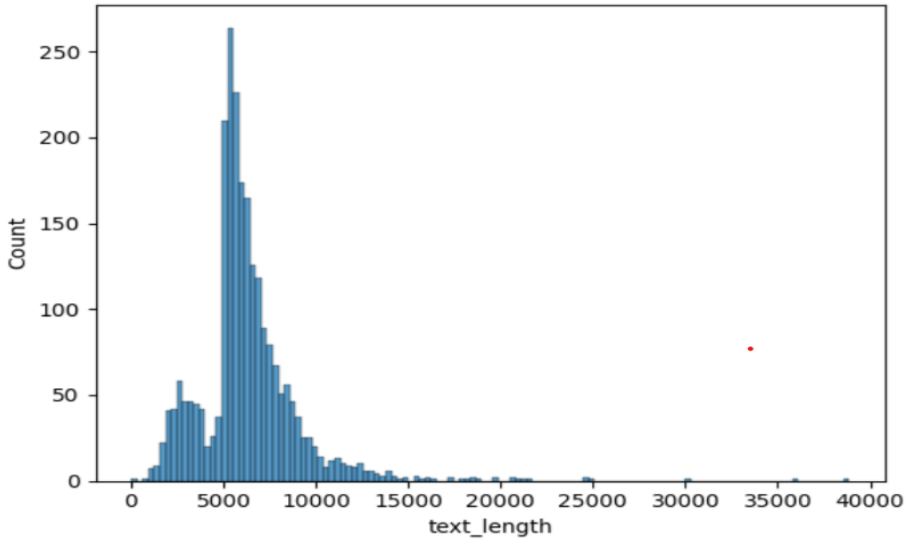

Figure:1

- The bar graph for representing the top 20 frequent words in the document is: (**Figure:2**)

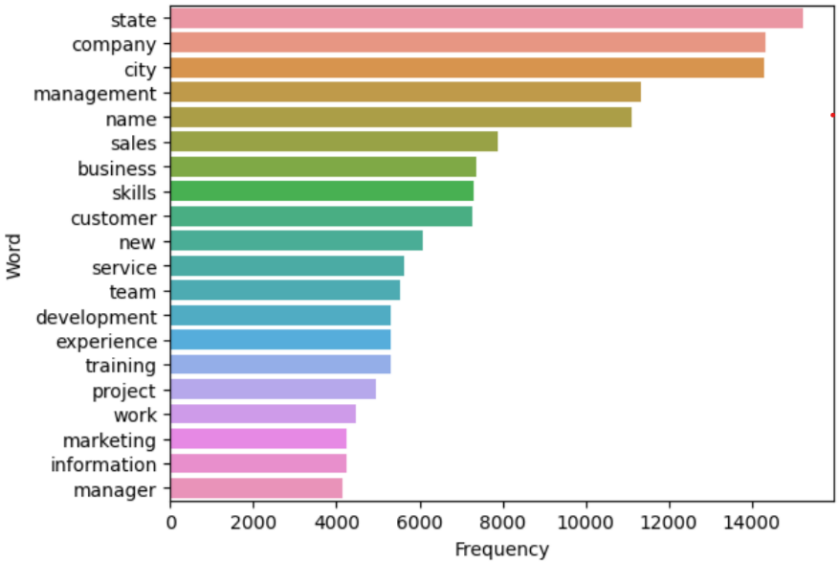

Figure:2

Plotting the twenty most frequent words from the dataset as precursors to performing TDIFvecotriztion. This plot helps us visualize the most frequent words and their count. This plot also helps us finding Term Frequency and Inverse Document Frequency values. From the above graph, it can be seen that state is the most frequently repeated word and next company, and city are the most frequently used words.

- Class imbalance is when instances in one class are considerably different from that of another class. This variation in distribution between the various classes is not desirable in our dataset while implementing models. This is because variation could create bias towards the majority class.

To visualize the class imbalance we plotted the count for each category.

