LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NI Artificial Intelligence

25% Individual Coursework

Submission: Final Submission

Academic Semester: Autumn Semester 2025

Credit: 15 credit semester long module

Student Name: Shidharth Kharga

London Met ID: 23049367

College ID: NP01CP4A230377

Assignment Due Date: 21/01/2026.

Assignment Submission Date: 21/01/2026

Submitted To: Mr. Binod Bhattarai

GitHub Link	*https://github.com/justwannar0ck/Job_Recomendation_ML.git*

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

With the modern digital age, one of the most important branches of Artificial Intelligence (AI) is Machine Learning (ML) that is transforming the manner of work and decision-making in industries. In essence, Machine Learning is not like the conventional software engineering with its paradigm shift in rule-based programming to the data-driven learning. In contrast to writing down the instructions to follow in every possible situation, ML algorithms employ statistical methods to establish patterns in a large amount of data so that systems can become more efficient based on experience.

This coursework explores the application of machine learning and natural language processing for the recommendation of jobs. “Artificial Intelligence is the development of computer systems able to perform tasks that would typically require human intelligence.” “Machine Learning is a type of artificial intelligence that involves training models on data in order to make predictions or decisions without being explicitly programmed.” Supervised machine learning is a type of machine learning in which models learn to predict or classify data by training with labelled examples. Classification or Regression are some of the tasks that fall into supervised machine learning.

NLP is the area that deals with the application of ML to text-based job data such as job titles and descriptions with the aim of developing models for recommending jobs. NLP is applied using the job-skill set data from Hugging Face Datasets. This data set is made up of approximately 1.17K job entries with features such as job title, description, category (categorized into 5 broad categories of jobs), and finally the job skill set. (Batuhanmtl, 2024)

2. BACKGROUND OF THE STUDY

2.1. Dataset Domian and Context

The digital information is in the digital era of the recruitment industry with an enormous flow of data. Millions of jobs are being posted, resumes are produced by the candidate, and skill measurements are being done every day, forming a rich and unstructured pool of information. An example of this field is the dataset that this paper uses batuhanmtl/job-skill-set. It comprises of unstructured textual data fields including Job Titles, Skill Sets and Job Descriptions. This area is under the auspices of Text Classification and Natural Language Processing (NLP), in which the key issue is to derive useful semantic relations out of free-text inputs that differ considerably in length, form, and vocabulary.

2.2. The Existing Problem

The matching of the candidates to the appropriate job roles is not efficient even though there is a lot of data available.

- **Terminology Fuzziness:** Various companies apply different terms to the same positions (e.g., the terms "Software Engineer" and "Application Developer" are used interchangeably) and therefore the search by key word is not that reliable.
- **High Dimensionality:** The job descriptions have thousands of distinct words and can be regarded as a noisy environment in which valuable abilities are drowned in the overall corporate language.
- **Scalability Problems with Manual Processes:** The traditional approaches have human recruiters to manually read these descriptions. Not only is this method time-consuming and expensive, but it also has the likelihood of being biased in thought and fatigued. The human level verification is a bottleneck as the number of applications increase hence missing chances of both the applicant and the employer.

2.3. Importance of Machine Learning Solution

These scalability issues are addressed by modern industries through automating them with data. Simple rule-based approaches (e.g. simple if-else keyword matching) are not very good in dealing with complex and non-linear patterns in natural language. As an example, a rule-based system may not identify that an applicant who has skills in pandas and scikit-learn can be a fit to a Data Scientist position when the word Data Science does not appear on his/her profile.

A strong alternative can be provided by Machine Learning that can learn probabilistic relationships between skills and job categories. With training algorithms on past data, an ML based system will be able to generalize patterns like knowing that React and Node.js are highly correlated with Web Development despite the vague job title. Having open datasets allows one to perform a comparative analysis of various algorithms (including Naive Bayes, Logistic Regression, and Random Forest) to have a more profound insight into which mathematical methods are most effective in dealing with the sparsity and complexity of recruitment data. Such a shift towards manual curation to the algorithmic prediction is what is necessary to construct real-time scalable career recommendation systems.

3. RELATED WORKS

Machine Learning and its application in the recruitment and text classification has been a research topic of considerable focus. The available literature has mostly concentrated on resume parsing, prediction of job titles and comparison of the classification algorithms.

