Best Intern Selection Using Machine Learning

1. Problem statement

Develop an algorithm to filter resumes from the provided dataset and identify the best candidates for the internship position. Utilize the trial version of an AI tool to assist in the resume filtering process and showcase a demo for the first phase of the screening task.

The algorithm should consider relevant attributes from the resumes to evaluate the candidates and make informed hiring decisions. The goal is to identify the most qualified and suitable interns based on their skills, qualifications, experience, and any other relevant factors.

2. Dataset

The provided dataset consists of **1136** rows and **11** columns, containing comprehensive information about the candidates. Each row represents a candidate, while the columns capture various aspects of their skills, grades, and degree. The dataset is structured as follows:

2.1 Skills-related columns:

- Python: Represents the candidate's proficiency level in Python (integer values).
- ML: Represents the candidate's proficiency level in Machine Learning (integer values).
- NLP: Represents the candidate's proficiency level in Natural Language Processing (integer values).
- DL: Represents the candidate's proficiency level in Deep Learning (integer values).
- Other_skills: Contains additional skills possessed by the candidate (textual data).

2.2 Grade-related columns:

- Graduation_year: Indicates the year of the candidate's graduation (integer values).
- Post_grade: Represents the candidate's grade in their post-graduation studies (textual data).
- UG_grade: Indicates the candidate's grade in their undergraduate studies (textual data).
- 12_grade: Represents the candidate's grade in their 12th-grade examination (textual data).
- 10_grade: Indicates the candidate's grade in their 10th-grade examination (textual data).

2.3 Degree-related columns:

- Availability: Indicates the candidate's availability for a 3-month internship (categorical data: 'yes' or 'no').
- Degree: Represents the candidate's highest degree (textual data).

2.4 years and stream columns:

- Availability: Indicate the bachelor or post-graduation year of the candidate.
- stream: Indicate the stream like computer science or IT

It can be used for further analysis and decision-making in the intern selection process.

2.4 Dataset Analysis:

In the given dataset, the columns "name", "post_grade", "ug_grade", "12_grade", and "10_grade" have a significant number of missing values. Specifically, these columns have more than **60%** missing values:

- Name: All 1136 entries are missing, which means this column has no available data.
- Post_grade: **952** missing values, indicating that only a small portion of the data is present for this column.
- UG_grade: **654** missing values, indicating a substantial amount of missing data.
- 12_grade: **643** missing values, suggesting a large number of missing entries.
- 10_grade: **709** missing values, indicating a significant portion of missing data.