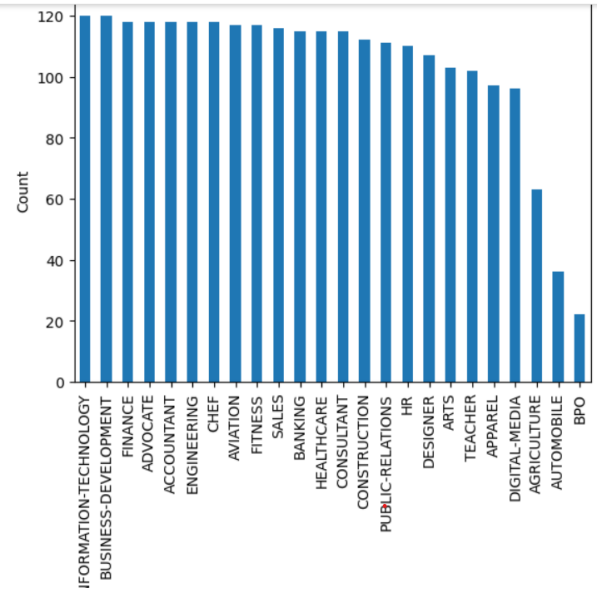

Figure:3

From this plot we can see that our data has imbalance mainly in BPO, Automobile and agriculture. (**Figure:3)**

Bar graph representing all the categories after filtering the data:(**Figure:4)**

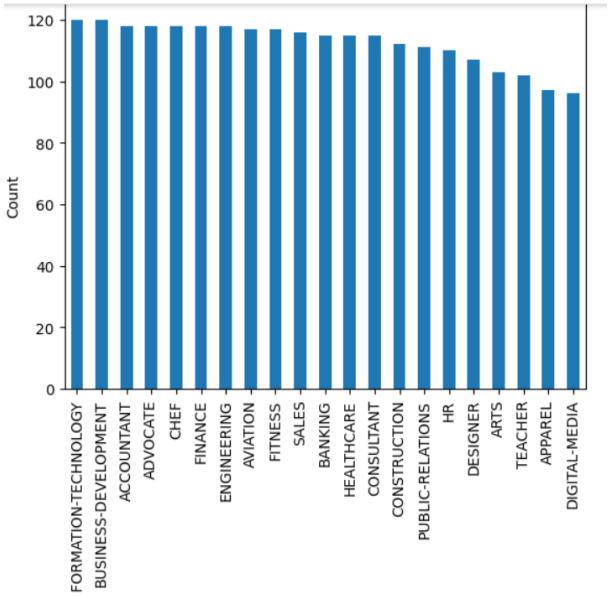

Figure:4

- After preprocessing the data for class imbalance a donut graph was plotted to visualize the distribution of each category. From this chart we can see that all the categories have almost equal distribution. (**Figure:5)**

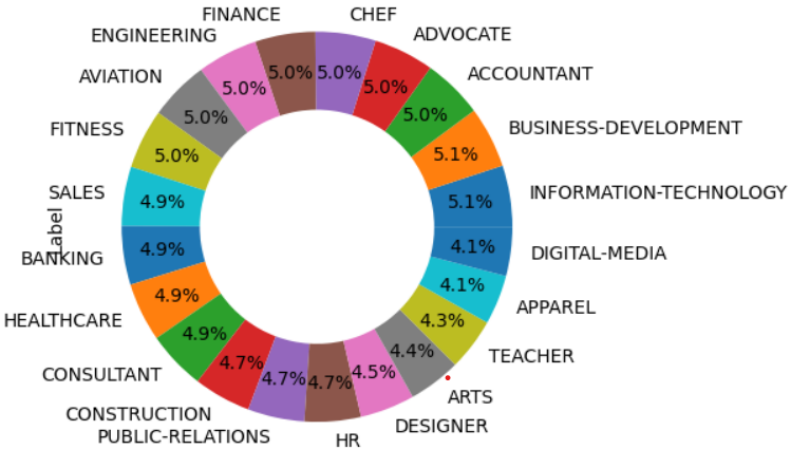

Figure:5

- Word cloud before preprocessing the data is:

  Word clouds provide a visual representation of the frequency and prominence of words within a dataset, making it easy to quickly identify common themes and issues. The size of each word in a word cloud is proportional to its frequency or importance within the dataset, which allows for quick and easy identification of the most frequently mentioned words. Word clouds are a useful tool for summarizing and visualizing text data. The word cloud of our dataset shows us that company, name, city, and state are some of the prominent words. (**Figure:6)**