Comparative Analysis of Classification Algorithms in a recent study of job prediction, a recommendation system was created to direct students to the appropriate career paths. Their study was aimed at comparing different classification algorithms such as Decision Trees and Logistic Regression. They discovered that although linear models such as the Logistic Regression were offering a good baseline, non-linear models (Decision Trees) offered a better prediction accuracy of 87.5 percent where the range of skill sets were involved. The given work demonstrates that it is crucial to test various algorithms to identify the best ratio between the speed and accuracy of text-based predictions. (Sahu, 2023)

Efficiency of Naive Bayes in Text Classification was examined and the particular use of Naive Bayes algorithm in the text classification of academic information systems. They showed that the Naive Bayes form of classification is remarkably efficient and effective with high-dimensional text data and their results showed that it had a testing accuracy of 97.5% in their area. The algorithm, however, they pointed out, is highly dependent on the independence assumption that is, it assumes that each word is independent of the others and as such it occasionally fails to understand the context of a complicated phrase (e.g., it fails to differentiate between Java the coffee and Java the language). (Humaidi, 2023)

Cosine Similarity versus TF-IDF – the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was offered in TF-IDF and Cosine Similarity for Job Matching as a Resume Parser and Job Recommendation System. Their effort was on transforming the unstructured text in the resume into numerical vectors and ranking job relevance with Cosine Similarity. The authors found that TF-IDF is a better way to extract keywords than the simple "Bag of Words" model since it reduces the weight of common words (such as the, and) and emphasizes rare and meaningful skills, which are essential in the process of job matching. (Chandak, 2024)

Hierarchical Classification of Recruitment to solve the problem of complexity in job titles a Hierarchical Job Classification technique was brought about. Their study held that flat classification (when it is necessary to predict 1,000 job titles simultaneously) is error prone. They instead suggested a multi-tier model which initially forecasts the general industry (e.g. "Engineering") and then digs down to positions (e.g. "Backend Developer"). This approach was effective to decrease the number of false positives and enhance the capacity of the system to accommodate new unseen job titles, which was one of the primary motivators of the hierarchical structure that was appreciated in this project. (Kabir, 2025)

Impact of Resume Parsing on Recruitment Efficiency investigated the semantic analysis of resumes of candidates based with the help of Natural Language Processing (NLP). They automated the mining of skills, education and experience and formed a structured database of unstructured PDF files. The research also discovered that this first screening process was automated and thus took about 40 percent less time to hire, which proves the practical importance of ML based career assistants in contemporary HR departments. (Jain, 2024)

4. PROBLEM STATEMENT

The main problem of the modern talent acquisition is the effective and precise correspondence of the unstructured skill set of the candidate with the job opportunities in the large and diverse market. The conventional automated systems usually utilize ludicrous keyword matching, which does not reflect the semantic subtlety of the job descriptions or the hierarchical structure of the professional sectors. Additionally, when there are noisy data e.g. job titles that are not well trained and have a limited number of samples, it is likely that the model will not perform well and give inaccurate advice.

This paper will attempt to solve these inefficiencies by constructing a hierarchical machine learning pipeline that will first classify an industry into a broad category and then classify it further into a job title. To identify which of the three different algorithms - Naive Bayes, Logistic Regression and Random Forest best handles high-dimensional text data and offers the best confidence in real-world career advice, the research assesses the accuracy and performance of all the three algorithms. Using data filtering thresholds and N-gram feature engineering, this paper attempts to reduce the errors in prediction and create a solid framework of automated career path analysis.

5. RESEARCH QUESTIONS & OBJECTIVES

5.1. Research Questions

The research aims at determining the following questions to assess the effectiveness of the career assistant:

- **Q1: Algorithm Superiority:** Which machine learning model between Naive Bayes, Logistic Regression, and Random Forests shows the highest accuracy in the classification of multi-class jobs?

- **Q2: Hierarchical Performance:** Does the model do a much better job than predicting the broad Category (Industry) than predicting the granular Job Title?

- **Q3: Feature Impact:** What is the extent to which preprocessing methods, namely, N-gram vectorization and frequency thresholding, enhance job match reliability?

- **Q4: Confidence Correlation:** What is the relationship between the confidence score of the model, Match Confidence and the specificity of the skills of the user at the input?

5.2. Project Objectives

The main aim of the study is to develop a working AI Career Assistant. This will be done in the following objectives:

- **Objective 1. Multi-Level Pipeline Development:** To create and build a hierarchical classification pipeline and process unstructured job data into two different target variables – category and job title.

- **Objective 2. Comparative Algorithms Analysis:** To train, test and compare the performance of three baseline learners (Naive Bayes, Logistic Regression and Random Forest) to determine which engine is most robust to use in the application.

- **Objective 3. Data Optimization:** To perform text normalization and frequency-based threshold (minimum 10 samples per class) to guarantee the high quality of training data and reduce model noise.

- **Objective 4. Interactive Tool Creation:** To design an interactive Command Line Interface (CLI) to use the most effective model to deliver real-time suggestions of career paths and confidence measures.

- **Objective 5. Empirical Reporting:** To visualize and report the performance difference of the three models and give an answer as to why particular algorithms perform better on high-dimensional text data.

6. DATASET DESCRIPTION

Dataset source: Hugging Face Datasets (Batuhanmtl, 2024)

Type: Unstructured Textual / Tabular (Apache Parquet Format)

Instances: Approximately 1,170 total rows (Initial). Final count varies based on the frequency threshold (>=10 samples per class).

Features: job_description (detailed responsibilities), job_skill_set (key technical skills), and category (industry label).

Target Variable: Hierarchical Targets – category (Level 1 - Broad Industry) and job_title (Level 2 - Specific Role).

Missing Values: Handled via Listwise Deletion – rows with null values in the job title, skills and description are removed to maintain training integrity.