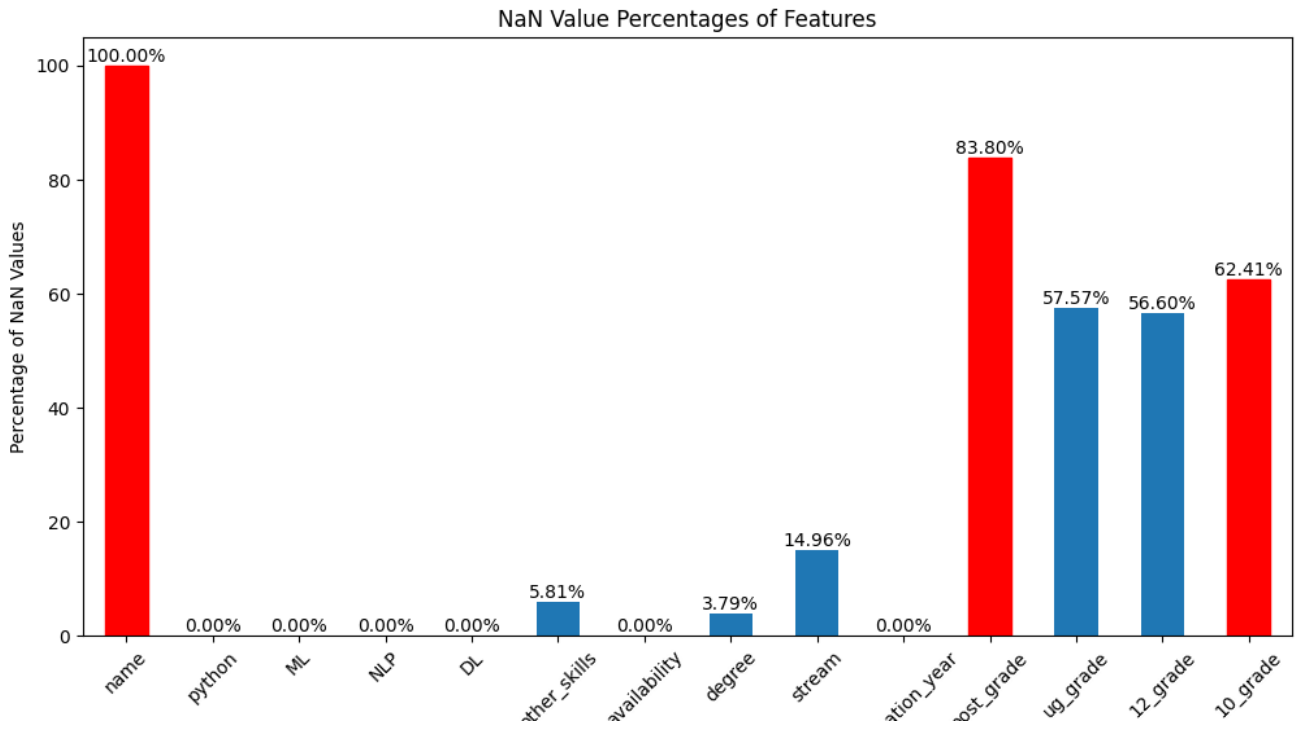

Fig: Missing values(%) in columns

Data Assumption: When columns have a high percentage of missing values, it can affect the overall analysis and interpretation of the data. It is important to consider appropriate strategies for handling missing data, such as imputation or exclusion of rows/columns based on the analysis objectives

3. Approaches

Since the dataset does not contain a target variable, it presents an unsupervised learning problem.

1. In this case, **we can utilize clustering and topic modeling** methods to explore patterns and extract insights from the data.

2. However, considering the context of **resume filtering, which involves classification, a more realistic approach would be to label the data after understanding the underlying patterns.** This would enable us to apply supervised learning models for classification purposes.

Therefore, the proposed approach involves the following steps:

Data Exploration and Preprocessing: Analyze the dataset to understand its structure, identify missing values, and perform necessary preprocessing steps such as handling missing data and data normalization.

Clustering: Apply clustering algorithms such as K-means, DBSCAN, or hierarchical clustering to identify potential clusters or groups within the dataset. This can help in identifying patterns and similarities among candidates.

Topic Modeling: Utilize techniques like Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) to extract meaningful topics from the textual data, such as the "other_skills" column. This can provide insights into the skills and expertise areas present in the candidate pool.

Manual Labeling: Based on the extracted patterns and insights, manually assign labels or categories to the candidates. This can involve creating criteria or thresholds for differentiating between potential candidates.

Supervised Classification: Employ supervised learning algorithms such as decision trees, random forests, or support vector machines to train classification models using the labeled data. This will enable us to classify and identify the best candidates based on the defined criteria.
By combining unsupervised methods (clustering and topic modeling) with supervised classification, we can effectively filter and classify the candidates to identify the best-fit candidates for the internship opportunity.

4. Data-Preprocessing

In order to further analyze and modify certain columns, such as grades, other skills, and degree levels, we have established criteria and thresholds to guide our mapping process. We have identified specific vocabulary and categories within these columns:

1. Other Skills: This column contains a unique set of skills that serve as secondary features. Since our data already includes primary skills in other columns, we consider these secondary skills to be less important. Therefore, we map them to binary values of 0 or 1.

2. Bachelor's Degrees: This column contains a vocabulary of different bachelor's degree types. We utilize this vocabulary to categorize and represent the **degrees with stream** appropriately.

3. Master's Degrees: Similarly, we have a vocabulary for master's degrees, allowing us to map and classify these **degrees with stream** accordingly.

By employing these criteria and mapping techniques, we are able to enhance the analysis and representation of these specific columns, aligning them with the overall focus and importance of the data. we have also dropped some columns like **names** and the **year.** Because they have no use .

