INTEGRATED JOB POST VERIFICATION AND PERSONALIZED JOB RECOMMENDATION SYSTEM

Team Members

Himanshi Raturi
Sumeet Suvarna
Shubhangi Dabral

Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington

CSCI-B 565 DATA MINING - Prof. Yuzhen Ye

Table of Contents

KEYWORDS:

Job Posting Verification, Personalized Job Recommendations, Data Quality, Job Market, Authentication, Recommendation Algorithms, Skills Matching, Data-Driven Insights, Features, data frame, data preprocessing, Data Mining, Machine Learning

ABSTRACT:

In today's fast-changing job market, it's crucial to have accurate and precise job postings. This project introduces a system that combines two important aspects: checking if job listings are genuine and offering personalized job suggestions. Leveraging advanced data mining techniques and a large dataset with detailed job information, company profiles, required qualifications, and incentives, this system helps job seekers trust the job listings and find the right job for them. With the rise of online job listings, it's challenging to know if they're real. The Job Posting Verification part of our system would carefully check job details, company information, and qualifications to ensure job listings are reliable. It will utilize data mining techniques to extract valuable insights from the data. Our Job Posting Verification system is designed to address the growing concern over the authenticity of online job listings. In this digital age, where fraudulent job offers are prevalent, our system provides a robust solution.

In addition, our Personalized Job Recommendation System will use advanced data mining algorithms to suggest jobs that match a person's skills and preferences. It will look at job titles, locations, specific details, and job descriptions to find the best matches, again making use of data mining for enhanced accuracy.

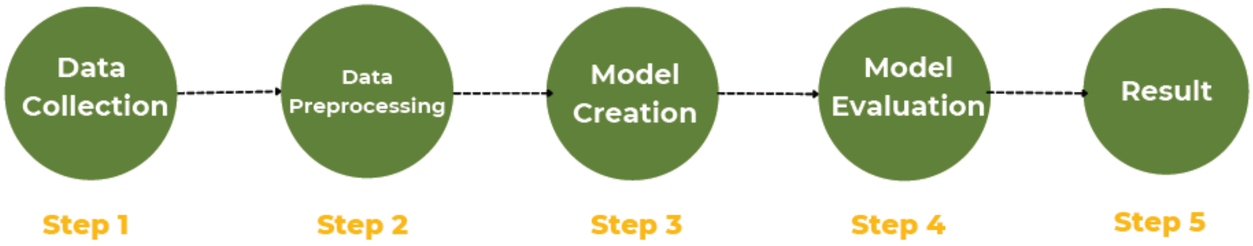
In summary, this project aims to solve the problem of false job information and help job seekers find the right jobs. By combining Job Posting Verification and Personalized Job Recommendations with data mining techniques, we want to create a trustworthy job market ensuring the authenticity of job postings while providing tailored job recommendations to users based on their personalized search query. This will contribute to a strong and dependable job market, enriched by data-driven insights. Our project represents a significant stride towards a more reliable, efficient, and user-centric job market. By harnessing the power of data mining and personalized algorithms, we are not only addressing the immediate challenges of job seekers but also contributing to a more robust and dynamic employment landscape.

INTRODUCTION:

Recognizing the need for a new and creative solution, this project introduces a special approach that combines the benefits of Job Posting Verification and a Personalized Job Recommendation System. The Job Posting Verification part acts as a protector for job seekers, carefully checking if job listings are real and trustworthy. It's essential for confirming important details like salary ranges, making sure company profiles are complete, and ensuring that listed requirements and benefits are relevant. This verification process is crucial for building trust and honesty in the job market, protecting job seekers from false information and misleading postings. This verification mechanism is pivotal in cultivating a job market grounded in transparency and reliability, thereby empowering job seekers with confidence and peace of mind.

Additionally, the Personalized Job Recommendation System focuses on honesty and integrity. It uses advanced data-driven methods to connect job seekers with opportunities that match their search specific skills, preferences, and geographic interests. By intricately mapping individual skills and preferences to job specifications, it ensures a high degree of compatibility and relevance. And it will cater to geographical preferences and interests in specific sectors or industries, thereby enhancing the suitability of job suggestions. By thoroughly analyzing job titles, locations, department details, and detailed job descriptions, this system is excellent at finding job listings that are not only relevant but also genuinely interesting to individual users.

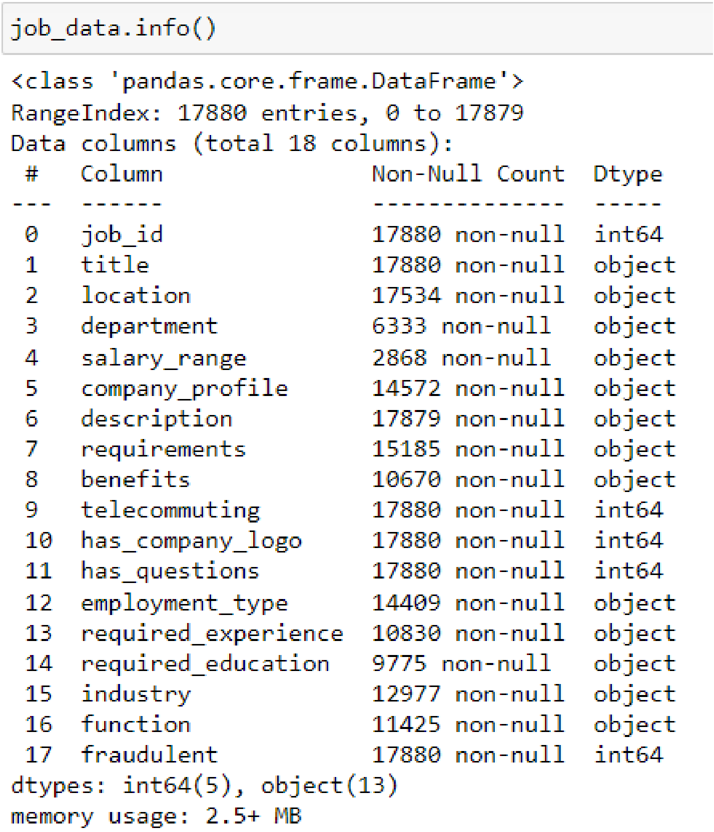
METHODS:



In our project, we employ advanced data mining techniques to tackle job market challenges. We begin with thorough data collection, creating a comprehensive dataset of job postings. Rigorous preprocessing ensures data integrity, handling

text cleaning, missing values, and feature extraction. Our system focuses on Job Posting Verification, scrutinizing company details, salary ranges, and benefits to eliminate fraudulent postings. Simultaneously, our Personalized Job Recommendation System analyzes skills and preferences against job attributes, providing tailored suggestions. Rigorous evaluation would enhance accuracy and user experience, fostering a trustworthy, data-driven job market connection.
The methods that we will follow for Integrated Job Post Verification and Personalized Job Recommendation System project is as follows:

1. **Data Collection**:
   The foundational step of our data mining project involved the meticulous collection of a comprehensive dataset, which was sourced from Kaggle, a well-known platform for data science competitions and collaborative projects.
   This dataset is a rich aggregation of job listings, encapsulating 18,000 job descriptions along with a diverse set of 18 attributes for each listing.