6.1. Summary

Domain Purpose: The data is in the Human Resources and Recruitment domain. It is mainly aimed at supporting the creation of automated job recommendation systems by offering a connection between technical skills set and their relevant job titles. Using practical job advertisements, the dataset could be used to train models that are able to understand the semantic connection between the background of a candidate and the industry-standard occupations.

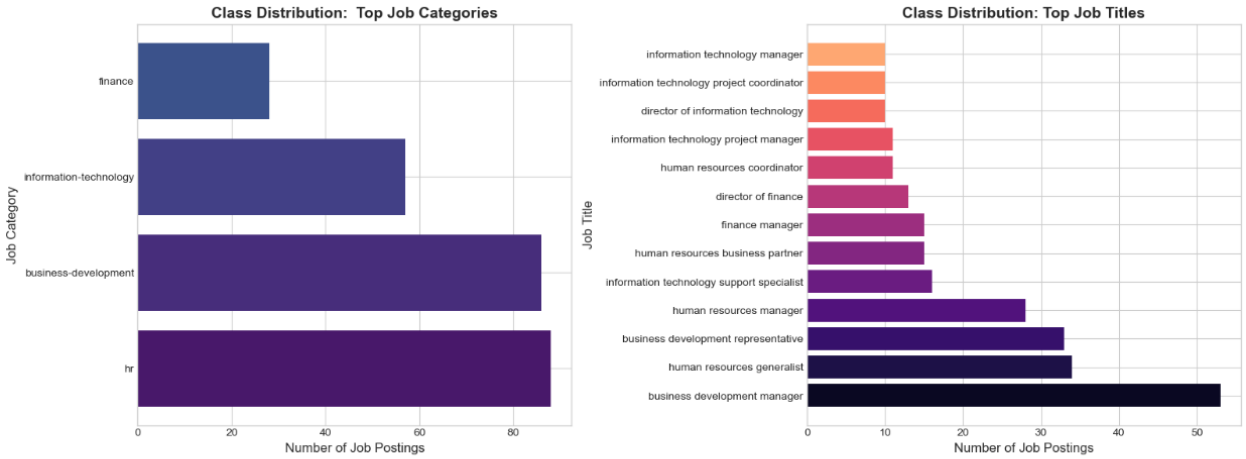
Limitations:

- **Class Imbalance:** The data is skewed considerably in favour of some job titles (e.g. IT-related jobs), but some have limited numbers. This necessitated the use of a frequency threshold to stabilize the models.
- **Tech-Centric Bias:** Most of the skills and descriptions are technology-based and this could restrict the application of the model in forecasting other positions other than the technical ones.
- **Noise:** Raw job descriptions usually have generic filler text of corporate nature, thus requiring a good preprocessing and TF-IDF weighting to extract meaningful keywords.

Suitability: The limitations notwithstanding, the dataset is very appropriate in this study due to the presence of paired features (job_skill_set and job_description). This enables the model to learn key words as well as unstructured prose. Moreover, it has a broader category column, which is a prime target of this project since it can be tested with the Hierarchical Classification strategies.

7. DATASET VISUALIZATION

7.1. Bar Plot – Class Distribution




```
Total unique categories: 4
Total unique job titles: 13
```

Figure 1: Bar Plot

Purpose: Displays a frequency distribution of the target classes (job categories and job titles) in our data set.

The bar charts indicate the distribution of classes of our two variables of interest. The top categories of jobs are shown in the left chart whereas the top job titles are shown in the right chart. The class imbalance in both targets is evident, as some of them have much more samples than others (e.g., IT, Sales). This imbalance is worth mentioning because it may influence the performance of the model, models are likely to be biased to majority classes. To this end, we employed a threshold filter (minimum 10 samples per class) and stratified train-test splitting to ensure the proportions of classes.

7.2. Histogram – Numerical Variable Distribution

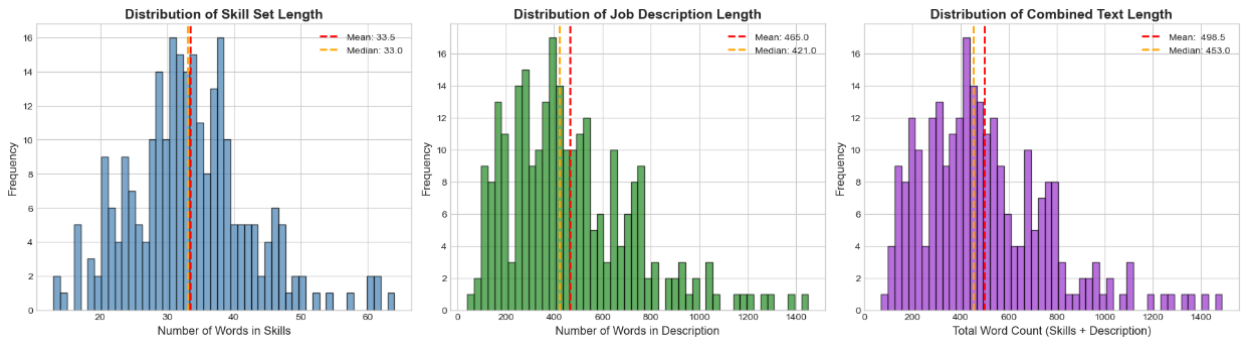


Figure 2: Histogram

Purpose: Visualizes the distribution of derived numerical characteristics (length of text) to learn the characteristics of data.