4.1 Handling missing Values(missing value indicator method):

- After creating the missing value indicator columns, we can further modify them based on certain criteria related to the original values.
- For instance, consider the "grad_10" column that indicates whether the "grade_10" column has missing values.
- By applying specific criteria, we can determine when to update the missing value indicator.
- For example, if a certain value or range of values in the "grade_10" column is deemed valid, we can check if the original value meets that criteria. If it does, we update the corresponding missing value indicator column to indicate that the value is no longer missing.

- This approach considers the data characteristics and incorporates

domain-specific knowledge, allowing us to handle missing values while retaining valuable information during the unsupervised learning phase

```
# Assuming your DataFrame is named 'df' and the grade column is named 'grade'
#here we are changing tha nan value with zero and making an missing value indicator
#variable which can choose have the cr(criteria)which shows if there is a value which
#greater than cr marks as 1 other wise zero
cr=8.0
df['ug_grade_missing'] = grade_mapping(df, 'ug_grade', cr)
df['post_grade_missing'] = grade_mapping(df, 'post_grade', cr)
df['12_grade_missing'] = grade_mapping(df, '12_grade', cr)
df['10_grade_missing']= grade_mapping(df, '10_grade', cr)
```

4.2 Otherskils columns:

- In our approach, we prioritize the important skills by identifying the columns in the dataset that contain the main skills**(python, data science, NLP, DL)** or relevant information about the candidates.
- These columns are considered more important and influential in our analysis. On the other hand, we treat the "other_skills" column as supplementary information rather than the primary focus.
- We tokenize the "other_skills" column and create a binary indicator variable to represent the presence or absence of skills.
- However, we assign a lower weight or consider the indicator variable as an additional feature, **giving less importance compared to the columns that contain the main skills.** By adopting this approach, we ensure that our analysis focuses on the most critical aspects of the candidates' profiles while still considering additional skills when they are present**.**

4.3 Degree columns:

- Our approach involves assigning different weights based on the degree level of candidates.
- Candidates with post-degree qualifications receive higher weights compared to those with only Bachelor's degrees.
- This prioritizes individuals who have pursued advanced education and have specialized knowledge in their field.
- By incorporating degree-level weighting, we emphasize the importance of higher education and select candidates who have demonstrated a strong commitment to their area of study.

```
    'Post Graduate Diploma in Big Data Analytics (PG-DBDA)'
    ]

# Step 1: Create Degree Level feature
df['Degree Level'] = 0

# Step 2: Preprocessing - Convert to lowercase and remove leading/trailing whitespace
df['degree'] = df['degree'].str.lower().str.strip()


for index, row in df.iterrows():
    degree = row['degree']
    if isinstance(degree,str):
      if any(keyword.lower().replace(' ', '') in degree for keyword in graduation_keywords):
          df.at[index, 'Degree Level'] = 1
      elif any(keyword.lower().replace(' ', '') in degree for keyword in masters_keywords):
          df.at[index, 'Degree Level'] = 2
```

4.5 Categorical Columns and Grades Columns:

- In our approach, we convert categorical values to numerical representations for analysis. For instance, "yes" is mapped to 1, and "no" is mapped to 0. This enables us to perform calculations and use algorithms that require numerical inputs.

- Similarly, for grades, we modify the missing indicator columns based on a threshold. Grades above 85% are mapped to 1, indicating the presence of grade information, while grades below or equal to 85% are mapped to 0.

- This allows us to capture the availability of grade data based on a specific criterion. These mappings and modifications facilitate effective analysis and decision-making in our internship selection process.

```
def grade_mapping(data, col, criteria):
    modified_col = data[col].apply(lambda x: 1 if (not pd.isna(x)
                                   and float(x.split('/')[0]) > criteria) else 0)
    return modified_col
```

5. Feature Selection

Feature selection is essential in unsupervised machine learning to identify relevant features. Two common techniques are Isolation Forest (ISF) and Variance using Principal Component Analysis (PCA).

1. ISF: Detects anomalies by isolating them from the dataset, assigning anomaly scores. It helps exclude less informative features that contribute less to the overall pattern or show minimal variations.

2. Variance using PCA: Reduces dimensionality by transforming features into orthogonal principal components. The components capture maximum variance. Features with low variance are disregarded, focusing on informative features.