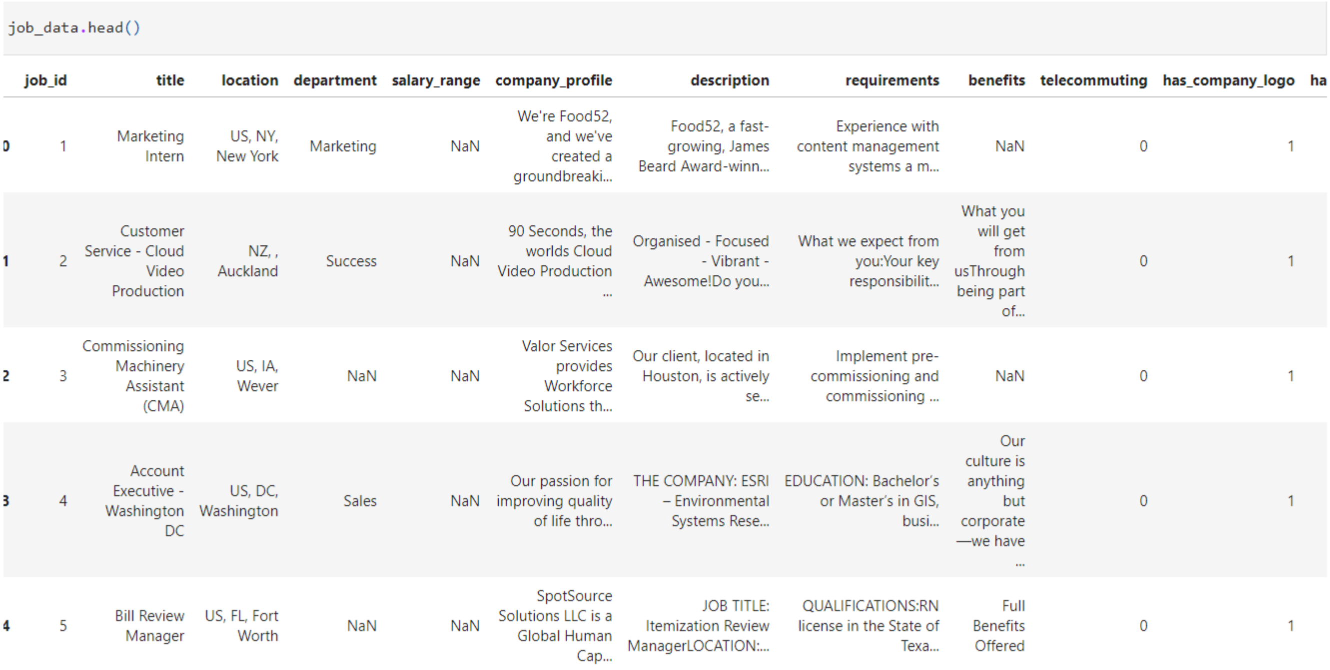


```
job_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   job_id               17880 non-null  int64 
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64 
 10  has_company_logo     17880 non-null  int64 
 11  has_questions        17880 non-null  int64 
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64 
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

The attributes encompass a wide array of both textual and meta-information related to the jobs, such as titles, descriptions, location details, company information, industry specifications, and other essential job-related metrics. The text data provides in-depth insights into the nature and requirements of the job, while the meta-information offers context and additional details that are crucial for both the verification of job postings and the personalized job recommendation processes. Each attribute was carefully chosen to ensure a robust framework for our data mining algorithms to operate upon, allowing for a nuanced analysis of job market trends and the development of a reliable job matching system. The integrity and comprehensiveness of this dataset are fundamental to the accuracy and effectiveness of our subsequent data analysis and model training phases.

Sample Data:



