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.

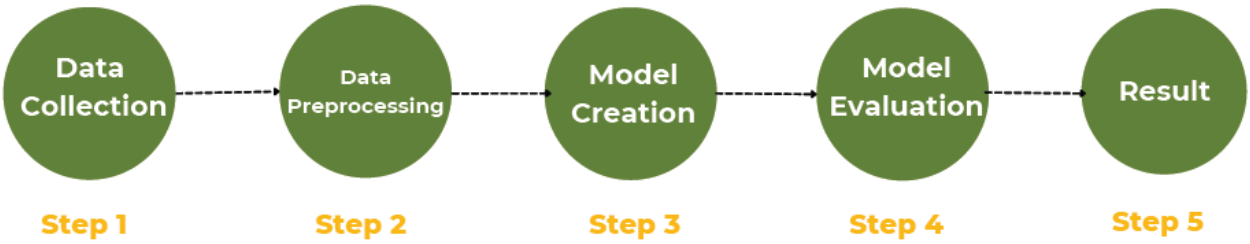
In addition, our Personalized Job Recommendation System will use advanced data mining algorithms to suggest jobs that match a person's skills and preferences. It will look at job titles, locations, specific details, and job descriptions to find the best matches, again making use of data mining for enhanced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

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:

   In our data cleaning process, we addressed numerous missing values, particularly in the 'location', 'company_profile', and 'requirements' columns, by substituting NaNs with empty strings where appropriate. We honed in on the 'location' data to extract a distinct 'Country' column, discarding any non-essential abbreviations. For the 'salary' column, we translated range values into averages to fill in gaps. To ensure the quality of our text analysis, we selectively removed any rows with missing 'description' entries. This meticulous approach was crucial for enhancing the precision of job matching in our system.

   2.2 Text Processing:

In our text processing stage, we combined key columns like 'title', 'description', and 'company_profile' into a 'job_text_info' column, creating a single, unified source of textual data for each job listing. We then standardized this text by converting it to lowercase and removing punctuation to eliminate noise, ensuring uniformity.The text was tokenized into words, and common stop words were removed to focus on the most meaningful elements. We applied lemmatization over stemming, considering the context to obtain the root form of each word. Finally, we recombined the tokens into a coherent text string, preparing the data for the feature extraction and machine learning phases.

```
# 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	
		Commissioning Machinery	US, IA,			Valor Services provides	Our client, located in Houston is actively	Implement pre-commissioning and				

```
#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
```

We also performed one-hot encoding and the comparison of how data looked before applying one-hot encoding and after applying one-hot encoding using get_dummies

method can be seen below:

Before one-hot encoding using get dummies:

```
#before dummies
job_data_new.head()
```

	title	department	salary_range	company_profile	telecommuting	has_company_logo	has_questions	employment_type
0	Marketing Intern	Marketing	NaN	We're Food52, and we've created a groundbreaki...	0	1	0	Other
1	Customer Service - Cloud Video Production	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	0	1	0	Full-time
2	Commissioning Machinery Assistant (CMA)		NaN	Valor Services provides Workforce Solutions th...	0	1	0	
3	Account Executive - Washington DC	Sales	NaN	Our passion for improving quality of life thro...	0	1	0	Full-time
4	Bill Review Manager		NaN	SpotSource Solutions LLC is a Global Human Cap...	0	1	1	Full-time

Then we applied one-hot encoding on title and department columns:

```
#applying one hot encoding on the title and department columns
features = pd.get_dummies(job_data_new, columns=['title','department'])
```

After applying one-hot encoding using get dummies:

```
#after dummies
features
```

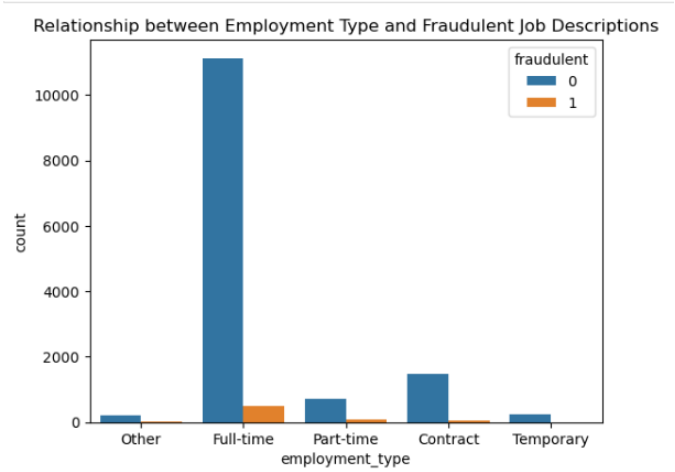
	telecommuting	has_company_logo	has_questions	title_ Electrician	title_ Environmental Technician I	title_ Piping Material Engineer	title_ Discipline Manager Civil, Structural, Marine, Architectural	title_ FEA Senior engineer	Man Man Organ Engi
0	0	1	0	0	0	0	0	0	
1	0	1	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	0	
3	0	1	0	0	0	0	0	0	
4	0	1	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
17875	0	1	1	0	0	0	0	0	
17876	0	1	1	0	0	0	0	0	
17877	0	0	0	0	0	0	0	0	
17878	0	0	1	0	0	0	0	0	
17879	0	1	1	0	0	0	0	0	

2.3 Feature Extraction:

In the feature extraction phase, we utilized TF-IDF vectorization on 'company_profile' and 'job_text_info' columns to transform the text data into numerical format, limiting to the top 5000 features for compatibility with machine learning algorithms such as Logistic Regression and Neural Networks.

2.4 Exploration Data Analysis:

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



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.

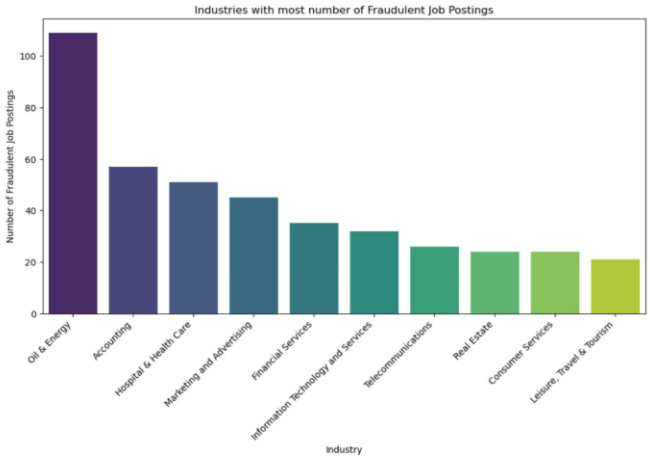


Figure 1

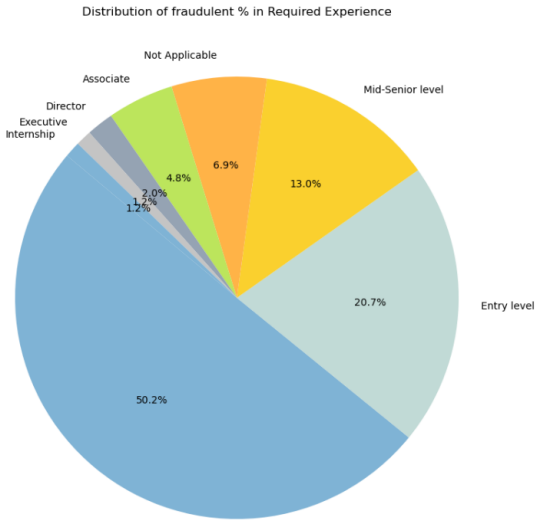


Figure 2

Figure 1 displays a bar chart revealing that the Oil & Energy industry has the highest frequency of fraudulent job postings, while the Real Estate sector has the fewest occurrences. This insight is crucial for job seekers to exercise caution in these industries, particularly in Oil & Energy.
In Figure 2, a pie chart indicates that the majority of fraudulent postings target Entry-level positions, emphasizing a trend where scams are more prevalent for candidates with less experience. Higher-level positions, such as Executive and Director roles, seem less affected, likely due to factors like 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 Logistic Regression:

- Model Selection: Logistic regression, a widely used technique for binary classification, was employed due to its effectiveness in handling both categorical and numerical variables.

- Implementation: The logistic regression model was implemented using the Scikit-Learn package, a versatile Python machine learning toolkit.
- Data Processing: Feature engineering was performed to enhance the model's performance. Unnecessary columns were removed during this stage.
- Text Data Transformation: For text columns like "company profile," Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was applied to convert textual data into a numeric format.
- Categorical Data Transformation: Categorical data, such as "title" and "department," underwent one-hot encoding to convert them into a binary matrix.

3.2 K-Nearest Neighbors (KNN):

- Algorithm Utilized: The K-Nearest Neighbors (KNN) algorithm was employed as part of supervised machine learning in our project.
- Library and Class: We utilized the KNeighborsClassifier from the scikit-learn library to implement the KNN algorithm.