- **Feature Variance :**

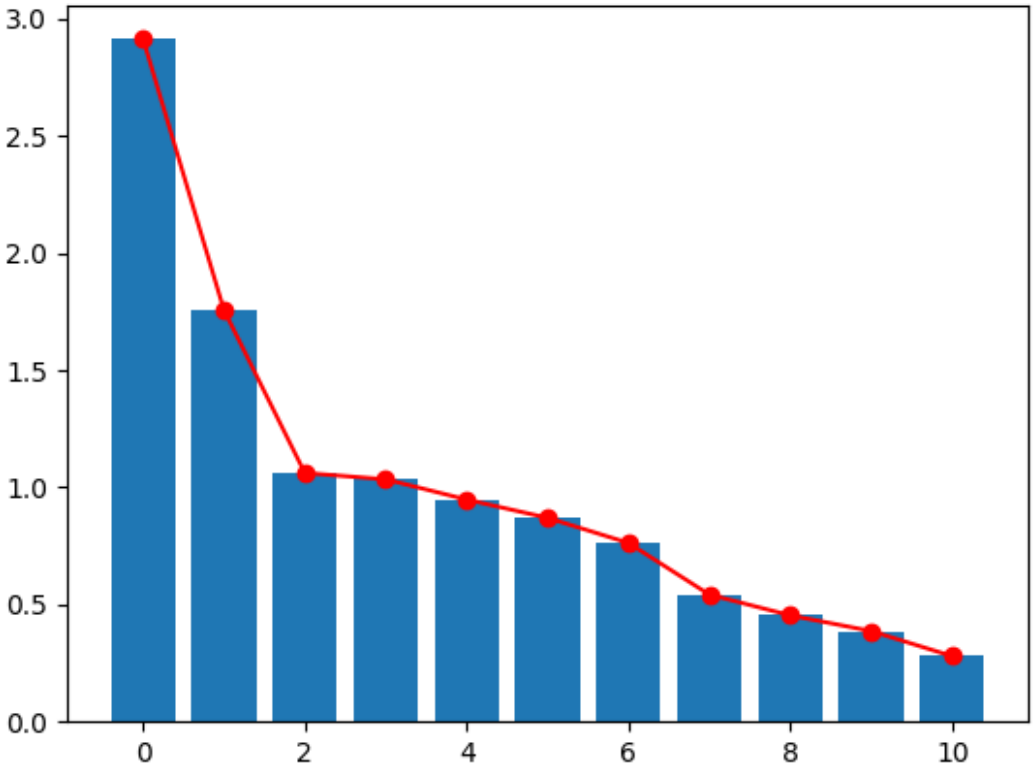

Fig: Features Variace using PCA

- By analyzing the variance, we can identify the principal components that contribute the most to the dataset's variability.
- **Components with higher variance carry more valuable information and capture important features and patterns in the data.**
- Therefore, by maximizing the variance, we can select the most informative principal components for dimensionality reduction and data representation.
- The variance provides a quantitative measure of the importance of each component and helps guide the decision-making process in PCA.

Outliers Detection with Isolation Forest:

- IFA is an unsupervised learning algorithm, meaning it does not require labeled data to detect outliers. It learns the underlying structure of the data solely based on its distribution and does not rely on pre-defined class labels.