We obtained numerical features by counting word lengths since our major features are text-based (skills and job descriptions). The histograms show:

- The length of skill sets is skewed right with majority of the entries having 10-50 words.
- The job descriptions are more elaborate and may take between 100-400 words.
- The sum of the text length displays the richness of features that can be used in TF-IDF vectorization.

The red dotted line represents the average, and the orange colour represents the median. The mean and median differences substantiate the right skewness of the distributions, which means that there are some long job postings.

7.3. Correlation Heatmap – Feature Relationships

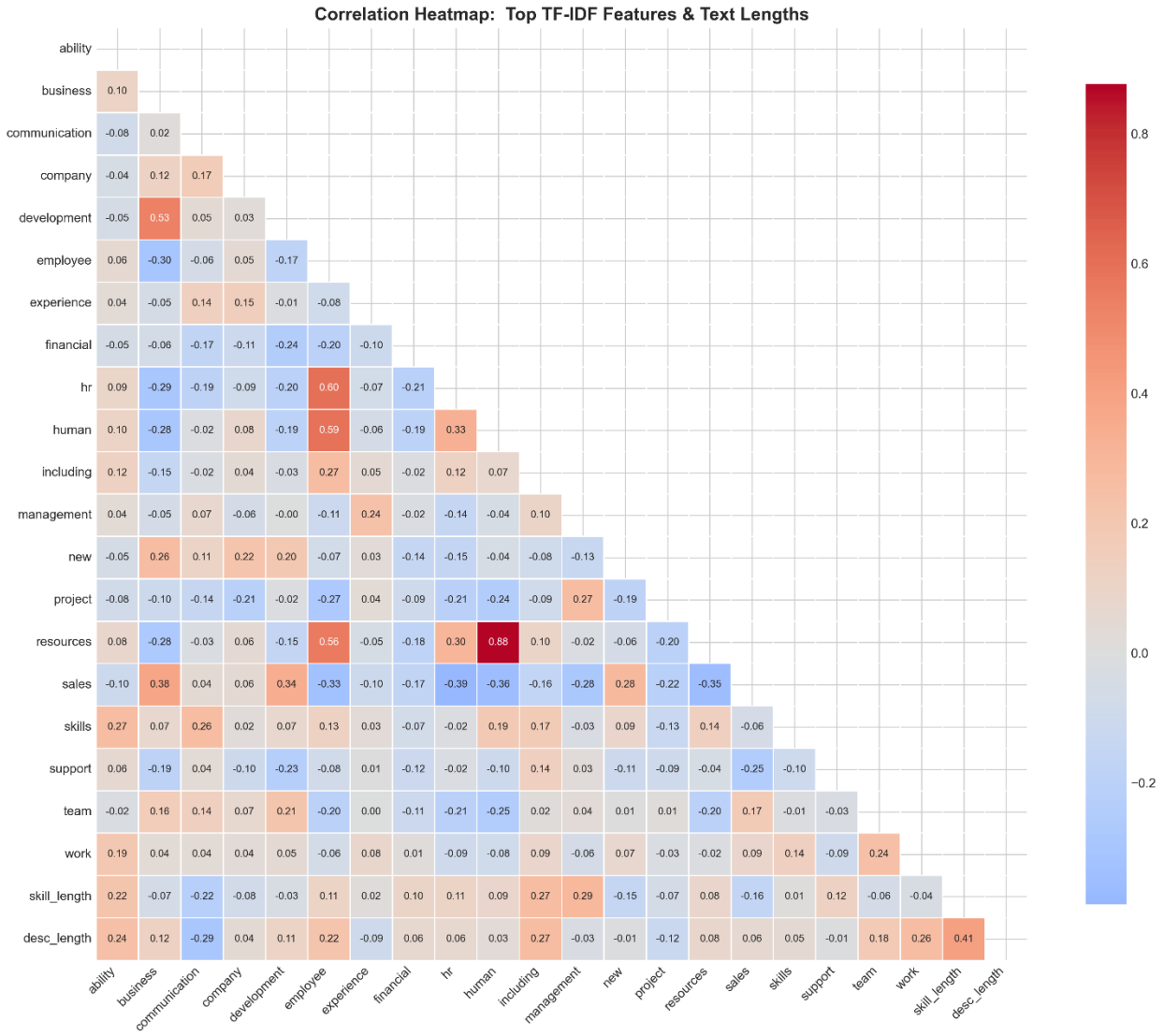


Figure 3: Heatmap

Purpose: Visualizes the associations between the best TF-IDF features and the numerical features derived.