- Classification Approach: KNN classifies a new data point based on the majority class of its k-nearest neighbors in the feature space.
- Parameter Setting: The 'k' value, representing the number of neighbors, was set to 5 in our implementation.
- Characteristic Utilization: KNN's less sensitivity to irrelevant or redundant features made it well-suited for our dataset.
- Parameter Optimization: The crucial 'k' value was optimized using grid search and random search techniques tailored to our dataset.
- Effectiveness: KNN is particularly effective for tasks with non-linear decision boundaries and proves valuable in scenarios featuring local patterns or clusters in the data.
- Suitability for Dataset Size: Its suitability for datasets of moderate size aligns well with our project, balancing computational efficiency and accuracy.

3.3 Random Forest Classifier:

- Model Selection: Chose the RandomForestClassifier from scikit-learn for ensemble learning.
- Handling Diverse Data: Selected for its effectiveness in managing diverse data types.
- Overfitting Mitigation: The model's ensemble nature helps mitigate the risk of overfitting.
- Non-linear Relationship Capture: Excels in capturing non-linear relationships and feature interactions.

- Decision Tree Aggregation: Utilizes predictions from multiple decision trees trained on different data subsets.
- Complex Pattern Recognition: Well-suited for unraveling complex patterns and relationships within the data.
- Decision Tree Quantity: Employed 100 decision trees for a balance between efficiency and performance.
- Hyperparameter Tuning: Utilized grid search for optimal parameter selection to enhance model performance.

3.4 Neural Networks:

- Model Selection: Utilized Multi-Layer Perceptron (MLP) Classifier for neural network-based classification.
- Deep Learning for Text Data: Chose DL techniques due to their effectiveness in handling complex patterns in text data.
- Scikit-Learn MLPClassifier: Leveraged the MLPClassifier from the scikit-learn package.
- Text Data Handling: MLPClassifier chosen for its ability to handle the complicated nature of textual data.
- Neural Network Architecture: MLP's architecture, with multiple layers and neurons, suited for learning non-linear decision boundaries.
- Text Data Formatting: Conducted text cleaning procedures using Natural Language Processing (NLP) techniques.
- Data Preprocessing: Ensured correct formatting of text data before inputting it into the MLPClassifier.

3.5 Recommendation for Jobs:

As part of our research, we've created a job suggestion system that analyzes and interprets user search queries by applying Natural Language Processing (NLP) techniques. The recommendation feature operates as follows:

Search Query Analysis: Using natural language processing (NLP) techniques, we are able to read and understand a user's search query—for example, "Looking for a job in marketing"—and determine the user's intention.