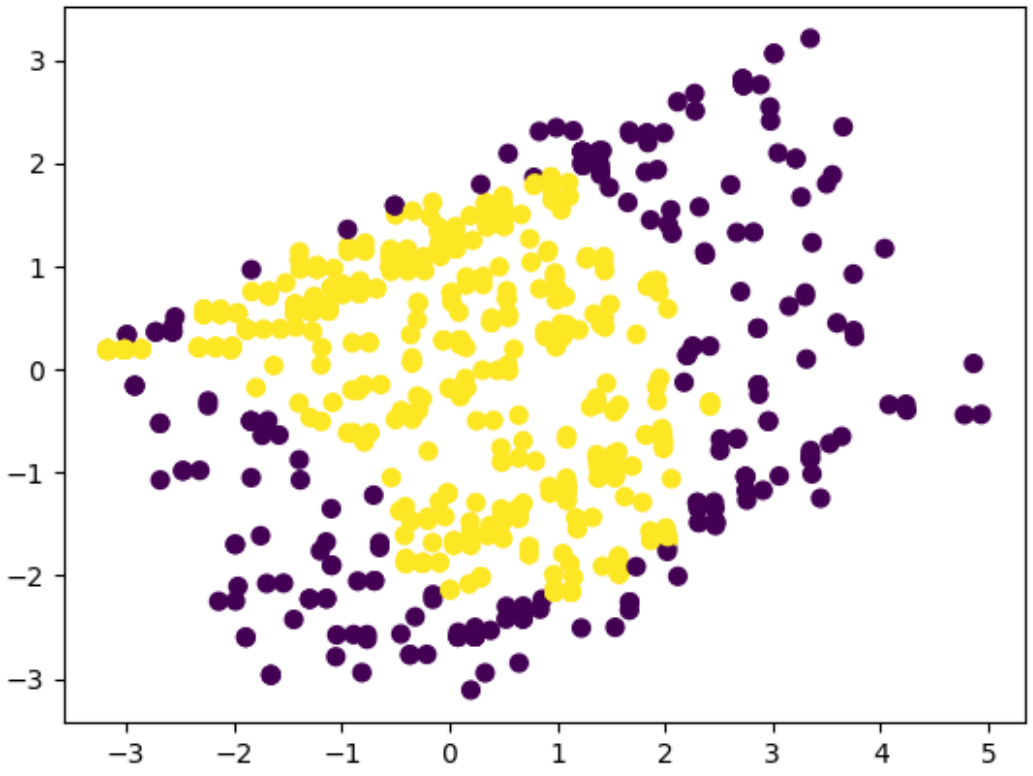

Fig: Outliers(purple) in the dataset

- By utilizing the Isolation Forest algorithm, we can effectively identify outliers or anomalies in the PCA-transformed data. **This enables us to gain insights into potentially unusual or anomalous patterns in the dataset, which can be valuable for further analysis or decision-making processes.**

5.2 Best Feature Selection Approach:

Algorithm 3: Exhaustive search for optimal model-features combination.

$PFA_k \leftarrow$ Principal Feature Analysis class with k features selected
iForest $\leftarrow$ Isolation Forest class
PCA $\leftarrow$ Principal Component Analysis class
df $\leftarrow$ DataFrame Object
n $\leftarrow$ Total number of features
Result: Scores
for $K \in \{4 \dots n\}$ **do**
  #fit the PFA Class:
  PFA_k(df)
  #Get the Transformed Matrix:
  x = PFA.features
  #Apply Isoloation Forest and clean data from outliers:
  x = iForest(x)
  # Apply PCA on kept features:
  x = PCA(x)
  # Fit each model on transformed data:
  $\hat{y}_{BIRCH}$ = BIRCH(x)
  $\hat{y}_{agg\_clustering}$ = AgglomerativeClusterint(x)
  $\hat{y}_{KMeans}$ = KMeans(x)
  $\hat{y}_{Mini-Batch-KMeans}$ = Mini-Batch-KMeans(x)
  #Compute the silhouette score coefficient for each model
  Scores $\leftarrow$
   $(S(\hat{y}_{BIRCH}, x), S(\hat{y}_{agg_clustering}, S(\hat{y}_{KMeans}, x), S(\hat{y}_{Mini-Batch-KMeans}, x))$
end

Explanation:

- we use Principal Feature Analysis (PFA) along with outlier detection and clustering techniques to select the most informative features for the internship selection process.

- PFA helps us identify a subset of features based on principal components. We then detect outliers using the Isolation Forest algorithm and reduce the dimensionality of the data using PCA.
- Next, we apply three clustering algorithms to the reduced feature space and evaluate the quality of the clusters using the silhouette score.
- Finally, we iterate over different numbers of desired features, store the silhouette scores, and track the selected feature names.
- This approach helps us identify relevant features and understand the underlying patterns in the dataset.