Figure:6

5. Preprocessing the data:

- Remove Punctuation: The first step in preprocessing your resume dataset is to remove all punctuation marks from the text. This includes commas, periods, semicolons, and other special characters. Punctuation marks do not add any meaning to the text and can interfere with the accuracy of text classification algorithms. By removing punctuation, you can create a cleaner and more consistent dataset for analysis.
- Eliminate Digits: The next step is to eliminate all digits from the resume text. Digits are numeric characters and do not provide any semantic meaning to the text. Removing digits ensures that the text is focused solely on language processing and eliminates the noise caused by numeric data.
- Convert Text to Lowercase: The third step is to convert all text to lowercase. This is important because text classification algorithms treat uppercase and lowercase letters as separate entities, even if they represent the same word. Converting all text to lowercase ensures that the text is consistent and makes it easier to identify and match similar words.
- Token Creation, Lemmatization, and Proper Noun Removal: The final step is to perform token creation, lemmatization, and proper noun removal. Token creation involves splitting the text into individual words, known as tokens. Lemmatization involves converting words to their base form to ensure consistency and reduce the number of unique words in the dataset.

6. Modeling

In our resume classification project, we split our data into 80% for training and 20% for testing. We used TfidfVectorizer with minimum document frequency of 4 and maximum document frequency of 0.7 to convert text into numerical features. This method assigns higher weights to words that are important for a given document and lower weights to words that are less important. We experimented with six different machine learning models for our task: Support Vector Classifier, Naive Bayes, Random Forest Classifier, Logistic Regression, Gradient Boosting Classifier, and Decision Tree Classifier. Here is a brief description of each model:

- Support Vector Classifier (SVC): This is a binary classification algorithm that separates data points with a hyperplane. It has a high accuracy and works well on high-dimensional data, making it a popular choice for text classification tasks.
- Naive Bayes: This is a probabilistic algorithm that assumes that features are independent of each other given the class label. It is simple and efficient, but may not perform well when features are correlated.
- Random Forest Classifier: This is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is robust and can handle noisy data, making it a good choice for text classification tasks.

- Logistic Regression: This is a linear model that estimates the probability of a binary outcome. It is simple and interpretable, but may not perform well when the relationship between features and outcome is non-linear.
- Gradient Boosting Classifier: This is an ensemble learning algorithm that combines multiple weak learners to make predictions. It is flexible and can handle non-linear relationships between features and outcome, making it a popular choice for text classification tasks.
- Decision Tree Classifier: This is a tree-based algorithm that makes predictions based on a set of decision rules. It is simple and interpretable, but may not perform well on high-dimensional data.

Support Vector Classifier

The Support Vector Classifier (SVC) was trained on a dataset of resumes to classify them into different categories. The model achieved a high training accuracy of 0.9978, indicating that it was able to fit the training data very well. However, the testing accuracy of 0.6998 suggests that the model may not generalize well to new resumes.

The precision score of 0.7249 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 72.49% of

the time. The recall score of 0.6999 suggests that the model correctly identified 69.99% of the resumes belonging to the positive category.

```
Training Accuracy of Support Vector Classifier:  0.9978835978835979
Testing Accuracy of Support Vector Classifier:  0.6997885835095138
Precision score for testing: 0.7248637492041559
Recall score for testing: 0.6998595814070403
F1 score: 0.6923474904568159
```

Overall, while the SVC performed well in training, it may require further analysis and optimization to improve its performance in classifying new resumes.(**Figure:7)**