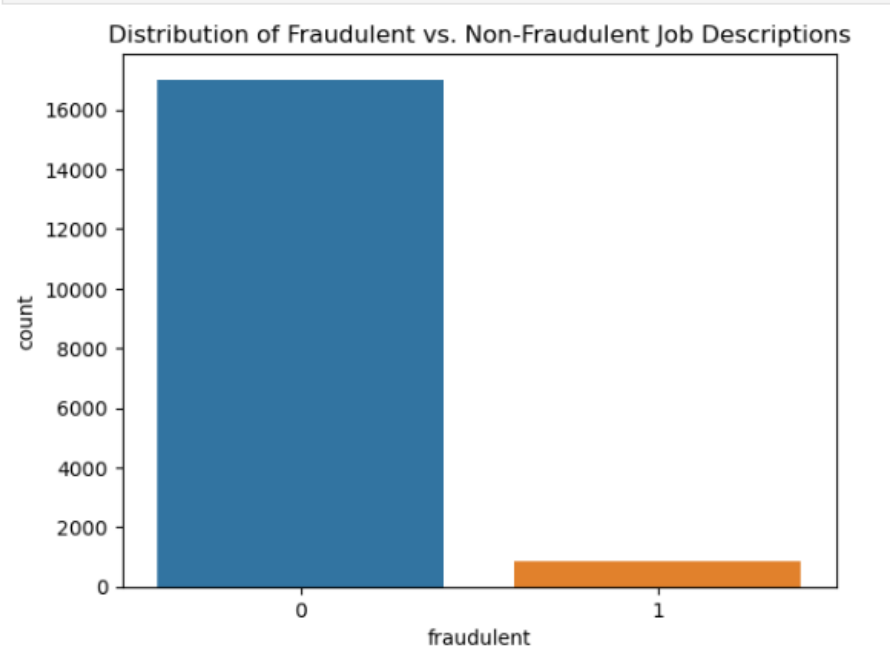
Cosine Similarity matching: Next, we use this method to search the job listing dataset for matches to the query. The cosine similarity between the user's query and the job descriptions in the dataset is calculated to achieve this. Regardless of the size of the documents, cosine similarity is a metric used to assess how similar they are.

Extracting Relevant Listings: The system pulls rows from the dataset that match the user's query based on the cosine similarity scores. The job listings in these rows most closely match the search terms entered in the query.

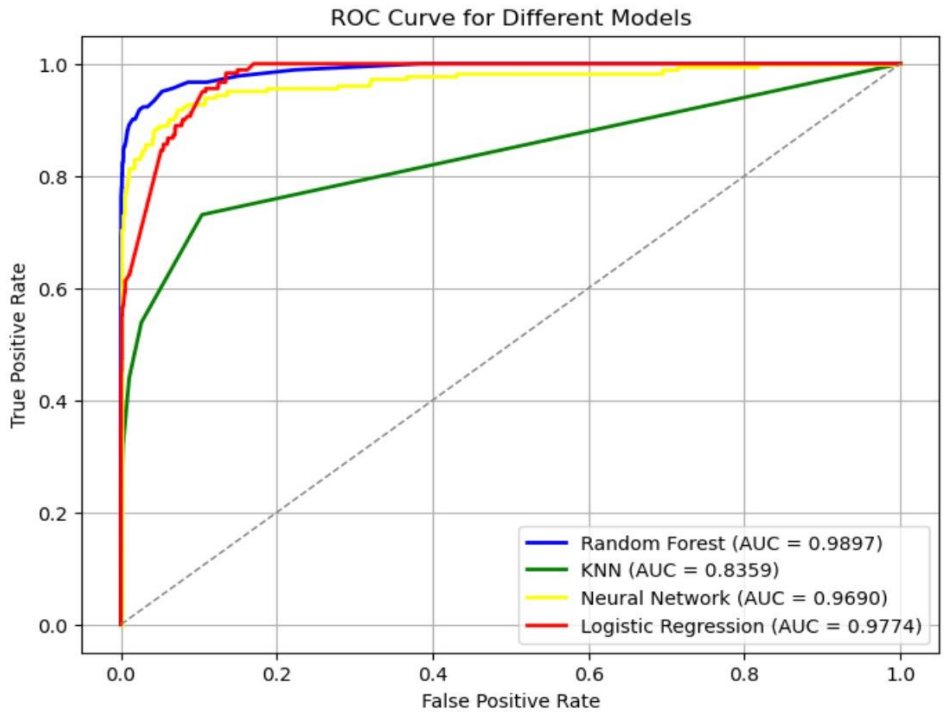
Result Presentation: The result is a list of job titles with pertinent information about the department, industry, function, and description included. To ensure that the recommendations are closely aligned with the user's interests and the intent expressed in their search query, the job listings that have the highest cosine similarity to the user's query are displayed.

4. **Model Evaluation and Results:**
   Given the context, the ROC-AUC Score was been selected as the preferred evaluation metric for the model due to the following reasons:

- The ROC-AUC Score is particularly useful for binary classification problems. It provides a single measure to summarize the performance of the model across all classification thresholds.
- It is also robust in situations where there is class imbalance, as it evaluates the model's ability to distinguish between the two classes without being affected by the uneven distribution of the classes. And in our case there was significant uneven distribution between two classes of fraudulent and non-fraudulent job postings. This can be seen in below bar graph visualization where imbalance between job descriptions classified as fraudulent and those categorized as non-fraudulent is seen. This stark contrast suggests that the dataset contains a significantly higher number of non-fraudulent job postings compared to fraudulent ones.



The value of the AUC lies between 0 and 1, with 1 indicating perfect classification and 0.5 representing a model with no discriminative ability, equivalent to random guessing. Typically, the closer the ROC-AUC Score is to 1, the better the model is at predicting fraudulent job postings in this scenario.



The ROC curve graph visually represents the true positive rate against the false positive rate for different models at various thresholds. According to the above ROC-graph, Random Forest and Logistic Regression have curves that reach towards the top-left corner, reflecting their higher AUC values, which suggests they are better at distinguishing between classes than KNN. The KNN curve is lower, which corresponds with its lower AUC value from the table, confirming its relatively poorer performance in separating the classes. The Neural Network (MLPClassifier) curve is not the highest, but it is above the KNN and close to the Logistic Regression, aligning with its high accuracy and good AUC value.