```
job_data.head()
```

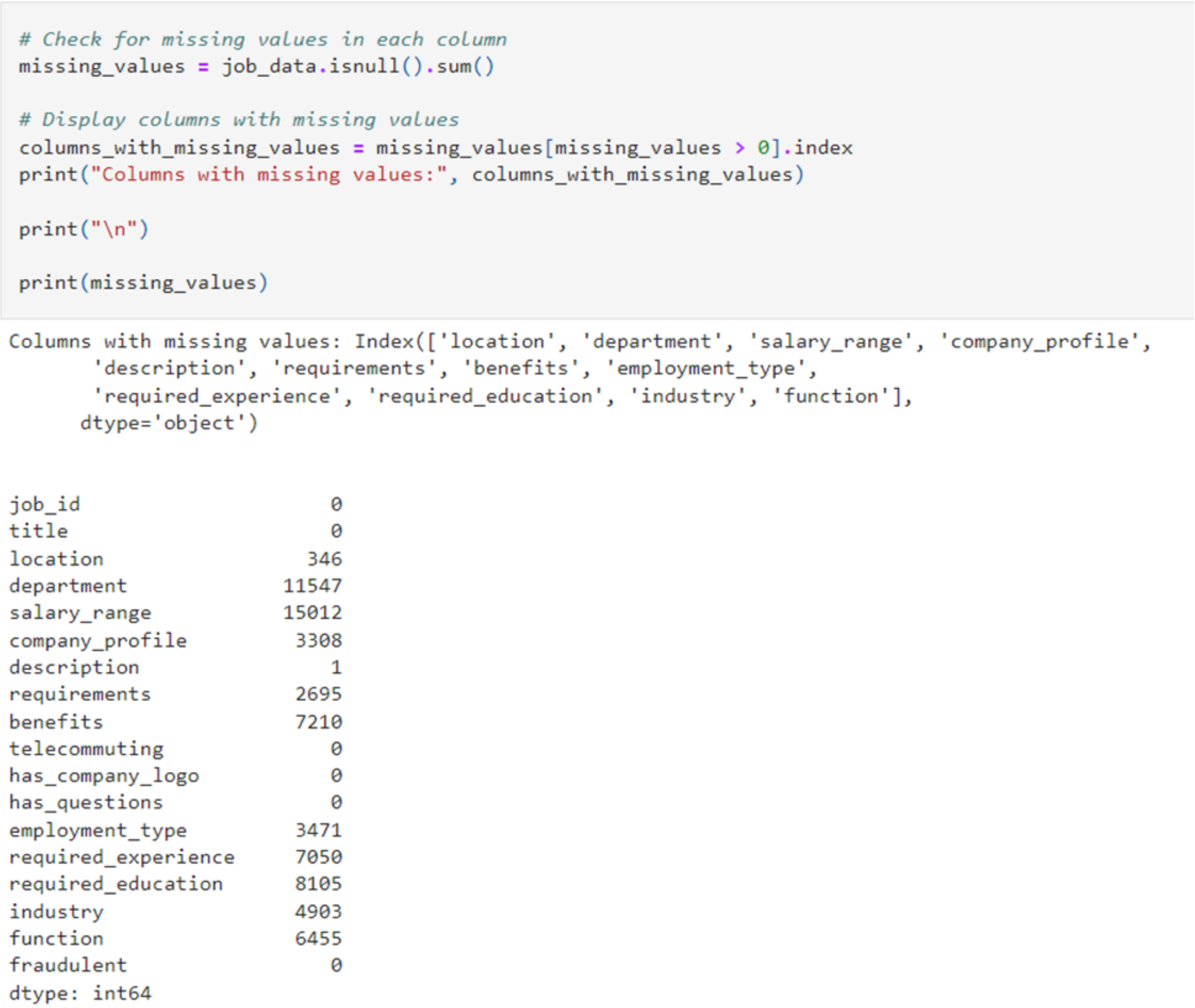
	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_company_logo	ha
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...	NaN	0	1	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	0	1	
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...	NaN	0	1	
3	4	Account Executive - Washington DC	US, DC, Washington	Sales	NaN	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our culture is anything but corporate—we have ...	0	1	
4	5	Bill Review Manager	US, FL, Fort Worth	NaN	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Benefits Offered	0	1	

2. Data Preprocessing And Exporatory Data Analysis:

A critical stage in our data mining project was the preprocessing of our dataset, which involved several steps to ensure the data's quality and readiness for analysis.

2.1 Data Cleaning:

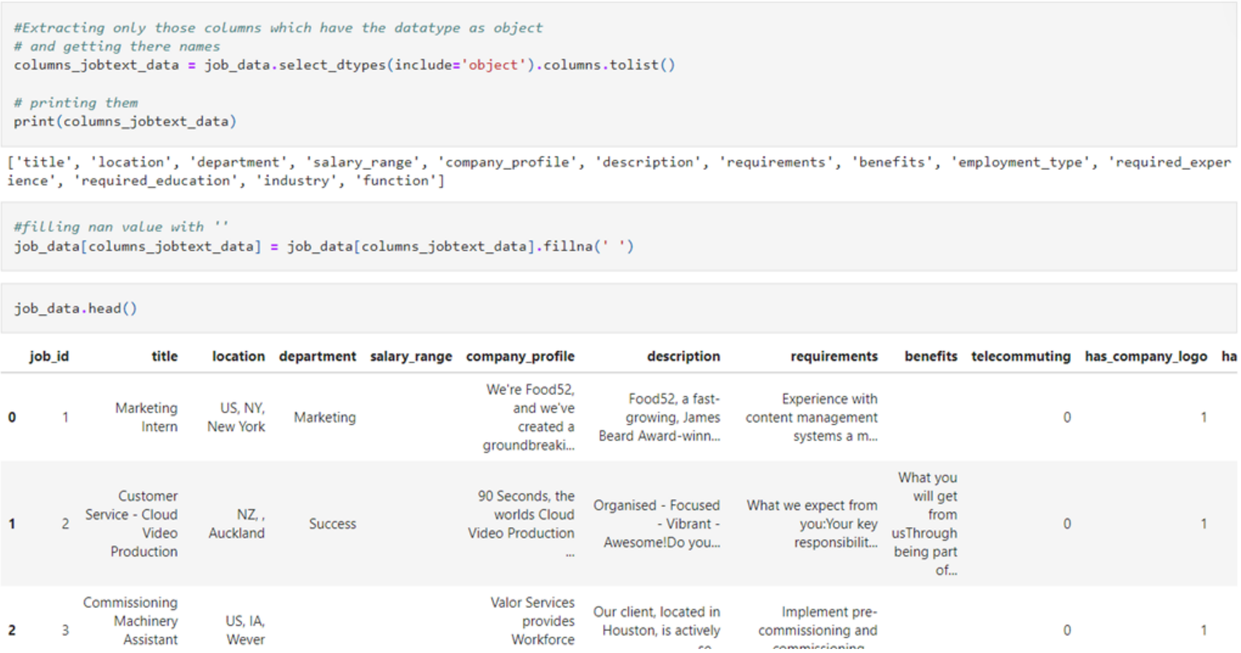
For Data Cleaning, we addressed missing values and corrected any inconsistencies present in the job postings data. From the image below you can see that there were a lot of missing values in ‘location’, ‘company_profile’,’requirements’, etc. columns.



```
# Check for missing values in each column
missing_values = job_data.isnull().sum()

# Display columns with missing values
columns_with_missing_values = missing_values[missing_values > 0].index
print("Columns with missing values:", columns_with_missing_values)

print("\n")

print(missing_values)
```

```
Columns with missing values: Index(['location', 'department', 'salary_range', 'company_profile',
       'description', 'requirements', 'benefits', 'employment_type',
       'required_experience', 'required_education', 'industry', 'function'],
      dtype='object')


job_id                     0
title                      0
location                 346
department             11547
salary_range           15012
company_profile         3308
description                1
requirements            2695
benefits                7210
telecommuting              0
has_company_logo           0
has_questions              0
employment_type         3471
required_experience     7050
required_education      8105
industry                4903
function                6455
fraudulent                 0
dtype: int64
```

These missing values were handled by replacing the NaN values with empty strings for some of the attributes.



```python
#Extracting only those columns which have the datatype as object
# and getting there names
columns_jobtext_data = job_data.select_dtypes(include='object').columns.tolist()

# printing them
print(columns_jobtext_data)
```

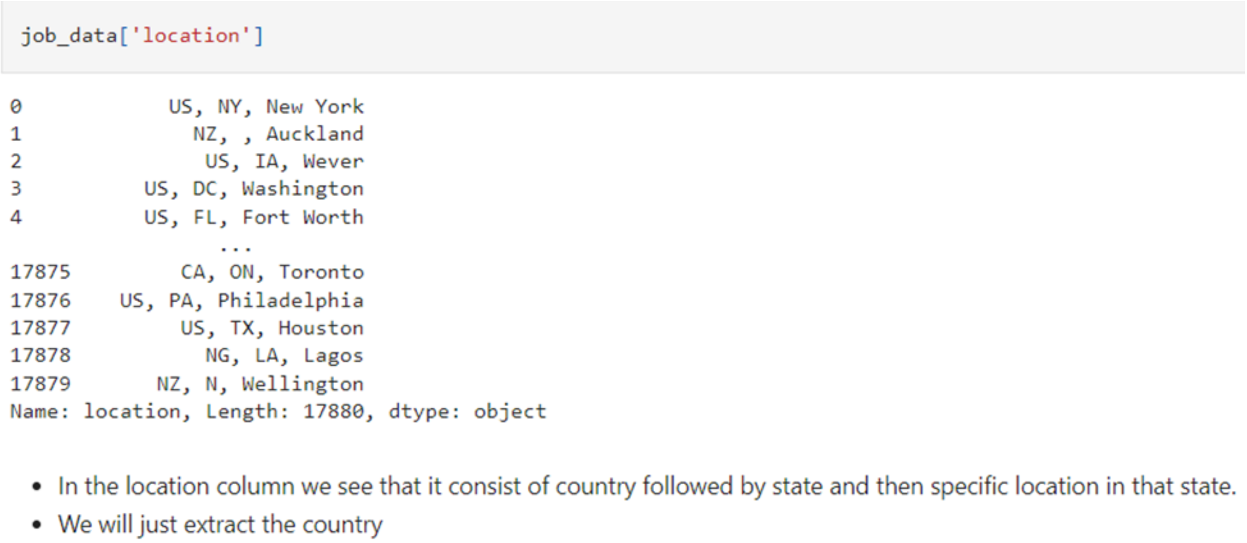
```
['title', 'location', 'department', 'salary_range', 'company_profile', 'description', 'requirements', 'benefits', 'employment_type', 'required_exper
ience', 'required_education', 'industry', 'function']
```