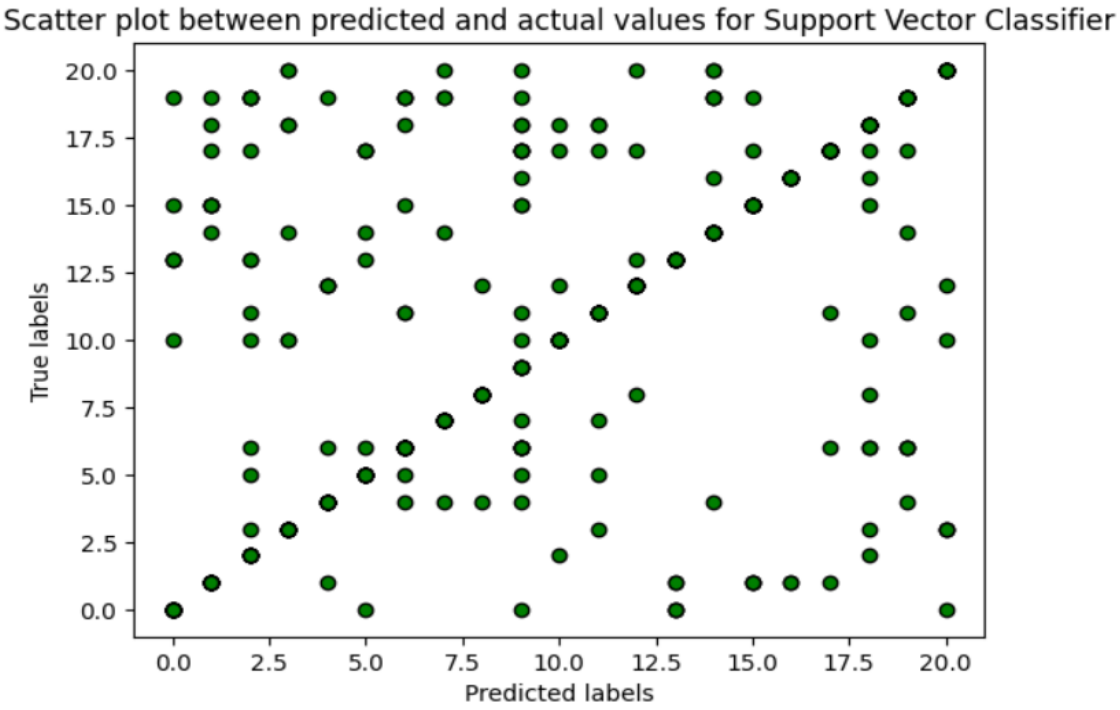

Figure:7

Naive Bayes

The Naive Bayes model was trained on a dataset of resumes to classify them into different categories. The model achieved a training accuracy of 0.7365, indicating that it was able to fit the training data moderately well. However, the testing accuracy of 0.5645 suggests that the model may not generalize well to new resumes.(**Figure:8)**

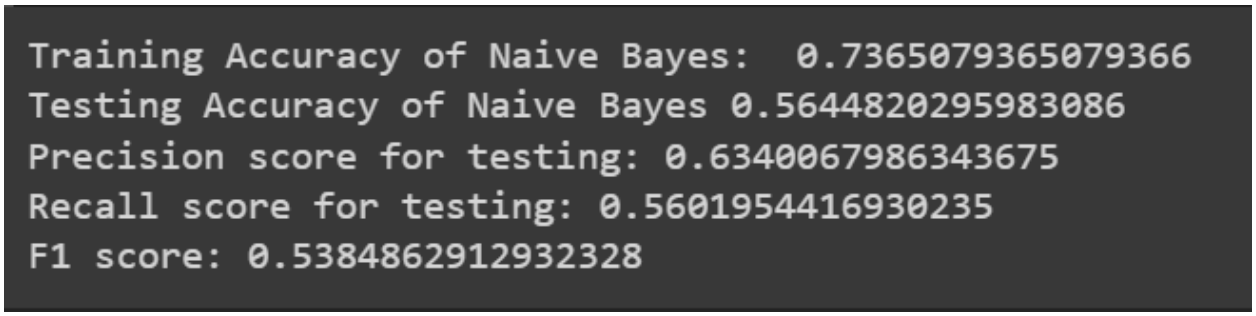

Figure:8

The precision score of 0.6340 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 63.40% of the time. The recall score of 0.5602 suggests that the model correctly identified 56.02% of the resumes belonging to the positive category.

The F1 score of 0.5385 is the harmonic mean of precision and recall, providing an overall measure of the model's performance. Overall, the Naive Bayes model

performed moderately well in training but may require further analysis and optimization to improve its performance in classifying new resumes.(**Figure:9)**