So we got the best features:

```
# What is the best
best_score = -1
best_model = ""
best_key = -1

for k, scores in feature_scores.items():
    max_score = max(scores)
    if max_score > best_score:
        best_score = max_score
        best_model = models[scores.index(max_score)]
        best_key = k

#print("Best model:", best_model)
print("Best score:", best_score)
print("Best key (number of features):", best_key)
print('Best features are : ',features_names[best_key])
```

```
Best score: 1.0
Best key (number of features): 3
Best features are :  Index(['NLP', 'DL', 'ug_grade_missing'], dtype='object'
```

6. Model and Metrics

Models :
K-Means: A centroid-based clustering algorithm that partitions the data into K clusters by minimizing the sum of squared distances between data points and their assigned cluster centroids.

2. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** A density-based clustering algorithm that groups together data points based on their density and separates regions of low density, allowing for the detection of arbitrary-shaped clusters.

3. **Agglomerative Clustering**: A hierarchical clustering algorithm that starts with each data point as a separate cluster and progressively merges clusters based on the similarity between them, forming a hierarchy of clusters.

Metrics:

4. **Silhouette Score**: A metric that measures how close each sample in one cluster is to samples in the neighboring clusters, providing an indication of the clustering quality. Higher values indicate well-defined and distinct clusters.

5. **Calinski-Harabasz Index:** A metric that calculates the ratio between the within-cluster dispersion and the between-cluster dispersion, providing a measure of cluster compactness and separation. Higher values indicate better-defined and well-separated clusters.

6. **Davies-Bouldin Index**: A metric that measures the average similarity between clusters, considering both the within-cluster dispersion and the between-cluster distances. Lower values indicate more cohesive and well-separated clusters.

7. Training

1. **Model Training**: We train three clustering algorithms (K-Means, DBSCAN, Agglomerative Clustering) on the training data (x_train).

2. **Evaluation of Training Set:** We evaluate the performance of each algorithm on the training set by calculating metrics such as the Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. These metrics assess the quality and separation of the clusters formed by each algorithm.

3. **Evaluation of Testing Set:** We assess how well the trained algorithms generalize to unseen data by applying them to the testing set (x_test) and calculating the same evaluation metrics. This allows us to evaluate the algorithms' performance on new and unseen internship candidate data.

4. **Best Model Selection:** We select the algorithm with the highest Silhouette Score as the best-performing model. The Silhouette Score reflects the quality and separation of the clusters. By choosing the algorithm with the highest score, we ensure that the selected model produces well-defined and distinct clusters.

7.2 Two-Phase Training:

We perform model training in two phases. **The first phase** involves training the clustering algorithms using all available features.The first phase aimed to incorporate the remaining candidates whose data contained missing values. As previously mentioned, a missing value indicator column was created to handle these missing values. By considering this additional information, the model sought to provide a comprehensive evaluation of candidates, taking into account the missing data. The trained model classified these candidates into the same four clusters: moderate fit, good fit, bad fit, and best fit.

In the second phase, we select the best features determined from previous analyses and train the algorithms using only these selected features. the second phase focused on candidates whose data contained complete information,

excluding the features with missing values. This model aimed to capture the patterns and characteristics of candidates who had comprehensive data available, thus providing insights into the best-fit candidates without the potential bias introduced by missing values. This approach helps us evaluate the impact of feature selection on the clustering performance and identify the set of features that yield the best results.

By employing a two-phase training approach and evaluating the performance of different clustering algorithms, we can make informed decisions about the most suitable algorithm for clustering internship candidates and identify the most relevant features for the task.

8. Target Labeling Based on Clusters

In this step, we assign target labels to the data based on the clusters generated by the K-Means algorithm. Here's an overview of the process:

1. **Agglomerative Clustering:** We apply the K-Means algorithm with a specified number of clusters (4 in this case) to the scaled training data.

2. **Cluster Labels:** The K-Means algorithm assigns cluster labels to each data point, ranging from 0 to 3.

3. **Cluster Names:** We define meaningful names for each cluster based on their characteristics and suitability for the internship opportunity. In this case, the clusters are labeled as "Best Fit," "Good Fit," "Moderate Fit," and "Reject Fit."

4. **Mapping Labels to Names:** We create a new column in the data frame called "Cluster Name" and map the cluster labels to their corresponding names using a dictionary.