```python
#filling nan value with ''
job_data[columns_jobtext_data] = job_data[columns_jobtext_data].fillna(' ')
```

```python
job_data.head()
```

	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_company_logo	ha
0	1	Marketing Intern	US, NY, New York	Marketing		We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...		0	1	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success		90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	0	1	
2	3	Commissioning Machinery Assistant	US, IA, Wever			Valor Services provides Workforce	Our client, located in Houston, is actively	Implement pre-commissioning and		0	1	

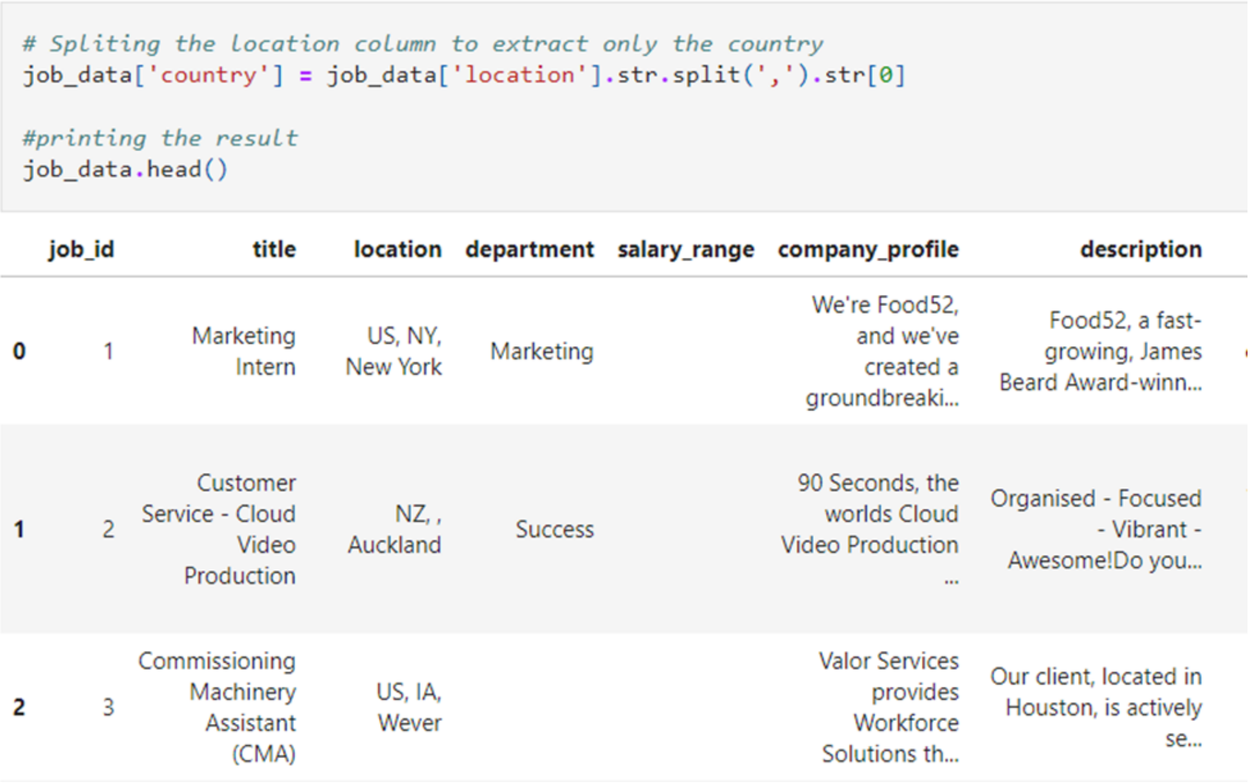
We also performed refinement of location and salary information, which are pivotal in job matching accuracy.



```python
job_data['location']
```

```
0            US, NY, New York
1              NZ, , Auckland
2               US, IA, Wever
3          US, DC, Washington
4          US, FL, Fort Worth
                ...
17875         CA, ON, Toronto
17876    US, PA, Philadelphia
17877         US, TX, Houston
17878           NG, LA, Lagos
17879       NZ, N, Wellington
Name: location, Length: 17880, dtype: object
```

- In the location column we see that it consist of country followed by state and then specific location in that state.
- We will just extract the country



```
# Spliting the location column to extract only the country
job_data['country'] = job_data['location'].str.split(',').str[0]

#printing the result
job_data.head()
```

	job_id	title	location	department	salary_range	company_profile	description
0	1	Marketing Intern	US, NY, New York	Marketing		We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success		90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever			Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...

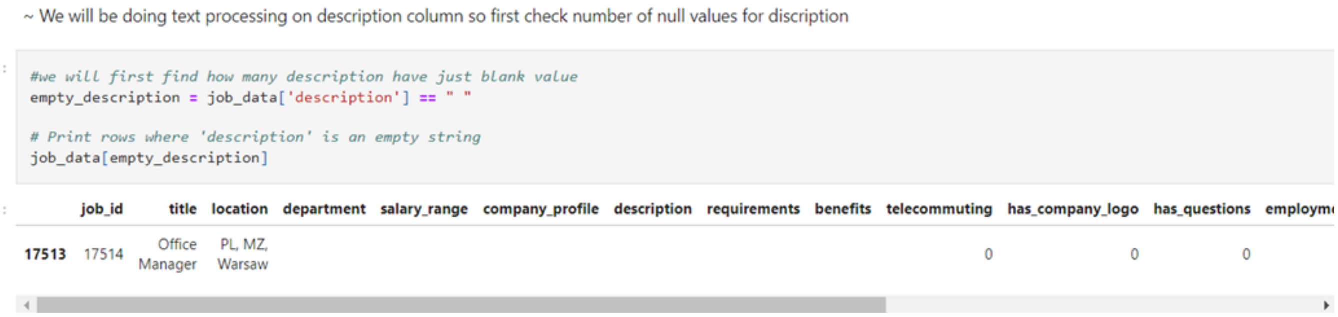
Using the location column, we derived the 'Country' column by only selecting relevant parts from the column and ignoring the abbreviations.

(Insert actual screenshot displaying country column)

For the salary column, the values were in the range format with hypen (-). To handle the missing values in the salary column we calculate the average of the salaries.

(Insert actual screenshot displaying salary column)

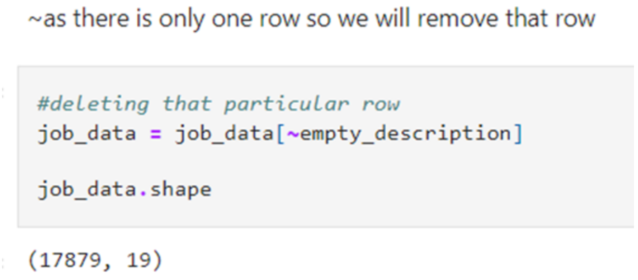
We were thorough in our approach, dropping rows with null values specifically, in the 'description' column to maintain the integrity of our textual analysis.



~ We will be doing text processing on description column so first check number of null values for discription