The correlation heatmap shows the relations between the 20 best TF-IDF features and the 20 largest text length measure. Key observations:

- The majority of TF-IDF features inter-correlate (near 0) implying that they extract independent information, which is an ideal machine learning feature.
- There is moderate positive correlation between skill_length and desc_length, which indicates that long lists of skills tend to be accompanied by more detailed descriptions.
- Some domain specific words (e.g. management, experience) exhibit some positive correlations, which is expected due to similar vocabulary in job posting.

The poor overall correlation of features indicates that our TF-IDF vectorization produces a variety of non-redundant features to be used in classification.

7.4. Boxplot – Outlier Detection

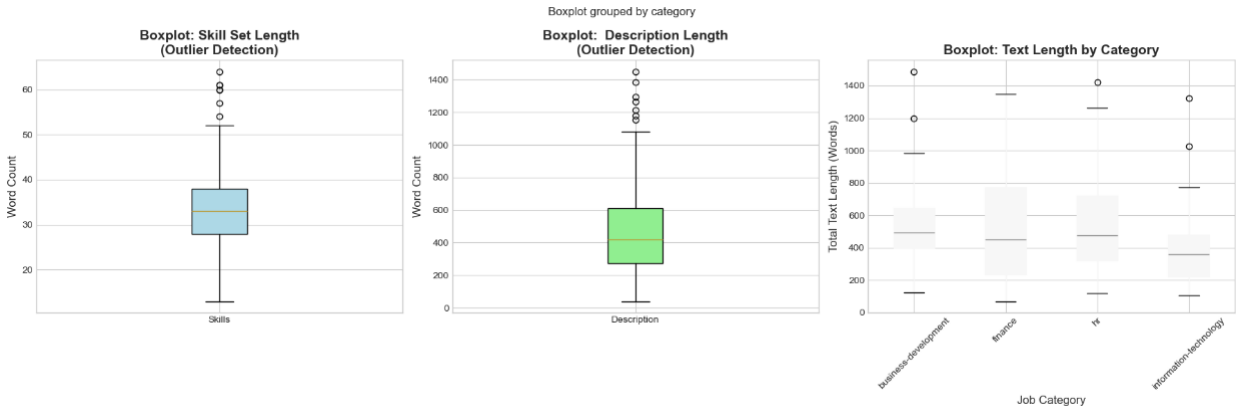


Figure 4: Boxplot

Purpose: Determines the outliers in the data that can influence the training of the model.

Boxplots are used to detect the outliers in our derived features:

- The length of the skill set has a few upper outliers (job advertisements with very long skill sets).
- Description length has many outliers at both extremes with some very short and others very long.
- The boxplot based on categories indicates that various categories of jobs have different average text lengths (e.g., positions that require a lot of technical writing might have longer descriptions of the skills).

The dataset did not filter outliers because they provide valid job postings. Nonetheless, they have some departure in model predictions due to their existence. IQR technique determined that some 7% of data points were outliers.

7.5. Scatter Plots – Relationships between features

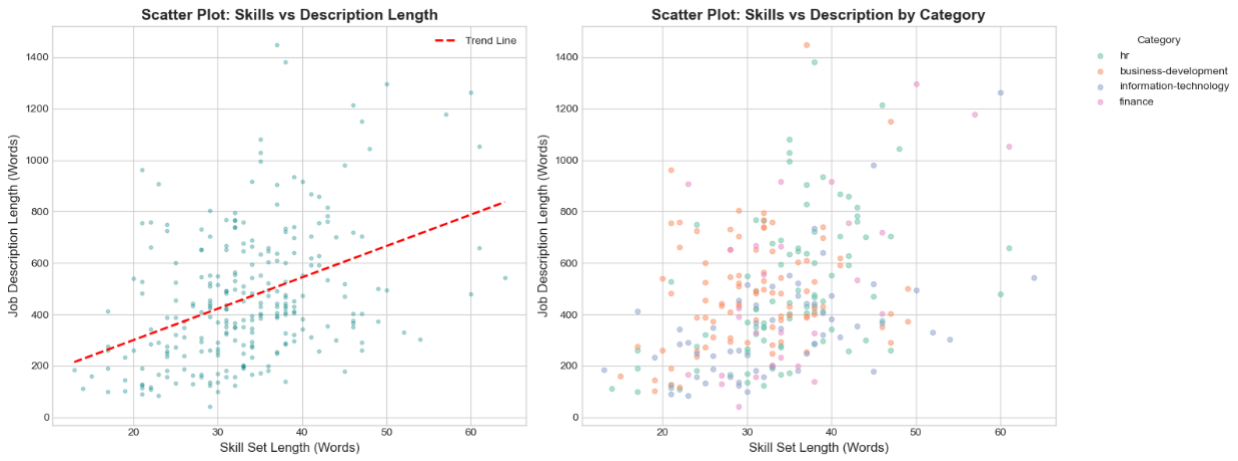


Figure 5: Scatterplot

Purpose: Investigates connections among characteristics and possible grouping in terms of categories.