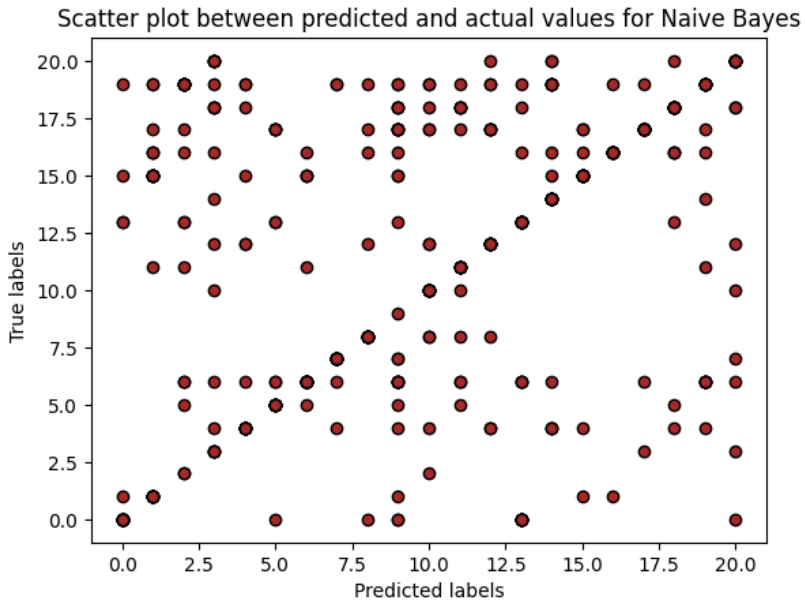

Figure:9

Random Forest Classifier

The Random Forest Classifier model was trained on a dataset of resumes to classify them into different categories. The model achieved a high training accuracy of 0.9995, indicating that it was able to fit the training data very well. However, the testing accuracy of 0.6638 suggests that the model may not generalize well to new resumes.(**Figure:10)**

```
Training Accuracy of Random Forest:  0.9994708994708995
Testing Accuracy of Random Forest:  0.6638477801268499
Precision score for testing: 0.6877846960178601
Recall score for testing: 0.661302533264848
F1 score: 0.6517486131240673
```

Figure:10

The precision score of 0.6878 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 68.78% of the time. The recall score of 0.6613 suggests that the model correctly identified 66.13% of the resumes belonging to the positive category.

The F1 score of 0.6517 is the harmonic mean of precision and recall, providing an overall measure of the model's performance. Overall, while the Random Forest Classifier model achieved a high training accuracy, it may require further analysis and optimization to improve its performance in classifying new resumes.

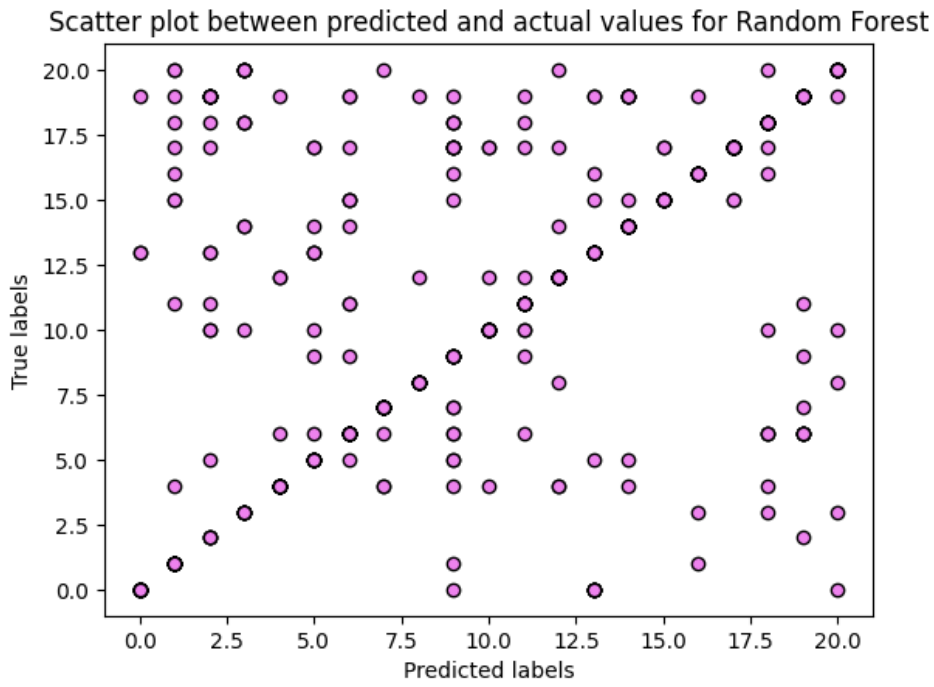

Figure:11

Logistic Regression

The Logistic Regression model was trained on a dataset of resumes to classify them into different categories. The model achieved a training accuracy of 0.8444, indicating that it was able to fit the training data reasonably well. However, the testing accuracy of 0.6490 suggests that the model may not generalize well to new resumes.(**Figure:12)**

```
Training Accuracy of Logistic Regression :  0.8444444444444444
Testing Accuracy of Logistic Regression :  0.6490486257928119
Precision score for testing: 0.6690634505459216
Recall score for testing: 0.6466014365802198
F1 score: 0.637845675602779
```

Figure:12

The precision score of 0.6691 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 66.91% of the time. The recall score of 0.6466 suggests that the model correctly identified 64.66% of the resumes belonging to the positive category.