```
#we will first find how many description have just blank value
empty_description = job_data['description'] == " "

# Print rows where 'description' is an empty string
job_data[empty_description]
```

	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_company_logo	has_questions	employm
17513	17514	Office Manager	PL, MZ, Warsaw							0	0	0	



~as there is only one row so we will remove that row

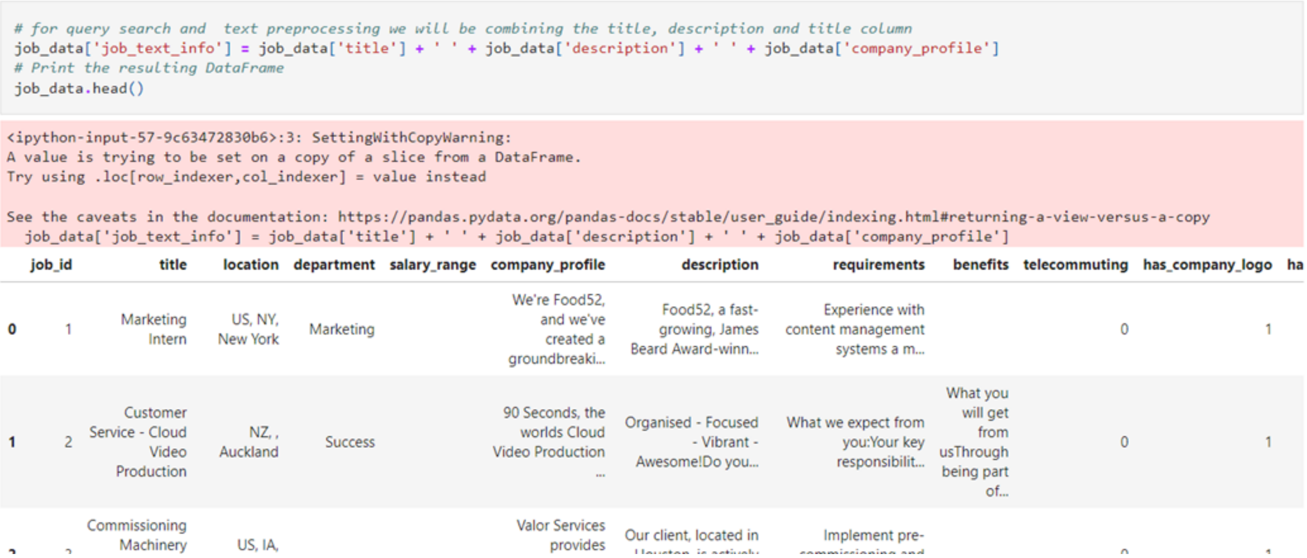
```
#deleting that particular row
job_data = job_data[~empty_description]

job_data.shape
```

```
(17879, 19)
```

2.2 Text Processing:

Moving to Text Processing, we employed a range of techniques to refine our textual data. We initiated tokenization, breaking down text into individual elements for easier handling. This was followed by stemming, where words were reduced to their base or root form. Additionally, we removed stop words, which are commonly used words that contribute little to the interpretive value of the data.

This step was essential to distill the text to its most informative components, paving the way for more effective feature extraction and subsequent analysis.

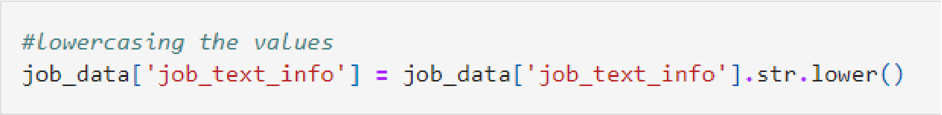


```
# for query search and  text preprocessing we will be combining the title, description and title column
job_data['job_text_info'] = job_data['title'] + ' ' + job_data['description'] + ' ' + job_data['company_profile']
# Print the resulting DataFrame
job_data.head()
```

```
<ipython-input-57-9c63472830b6>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  job_data['job_text_info'] = job_data['title'] + ' ' + job_data['description'] + ' ' + job_data['company_profile']
```

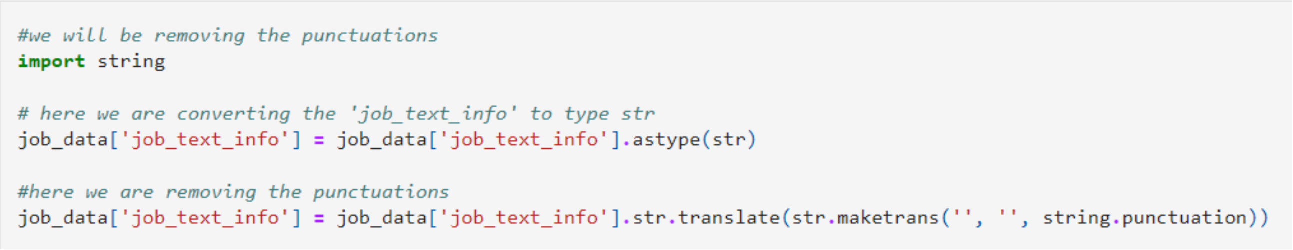
	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_company_logo	ha
0	1	Marketing Intern	US, NY, New York	Marketing		We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...		0	1	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success		90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	0	1	
2	3	Commissioning Machinery	US, IA,			Valor Services provides	Our client, located in	Implement pre-		0	1	

Initially, the process begins by consolidating the textual content from the 'title', 'description', and 'company_profile' columns into a single column named 'job_text_info'. This amalgamation is crucial to create a unified text corpus that captures all relevant information for each job posting.



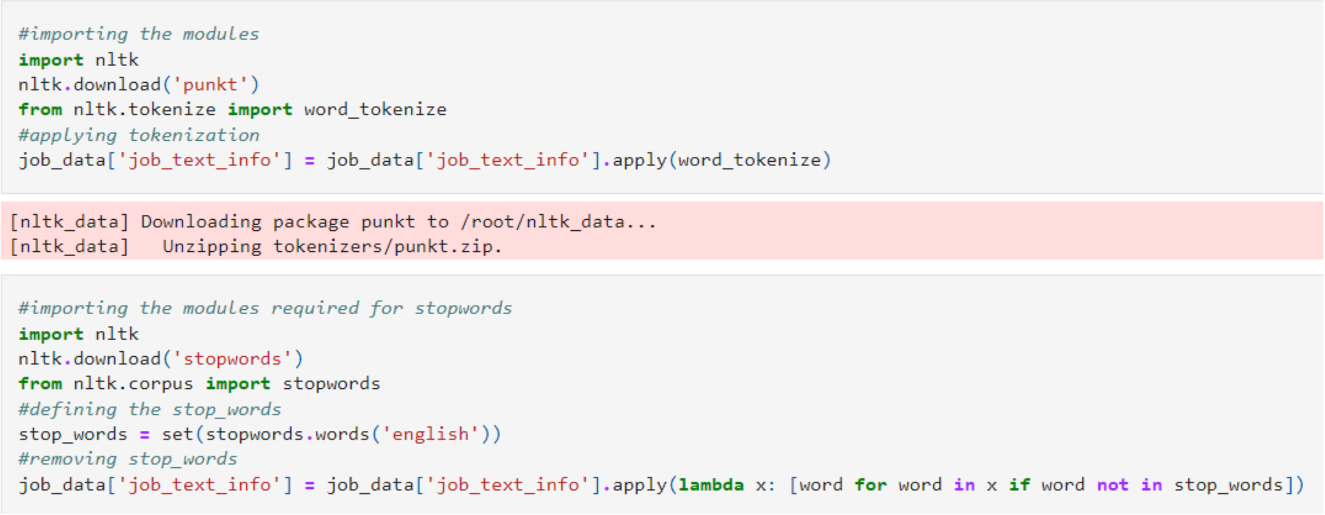
```
#lowercasing the values
job_data['job_text_info'] = job_data['job_text_info'].str.lower()
```

Subsequently, the combined text in 'job_text_info' is converted to lowercase. This step is important for standardizing the text and is a common preprocessing technique to ensure that the same words in different cases are treated as the same token