The scatter plots demonstrate the correlation between the length of the skill set and that of job description:

- **Left plot:** There is a weak positive correlation between jobs that demand more skills and length of description. The line of trend supports this relationship.
- **Right plot:** There is some clustering of color-coded by category. As an example, IT jobs (green) are likely to be concentrated in areas that have moderate and high levels of skills, whereas other groups are not.

Those visualizations affirm that the length of the text is somewhat discriminative, but the real content (acquired through TF-IDF) is necessary to perform the classification correctly.

8. PROPOSED ML ALGORITHMS

8.1. Algorithm Selection & Justification

In this comparative analysis, three different supervised learning algorithms were chosen to represent various mathematical methods, Probabilistic (Naive Bayes), Linear/Discriminative (Logistic Regression) and Ensemble (Random Forest). This variety guarantees the thorough analysis of the processing of high-dimensional text data by various logic structures.

Table 1: Selected Algorithm Table

Algorithm	**Category**	**Reason for Selection in this Study**
Multinomial Naïve Bayes	Probabilistic	The NLP Baseline: Particularly created to work with discrete data (such as the number of words). It is computationally effective and highly efficient on high dimensional, sparse data (TF-IDF) despite its assumption of independence of features.
Logistic Regression	Linear Model	The Discriminative Baseline: selected due to its simplicity and explainability. It forms a linear decision boundary that is surprisingly accurate on text classification and gives well-calibrated probabilities on our score of Match Confidence.

Random Forest	Ensemble	Non-Linear Capability: In contrast to the previous two, Random Forest can learn non-linear, complicated associations among combinations of skills and job titles. It minimizes overfitting by use of the Bagging technique and the noise in job descriptions is also addressed.

8.2. Selection Strategy Overview

These algorithms were selected based on the Occam Razors principle of machine learning model selection:

- **Start Simple (Naive Bayes):** To determine a minimum level of performance.

- **Add Complexity (Logistic Regression):** To compare the performance of a discriminative (learns the boundary between classes) approach to a generative approach.

- **Maximize Power (Random Forest):** To develop the strength of multiple decision trees to respond to the complexity of the 20+ job categories and the hierarchy of the data.

9. METHODOLOGY

The creation of the AI Career Assistant was a multi-phase workflow that involved a logical and careful processing of raw unstructured text to convert it into a powerful hierarchical classification system. This process is as follows:

9.1. Importing Dataset

The processing of the dataset started with the absorption of the dataset with the Pandas library. The data was loaded out of an Apache Parquet file (train-00000-of-00001.parquet) which is a columnar storage format that is optimized to provide efficient data retrieval. The first data comprised unstructured job title, skill sets and job description.

9.2. Data Cleaning & Preprocessing

This involves eliminating data errors and converting data to its final form, a data set. A strict cleaning pipeline was used to guarantee the stability of the model and the quality of data:

- Null Removal: Listwise deletion (dropna) was used to remove rows with missing values in the important columns (job_title, job_skill_set, category).
- Class Balancing (Thresholding): An analysis of the target variable showed that there was a lot of class imbalance. A frequency threshold of 10 was used to limit noise and guarantee that there is enough training data in each of the classes. Retention was done only to job titles that came up 10 or above.
- Normalization of text: A custom clean text function was used on the text data. This was done by turning text to lower case, eliminating special characters through Regular Expressions (re.sub) and standardizing whitespace.

9.3. Exploratory Data Analysis

Matplotlib and Seaborn were used to analyse the structure of the dataset before modelling:

- **Class Distribution:** The most common categories and job titles were plotted as bar charts and the preponderance towards technical jobs was confirmed.
- **Text Length Analysis:** Histograms and Boxplots were created to identify the outliers in the description lengths, to be able to ensure that the selected vectorizer would be able to process the difference in the size of the text.
- **Correlation:** The correlation between the length of description and skill density was checked with the help of a heatmap.

9.4. Feature Selection & Encoding

To transform the text data into a numeric form that can be read by the machine, the concept of Bag of N-Grams was used:

- **Feature Merging:** The purged skills and job descriptions were merged in one column (combined_features). This gave the model signals (skills) that are rich in key words as well as semantic context (descriptions).
- **TF-IDF Vectorization:** Tfidfvectorizer was used to process the text.
- **Max Features:** Only goes to top 2,000 most important terms.
- **N-Grams:** 1-2 range was chosen, which will enable the model to include single words (python) and complex phrases (Data Science).

9.5. Train-Test Split

The data in a form of vectors was divided into training and test sets in an 80/20 proportion.

Stratification: It is important to note that the division was stratified on job_title. This provided that the allocation of job positions which was similar during training and testing to avoid that unusual job titles (those close to the mark of 10) would be practically removed out of the test sample.

9.6. Train Algorithms

There was implementation of a Hierarchical Classification strategy. Three different algorithms, namely Naive Bayes (MultinomialNB), Logistic Regression and Random Forest, were trained individually on two different targets:

- **Level 1 (Industry):** The models were trained to make the predictive decision in the broad category (e.g., "IT" or Sales).
- **Level 2 (Role):** The models were trained to make predictions based on the job_title (e.g., Python Developer).