The F1 score of 0.6378 is the harmonic mean of precision and recall, providing an overall measure of the model's performance. Overall, while the Logistic Regression model performed reasonably well in training, it may require further analysis and optimization to improve its performance in classifying new resumes.

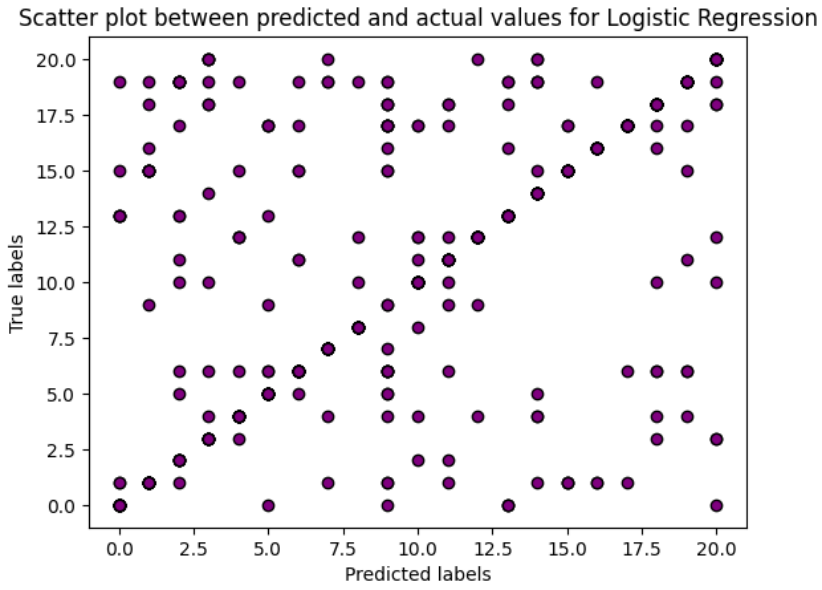

Figure:13

Gradient Boosting Classifier (Best Model)

The Gradient Boosting Classifier model was trained on a dataset of resumes to classify them into different categories. The model achieved a high training accuracy of 0.9995, indicating that it was able to fit the training data very well. The testing accuracy of 0.7167 suggests that the model may generalize well to new resumes.(**Figure:14)**

```
Training Accuracy of Gradient Boosting :  0.9994708994708995
Testing Accuracy of Gradient Boosting :  0.7167019027484144
Precision score for testing: 0.737468293352169
Recall score for testing: 0.7097100323738691
F1 score: 0.7155864056850219
```

Figure:14

The precision score of 0.7375 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 73.75% of the time. The recall score of 0.7097 suggests that the model correctly identified 70.97% of the resumes belonging to the positive category.

The F1 score of 0.7156 is the harmonic mean of precision and recall, providing an overall measure of the model's performance. Overall, the Gradient Boosting Classifier model performed well both in training and testing, indicating that it may be a good choice for classifying new resumes.

Decision Tree Classifier

The Decision Tree Classifier model was trained on a dataset of resumes to classify them into different categories. The model achieved a training accuracy of 0.8132, indicating that it was able to fit the training data reasonably well. However, the testing accuracy of 0.6300 suggests that the model may not generalize well to new resumes.(**Figure:15)**

```
Training Accuracy of Decision Tree :  0.8132275132275132
Testing Accuracy of Decision Tree :  0.6300211416490487
Precision score for testing: 0.6410050701052005
Recall score for testing: 0.6306555093305777
F1 score: 0.6310350916487251
```

Figure:15

The precision score of 0.6410 indicates that when the model predicted a positive result (i.e., a resume belonging to a particular category), it was correct 64.10% of the time. The recall score of 0.6307 suggests that the model correctly identified 63.07% of the resumes belonging to the positive category.