```python
#we will be removing the punctuations
import string

# here we are converting the 'job_text_info' to type str
job_data['job_text_info'] = job_data['job_text_info'].astype(str)

#here we are removing the punctuations
job_data['job_text_info'] = job_data['job_text_info'].str.translate(str.maketrans('', '', string.punctuation))
```

Next, all punctuation marks are removed from the text. This is done to reduce noise and prevent the algorithm from treating words followed by punctuation differently from the same words not followed by punctuation (e.g., "skills," and "skills")

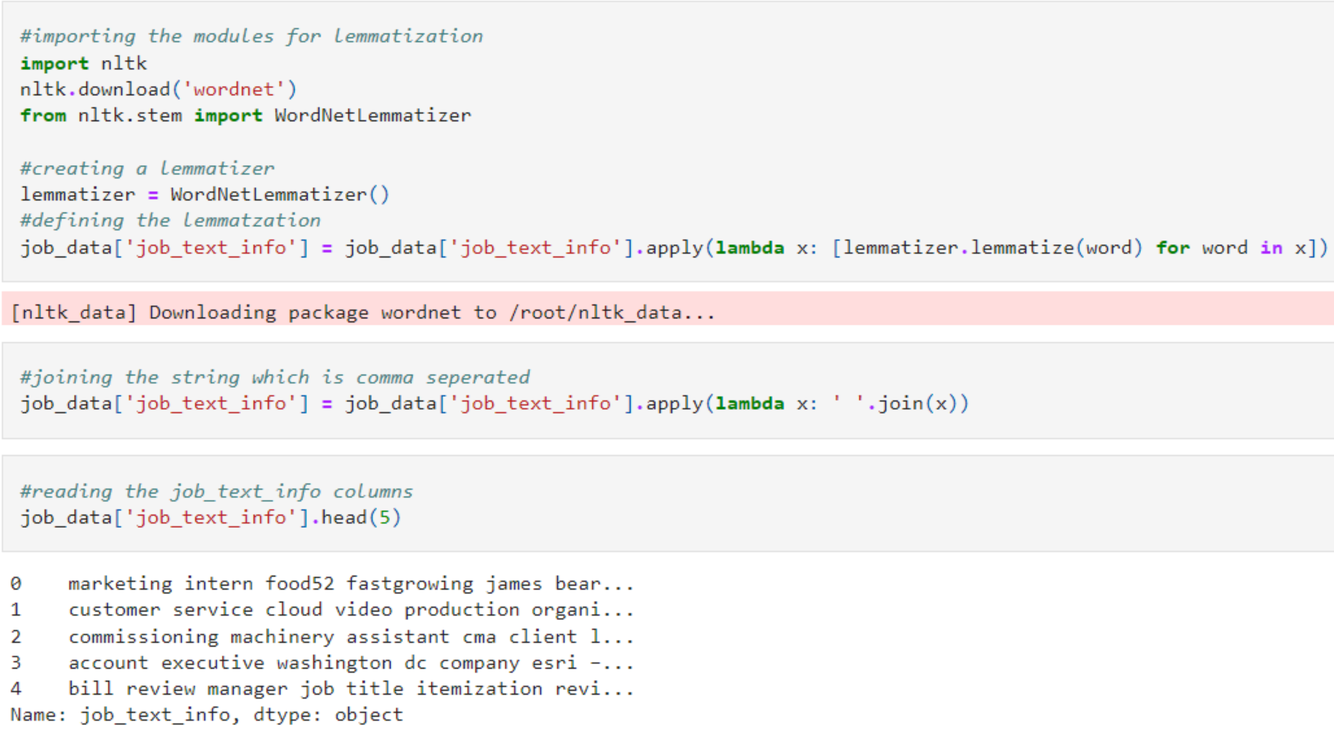


```python
#importing the modules
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
#applying tokenization
job_data['job_text_info'] = job_data['job_text_info'].apply(word_tokenize)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```python
#importing the modules required for stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
#defining the stop_words
stop_words = set(stopwords.words('english'))
#removing stop_words
job_data['job_text_info'] = job_data['job_text_info'].apply(lambda x: [word for word in x if word not in stop_words])
```

The cleaned text is then tokenized, meaning it is split into individual words or tokens. This is a foundational step for most natural language processing tasks as it enables the algorithm to work with the basic units of text. Following tokenization, common stop words (like "the", "is", "in", etc.) are removed from the text. Stop words are typically filtered out because they occur frequently and offer little value in terms of understanding the content of the text.



```
#importing the modules for lemmatization
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

#creating a lemmatizer
lemmatizer = WordNetLemmatizer()
#defining the lemmatzation
job_data['job_text_info'] = job_data['job_text_info'].apply(lambda x: [lemmatizer.lemmatize(word) for word in x])
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
#joining the string which is comma seperated
job_data['job_text_info'] = job_data['job_text_info'].apply(lambda x: ' '.join(x))
```

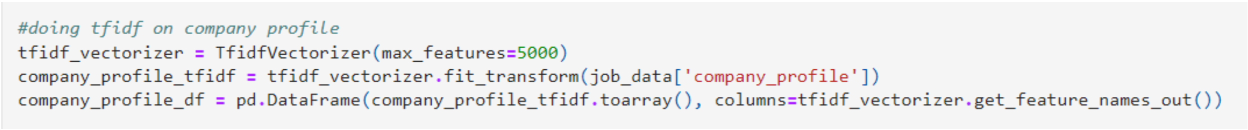
```
#reading the job_text_info columns
job_data['job_text_info'].head(5)
```