The model evaluation and results for the models are as followed:

MODEL	ACCURACY	ROC-AUC SCORE
Logistic Regression	96.812%	0.977
KNN	96.14%	0.714
Random Forest	98.014%	0.804
MLPClassifier	98.35%	0.884

In assessing our models, the KNN classifier demonstrated a commendable accuracy of 96.14%, indicating its strong ability to make accurate predictions. However, its ROC-AUC score of 0.714 suggests the potential for improvement in distinguishing between classes compared to other models. On the other hand, the Random Forest classifier outperformed, achieving a high accuracy of 98.014% and a robust ROC-AUC score of 0.804. This underscores its effectiveness in predicting outcomes and discriminating between positive and negative classes. The MLPClassifier exhibited remarkable performance with an accuracy of 98.35% and a high ROC-AUC score of 0.884, showcasing its effectiveness in classifying data with a balanced true positive rate and a low false positive rate. Lastly, the Logistic Regression model, thoroughly examined, yielded an accuracy of 96.812% and an impressive ROC-AUC score of 0.977, highlighting its reliability in accurately classifying job postings. These findings collectively emphasize the strengths and areas for potential improvement of each model within the context of our project's goals and dataset.

But the values we got using models were slightly different from the values we got in the ROC Curve graph. The discrepancy between the ROC-AUC values presented in the table and those depicted in the graph might be due to several factors like:

1) The ROC curve was created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings than those used during model evaluation
2) The graphical representation of the ROC curve may be an approximation, and the actual AUC calculation seen in the below table might be more precise, leading to differences when visually comparing the two.
3) The ROC-AUC score is an aggregate measure of performance across all possible classification thresholds. It's a single number that condenses the information of the ROC curve into a summary statistic while the graph provides a visual representation that may highlight specific performance at various thresholds which might not be fully captured by the single AUC score.

In the context of our project, the Logistic Regression and MLPClassifier are the standout models, offering a strong balance between accuracy and the ability to distinguish between fraudulent and non-fraudulent job postings.

Results for Recommendation:

The recommendation approach makes sure the user sees the most relevant job listings because the job listings are chosen based on how similar the text is to the search query. By giving users a customized experience that appropriately matches the kinds of jobs they are looking for, this strategy seeks to increase user satisfaction.