9.7. Evaluate Using Metrics

A custom get_metrics function was used to evaluate the performance of the models. This function calculated:

- **Precision:** Generality of correctness.
- **Weighted Precision, Recall, and F1-Score:** These weighted values were necessary to consider the imbalance between the classes and make sure that the high performance of the model was not due to over-fitting the majority class.

9.8. Conclusion & Deployment

Lastly, the most successful model (Random Forest) was incorporated into a graphical Command Line Interface (CLI).

Inference Pipeline: It takes a raw skill string entered by a user, cleans and TF-IDF process the results, and makes queries to the trained Random Forests models.

Output: A predicted Industry and Job Role are provided, as well as a Confidence Score (predict_proba), which the user can act upon and get a clear career suggestion.

10. EVALUATION METRICS

The tables below provide a summary of the performance of the three machine learning algorithms at the two levels. The reported metrics include Accuracy, Weighted Precision, Weighted Recall and Weighted F1-Score.

10.1. Industry Category Prediction

This table shows how well each model identifies the broad industry (e.g., IT, Sales).

Table 2: Category metrics evaluation

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	0.904	0.914	0.904	0.901
Logistic Regression	0.923	0.931	0.923	0.922
Random Forest	0.942	0.951	0.942	0.943

Observation: Random Forest is the most accurate at the Category level indicating that it is strong in making a distinction between general industry subjects.

10.2. Specific Job Title Prediction

This table shows the performance when predicting specific roles (e.g., Python Developer, Sales Manager).

Table 3: Job Title metrics evaluation

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	0.423	0.391	0.423	0.309
Logistic Regression	0.577	0.547	0.577	0.509
Random Forest	0.654	0.599	0.654	0.595

Observation: The accuracy decreases significantly between Level 1 and Level 2 in all the models. This is not surprising because the finer granularity and higher overlap in the classes of jobs are expected.

11. PSEUDOCODE

11.1. Main System Pseudocode

START

IMPORT pandas, numpy, sklearn libraries

LOAD "Job Skill Set" dataset into DataFrame

DEFINE Function preprocess_text(text):

CONVERT text to lowercase

REMOVE punctuation and special characters

REMOVE stop words (e.g., "the", "and")

RETURN cleaned_text

APPLY preprocess_text to "Skills" column

INITIALIZE TfidfVectorizer

FIT_TRANSFORM cleaned_skills into Feature_Vectors (X)

SET Job_Titles as Target_Labels (y)

SPLIT data into Training_Set (80%) and Testing_Set (20%)

FOR EACH Model IN [Naive_Bayes, Logistic_Regression, Random_Forest]:

INITIALIZE Model

TRAIN Model using Training_Set

PREDICT Labels using Testing_Set

CALCULATE Accuracy, Precision, Recall

PRINT Performance Report

ENDFOR

END

11.2. Naive Bayes Pseudocode

START NaiveBayes_Classification

IMPORT MultinomialNB

SET vectorizer = CountVectorizer(binary=True)

X_train = vectorizer.fit_transform(train_texts)

X_test = vectorizer.transform(test_texts)

model = MultinomialNB

model.fit(X_train, y_train_category)

y_pred = model.predict(X_test)

probabilities = model.predict_proba(X_test)

OUTPUT accuracy, precision, recall, f1

SAVE model & vectorizer

END

11.3. Logistic Regression Pseudocode

START LogisticRegression_Classification

```
vectorizer = TfidfVectorizer(ngram_range=(1,2), max_features=2000)
X_train = vectorizer.fit_transform(train_texts)
X_test  = vectorizer.transform(test_texts)

base = LogisticRegression(max_iter=1000)
model = OneVsRestClassifier(base)

model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)

evaluate metrics
```

END

11.4. Random Forest Pseudocode

START RandomForest_Classification

```
vectorizer = TfidfVectorizer(max_features=2000)
X_train = vectorizer.fit_transform(train_texts)

model = RandomForestClassifier(n_estimators=200, random_state=42)

IF task = multi_class:
    model.fit(X_train, y_train_category)
ELSE IF task = multi_label:
    wrapper = MultiOutputClassifier(model, n_jobs=-1)
    wrapper.fit(X_train, Y_train_skills)

y_pred = model.predict(X_test)
evaluate metrics

END
```