```
0    marketing intern food52 fastgrowing james bear...
1    customer service cloud video production organi...
2    commissioning machinery assistant cma client l...
3    account executive washington dc company esri -...
4    bill review manager job title itemization revi...
Name: job_text_info, dtype: object
```

After cleaning and tokenization, lemmatization is applied, which reduces words to their base or dictionary form (lemma). Unlike stemming, lemmatization considers the context and converts the word to its meaningful base form. Finally, the individual tokens are re-joined into a single string. This is necessary after tokenization and lemmatization because most algorithms expect the input as continuous text and not as a list of tokens.

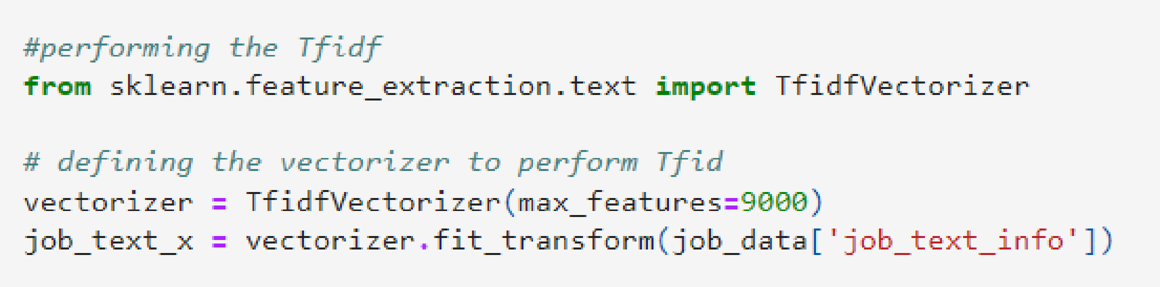
2.3 Feature Extraction:

For Feature Extraction, we carefully extracted relevant features from the job postings. This included crucial information such as job descriptions, company profiles, and specific requirements.



```
#doing tfidf on company profile
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
company_profile_tfidf = tfidf_vectorizer.fit_transform(job_data['company_profile'])
company_profile_df = pd.DataFrame(company_profile_tfidf.toarray(), columns=tfidf_vectorizer.get_feature_names_out())
```

In this step, we applied TF-IDF to the 'company_profile' column. You used the TfidfVectorizer from the sklearn.feature_extraction.text module, setting a cap at 5000 features.



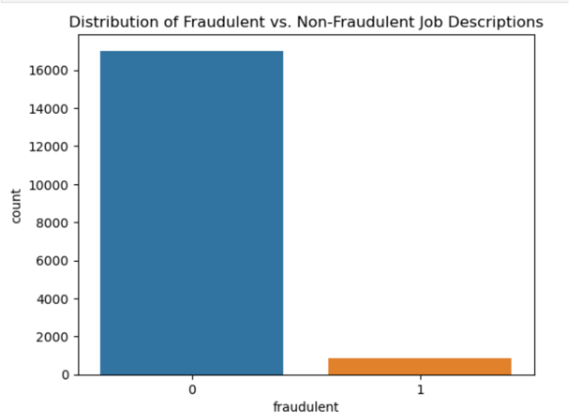
```
#performing the Tfidf
from sklearn.feature_extraction.text import TfidfVectorizer