5. Output: The resulting data frame contains the original cluster labels and their corresponding cluster names.

This target labeling process helps identify the suitability of candidates for the internship opportunity based on the clusters generated by the K-Means algorithm.

It provides a clear understanding of which cluster each candidate belongs to and enables further analysis and decision-making during the intern selection process.

9. Training a SuperVised Model

After labeling our data using the clustering method, we can further enhance our analysis by training a supervised model on the labeled dataset. In this regard, we have utilized a RandomForest Classification Model to train on the labeled data and evaluate its performance.

Through this approach, we were able to achieve accurate predictions. Incorporating a supervised model allows us to leverage the labeled information and gain deeper insights into the classification of internship candidates.

```
# Transform the test set using the fitted scaler
X_test_scaled = scaler.transform(X_test)

# Create a Random Forest Classifier
model = RandomForestClassifier()

# Train the model on the scaled training set
model.fit(X_train_scaled, y_train)

# Make predictions on the scaled test set
y_pred = model.predict(X_test_scaled)

# Calculate classification metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Print the classification metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

10. Result and Evaluation

1. Unsupervised Training Models:
   - Two unsupervised models were trained for the resume filtration task: one using all features and the other using the best features selected through feature selection techniques.
   - Clustering algorithms were employed to group candidates into four clusters: best, worst, good, and moderate.
   - The cluster labels provided insights into the suitability and potential of candidates based on their features.

2. Target Labeling and Classification Model:
   - The cluster labels obtained from the unsupervised models were used to label all the data, enabling the creation of target labels for each candidate.
   - A classification model was trained using the labeled data, allowing for the prediction of candidate suitability based on their features.
   - The classification model utilizes the labeled information to accurately categorize candidates into appropriate clusters.