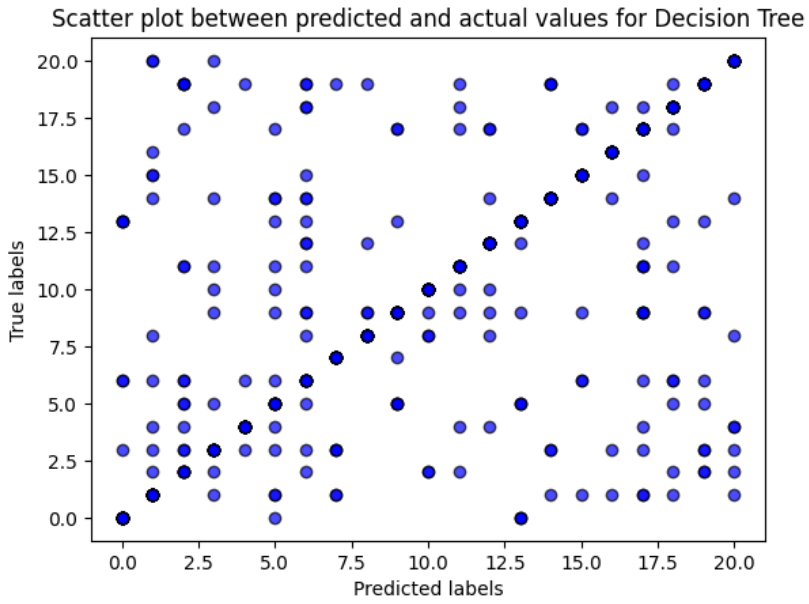

Figure:16

The F1 score of 0.6310 is the harmonic mean of precision and recall, providing an overall measure of the model's performance. Overall, the Decision Tree

Classifier model performed moderately well in both training and testing, but may require further analysis and optimization to improve its performance in classifying new resumes.

7. Model Comparison and Business Insights:

Model	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1 Score
Logistic Regression	0.8444	0.649	0.669	0.6466	0.6378
Decision Tree	0.8132	0.63	0.641	0.63065	0.631
Support Vector Classifier	0.9978	0.6997	0.7248	0.6998	0.6923
Naïve Bayes	0.7365	0.5644	0.634	0.5601	0.5348
Random Forest Classifier	0.9994	0.6338	0.6877	0.6613	0.6517
Gradient Boosting Classifier	0.9994	0.7167	0.7374	0.70971	0.7155

Figure:17

Gradient Boosting Classifier had the highest F1 score(**Figure:17)**

and performed well in both training and testing. The Logistic Regression model had a lower performance level, but may still be a viable option. The Random Forest Classifier and Naive Bayes model had lower F1 scores and may not perform as well. The Decision Tree Classifier also had a lower F1 score. Overall, the Gradient Boosting Classifier may be the best choice for classifying new resumes, but the Support Vector Classifier is also a viable option.

Business Impact

Our resume classifier project has successfully provided business insights that can lead to improved recruitment efficiency, better hiring decisions, a standardized recruitment process, and data-driven recruitment.

Improved Recruitment Efficiency: The GBC model can handle large volumes of resumes and make predictions quickly, which can improve the recruitment efficiency by automating the resume categorization process.

Better Hiring Decisions: The GBC model uses an ensemble of decision trees to make predictions, which can improve the accuracy of the predicted labels. This can help the recruitment team to identify the most suitable candidates for each job, leading to better hiring decisions.

Standardized Recruitment Process: The GBC model uses predefined parameters for categorizing resumes, which can help to standardize the recruitment process. This can lead to more consistent results and improved recruitment outcomes.

Data-driven Recruitment: The GBC model can analyze the predicted labels for each resume, providing insights into the types of candidates that are applying for each job. This can help the recruitment team to identify trends and make data-driven decisions to improve the recruitment process.

Continuous Improvement: The GBC model can be trained on new data and updated over time, leading to continuous improvement in the accuracy of the predicted labels. This can help the model to learn and adapt to the changing needs of the organization, leading to improved recruitment outcomes over time.

8. References:

- https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset
- https://pandas.pydata.org
- https://scikit-learn.org/stable/
- https://www.nltk.org/
- https://matplotlib.org/
- https://numpy.org/
- https://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize.regexp
- https://docs.python.org/3/library/string.html
- https://docs.python.org/3/library/re.html