# defining the vectorizer to perform Tfid
vectorizer = TfidfVectorizer(max_features=9000)
job_text_x = vectorizer.fit_transform(job_data['job_text_info'])
```

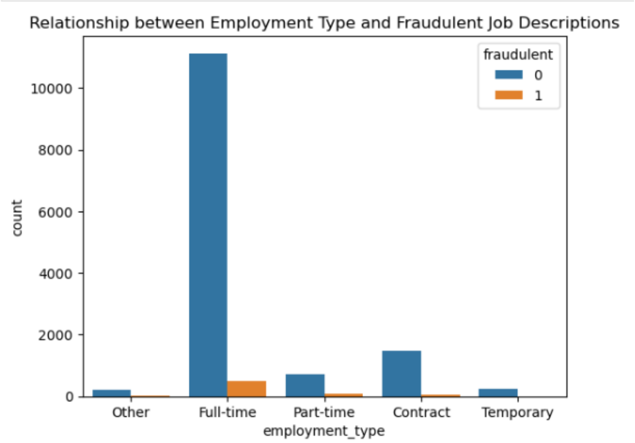
Next, you performed a similar TF-IDF vectorization on the 'job_text_info' column, which contains the combined text of the job title, description, and company profile. This is required for a Neural Network.

The feature extraction phase of your data processing involved the application of Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to the preprocessed text data. This technique is essential for transforming textual information into a structured, numeric format that can be utilized by machine learning algorithms, in this case, for a Logistic Regression approach and Neural Network approach.

<u>Exploration Data Analysis</u> :

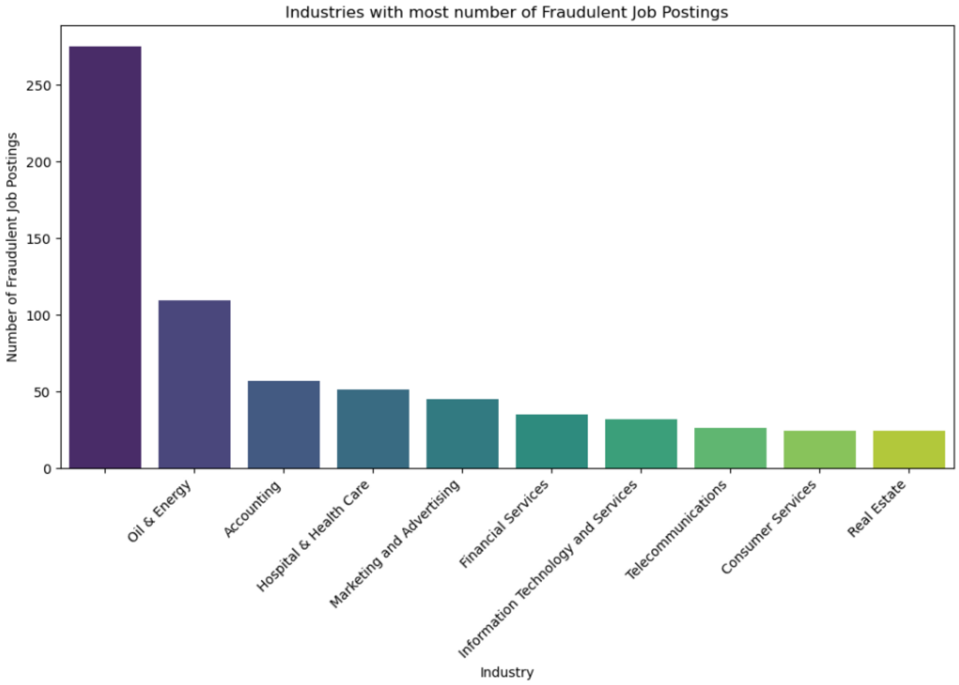
We performed a few visualizations on our dataset to visualize the relationship between different attributes.



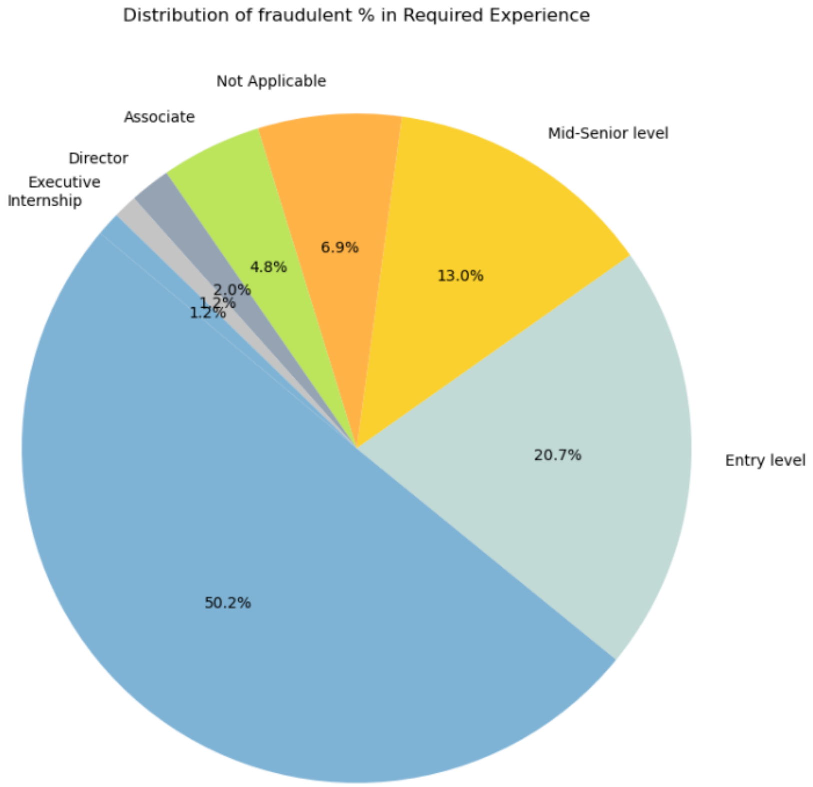


This code employs the seaborn library to generate a count plot illustrating the distribution of the target variable, "fraudulent." This visualization aids in assessing the balance or imbalance between job descriptions classified as fraudulent and those categorized as non-fraudulent. This stark contrast suggests that the dataset contains a significantly higher number of non-fraudulent job postings compared to fraudulent ones.

The employment types are categorized into 'Other', 'Full-time', 'Part-time', 'Contract', and 'Temporary'. From the chart, it is clear that full-time positions have the highest number of job postings overall and dominate both categories of fraudulent and non-fraudulent postings. The visualization serves to illustrate not only the distribution of job types but also the prevalence of fraud within each employment category, with full-time positions being the most common for both legitimate and fraudulent job listings.



The image shows a bar chart titled "Industries with the most number of Fraudulent Job Postings". It illustrates the distribution of fraudulent job postings across various industries. From this visualization, it can be inferred that the Oil & Energy industry is most prone to fraudulent job listings, according to this dataset, while the Real Estate industry has the fewest fraudulent postings. This information can be critical for job seekers who need to be cautious while searching for jobs in these sectors, especially in Oil & Energy, which shows a significantly higher frequency of fraud occurrences.



The image presents a pie chart that illustrates the distribution of fraudulent job postings across different levels of required experience. The largest segment, covering more than half of the chart, represents Entry level positions, indicating that they constitute the majority of fraudulent postings.This suggests that fraudulent job postings are more commonly associated with jobs requiring less experience, particularly those targeting Entry level candidates. This might be due to fraudsters targeting job seekers who may have less experience in identifying scam listings. Positions that require more experience, such as Executive and Director roles, seem to be less affected by fraud, likely due to a combination of factors including fewer applicants and a more rigorous recruitment process.

3. Model Creation:

We will train classification models to identify fraudulent job postings based on verified and trustworthy data points. Also, we will use classification models to understand user intent from search queries. And then classify queries into categories (e.g., job types, industries) for precise recommendations.

We utilized a variety of models, such as K-Nearest Neighbors (KNN), Random Forest Classifier, Logistic Regression, and Neural Network, to attain the best predictive performance.

3.1 K-Nearest Neighbors (KNN):

In our project, we utilized the K-Nearest Neighbors (KNN) algorithm as part of supervised machine learning, employing the KNeighborsClassifier from the scikit-learn library. KNN is a straightforward classification method that relies on the similarity between data points. It classifies a new data point based on the majority class of its k-nearest neighbors in the feature space. The selection of the 'k' value which in our case was taken as 5, representing the number of neighbors, is a crucial parameter influencing the model's performance. To optimize this parameter, grid search and random search techniques were applied, tailoring the algorithm to our dataset. KNN is particularly effective for tasks with non-linear decision boundaries and proves valuable in scenarios featuring local patterns or clusters in the data.

3.2 Random Forest Classifier:

4. Model Evaluation:

Job Posting Verification: Develop metrics to measure the accuracy of job posting verification. Evaluate false positive and false negative rates to ensure the effectiveness of the verification process.
Recommendation System: Use evaluation metrics like precision, recall, and F1-score to assess the performance of the personalized job recommendation system.

5. Conclusion and Results:

We will document and prepare detailed reports all the methods, algorithms, data preprocessing steps and evaluation metrics

By following these methods, we will create a robust Integrated Job Post Verification and Personalized Job Recommendation System, ensuring the authenticity of job postings while providing tailored job recommendations to users based on their personalized search query.

Discussion:

LITERATURE REVIEW:

Data cleaning and processing is the key aspect in the data mining process. The valuable insight about it is provided in research paper [1] which addresses the issue such as missing values, noise, incompleteness, inconsistency, and outliers. Various techniques like cleaning, integration, transformation, and reduction are provided in the paper to enhance data efficiency and facilitate more efficient knowledge discovery.

So, to get the better understanding about recommendation system more information is provided in paper [2] and [3] where it focus on methods such sampling, dimensionality reduction, and distance functions in data preprocessing overview of commonly used data mining methods in Recommender Systems, namely classification, clustering, and association rule discovery. The research employs data cleaning and various Machine Learning techniques, with the Random Forest Classifier demonstrating the highest prediction accuracy.

The paper [4] emphasizes the need for accurate job recommendations based on user profiles and preferences and provides insights into how to tailor your job recommendation system to individual job seekers. The approach of mining rules and content-based matching can be integrated into your recommendation engine to enhance personalization.

The paper [5] on detecting fraudulent job advertisements is essential for the job post verification aspect of your integrated system. It discusses the use of machine learning and classification techniques, including Gradient Boosting, to identify fraudulent job postings. You can incorporate these techniques to filter out fake job posts and ensure the integrity of the job listings.

The paper [6] on predicting fake job postings can be utilized to enhance the job post verification process. By leveraging data mining techniques and deep neural networks, you can develop a more robust mechanism for identifying and flagging potentially fraudulent job advertisements.
By combining the knowledge from these papers, we will create an integrated system that verifies job postings for authenticity, detects fraudulent ones, and offers personalized job recommendations to job seekers based on their profiles and preferences. This system will not only help job seekers find genuine job opportunities but also protect them from potentially harmful or misleading listings.

REFERENCES:

Research Papers

- https://www.researchgate.net/publication/319990923_Review_of_Data_Preprocessing_Techniques_in_Data_Mining [1]
- https://www.researchgate.net/publication/225924875_Data_Mining_Methods_for_Recommender_Systems [2]
- https://www.sciencedirect.com/science/article/pii/S2666412721000489 [3]
- https://gdeepak.com/thesisme/Applying%20Data%20Mining%20For%20Job%20Recommendations.pdf [4]
- https://link.springer.com/article/10.1007/s00146-022-01469-0 [5]
- https://www.researchgate.net/publication/349884280_A_Comparative_Study_on_Fake_Job_Post_Prediction_Using_Different_Data_mining_Techniques [6]