3. The combination of unsupervised training models, target labeling, and classification modeling has resulted in an effective approach for filtering and classifying candidates in the resume filtration process.

4. The models offer a systematic method to identify the most suitable candidates based on their features, ensuring an efficient and accurate selection process for the internship opportunity.

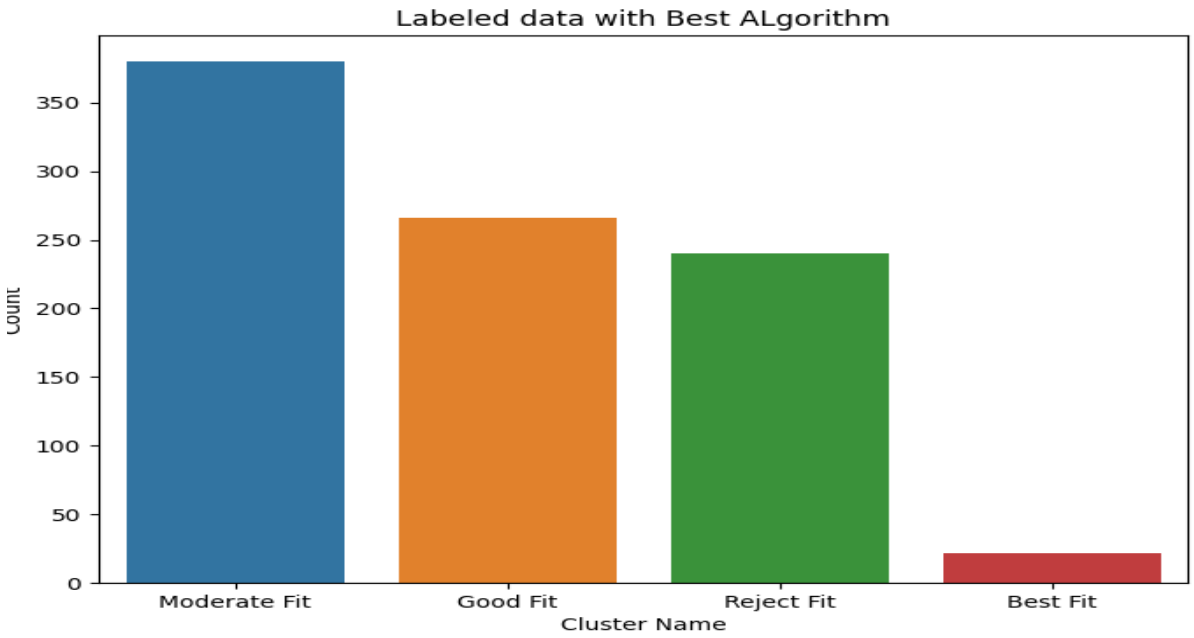

Fig: Clusters data with the best model trained on all features

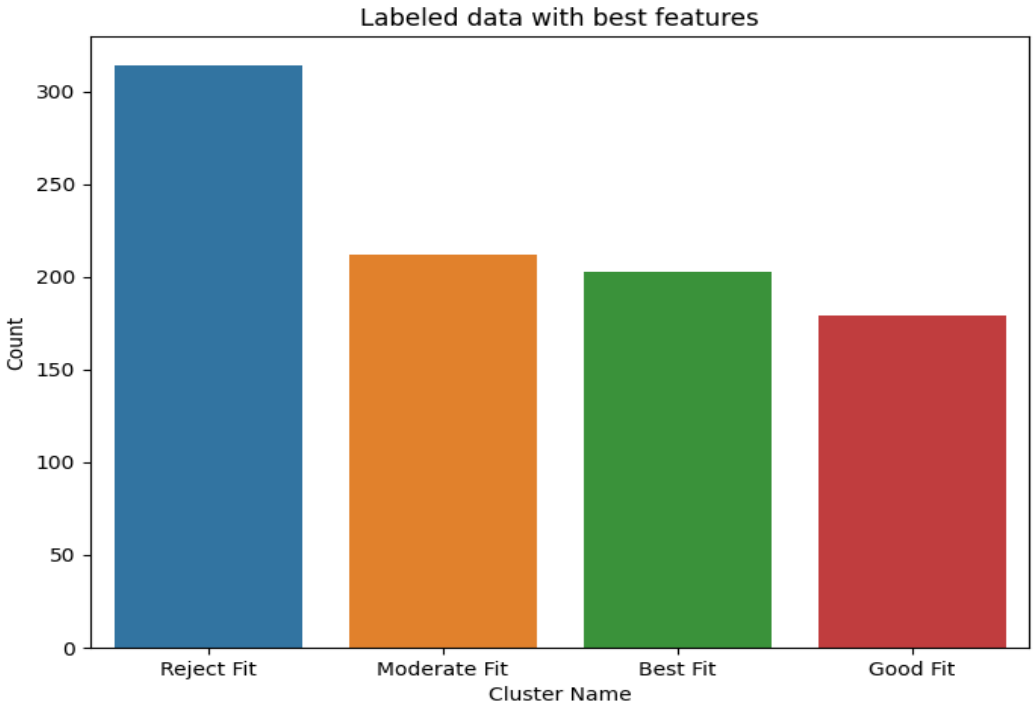

Fig: Clusters data with the best model trained on best features

Evaluations and challenges

We encountered several challenges and limitations that impacted the performance and outcomes of the internship candidate selection process. These challenges include:

1. Handling Missing Values:
    - One of the significant challenges we faced was the presence of a high percentage of missing values in many data columns.
    - Handling missing values, especially when they account for more than 70% of the data, can be complex.
    - The cluster model trained on all features might be biased due to the difficulty in accurately imputing or representing the missing values.
    - On the other hand, the cluster model trained on the best features proved to be more accurate and reliable in overcoming this challenge.
2. Limited Data Availability:
    - The dataset we worked with had limitations in terms of the quantity and quality of the available data.
    - Many columns contained a significant number of missing values, which limited the amount of information and insights that could be extracted.
    - This limitation affected the overall performance of our models and their ability to accurately classify and select the best-fit candidates.
3. Probabilistic Approach:
    - The labeling of data and the training of classification models were implemented using a probabilistic approach based on the cluster information.
    - However, the effectiveness of this approach is highly dependent on the quality and representativeness of the data.

In conclusion, **The cluster model trained on all features may exhibit bias due to the complexity of handling missing values.**

Conversely, **the cluster model trained on the best features demonstrated higher accuracy and reliability.**

These evaluations and challenges highlight the need for further refinement and **improvement in data collection** and preprocessing to enhance the accuracy and effectiveness of the internship candidate selection process.