```
user_query = "interested in finance"
result_indices = search_for_jobs(user_query, job_text_x, vectorizer, job_data)

# Display the indices
result_indices
```

job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_compan
17720	CUSTOMER SERVICE REP	US, TX, DALLAS				DescriptionJob Title: Customer Service Represe...	HIGH SCHOOL DIPLOMA	HEALTH,DENTAL INSURANCE , 401K , STOCK PLAN FO...	0	
7831	Position Finance Assistant	AU, VIC, Melbourne		25000-30000		We have positions available for confident, out...	Personal attributes would include: • Intermedi...	Benefits include: · Base wage of $450 per week...	0	
12722	Finance Analyst (Fixed term contract)	GB, LND, Shoreditch	Finance		ustwo offers you the opportunity to be yoursel...	Ustwo London are looking for an experienced Fi...	Technical requirements/skillsBring numbers to ...	Above anything we are people centred company t...	0	
7643	Banking and Finance Attorney	US, NC, Charlotte	Legal			Special Counsel's Charlotte office is searchin...			0	
4238	Senior Associate Corporate Finance & Planning	PK, IS, Islamabad	Finance			Position Title: Senior Associate Corporate Fi...	Essential Requirements:-- Experience in IPP (I...		0	

CONCLUSION AND RESULTS:

The project underscored the critical importance of data processing and feature engineering in the development of Machine Learning (ML) models, noting significant impacts on model accuracy. Through the diligent application of these preprocessing steps, the project was successful in implementing various ML models, including Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and MLPClassifier. Notably, Logistic Regression performed exceptionally well, achieving a test ROC-AUC of 97.7% and an accuracy of 96.81%, which aligned with the project's objectives.

Additionally, the project featured a search query analysis for job recommendation that effectively utilized Natural Language Processing (NLP) techniques and cosine similarity. This approach enabled the delivery of personalized and relevant job recommendations, enhancing the user experience by matching their search queries with the most suitable job listings.

In conclusion, the project achieved its goal of creating an effective job recommendation system and demonstrated the value of careful model selection and evaluation in addressing the challenges of job classification and recommendation.

DISCUSSION:

In the Discussion section of our report, we acknowledge the critical importance of thorough data cleaning and processing as fundamental steps in data mining, as highlighted in the literature [1]. This research points out issues like missing values and outliers and suggests methods such as data cleaning and transformation to improve the overall quality and usefulness of data.

Further exploration into recommendation systems is provided in studies [2] and [3], which detail preprocessing techniques such as sampling and dimensionality reduction. These studies review common data mining methods like classification and clustering, used in building recommender systems. They also note that the Random Forest Classifier showed promising results in terms of prediction accuracy.

The research presented in paper [4] stresses the importance of accurate job recommendations that reflect user profiles and preferences. It offers insights into enhancing a recommendation system's personalization by incorporating content-based matching techniques.

The necessity of detecting fraudulent job postings, an essential aspect of job post verification, is discussed in paper [5]. It describes how machine learning techniques, including Gradient Boosting, can be used to identify fake job listings, suggesting these as valuable additions to the verification process.

Lastly, paper [6] discusses the prediction of fraudulent job postings, emphasizing the use of deep learning for developing a reliable detection system.

Collectively, these papers inform the creation of an integrated system for our project. This system will not just recommend jobs based on user preferences but also ensure the authenticity of the listings, protecting job seekers from misleading information.

AUTHOR CONTRIBUTION:

Author	Contribution
Sumeet	Worked on data cleaning and text processing, EDA and visualizations, Logistic Regression model and Technical Documentation.
Himanshi	Worked on feature extractions, EDA and visualizations, KNN, Random Forest Classifier Model and Technical Documentation.
Shubhangi	Worked on text processing, MLP Classifier model and Job recommendation algorithms and Technical Documentation.

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]