12. FLOWCHARTS

12.1. Main System Flowchart

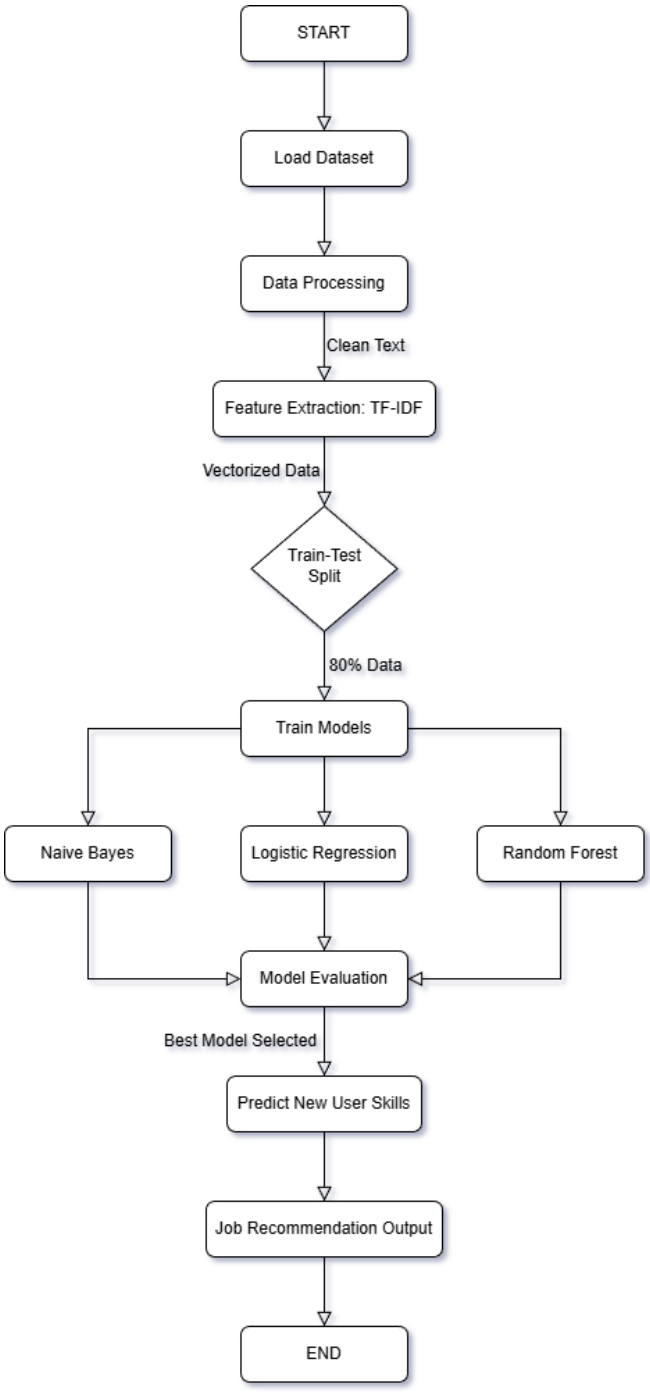


Figure 6: Main System Flowchart

12.2. Naive Bayes Flowchart

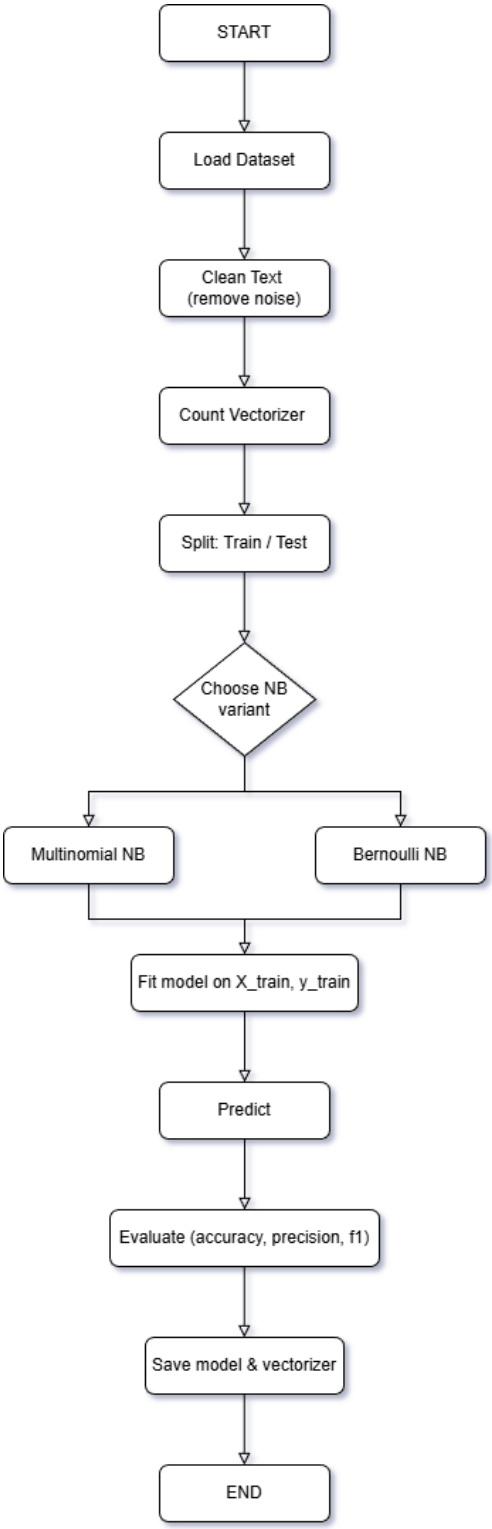


Figure 7: Naive Bayes Flowchart

12.3. Logistic Regression Flowchart

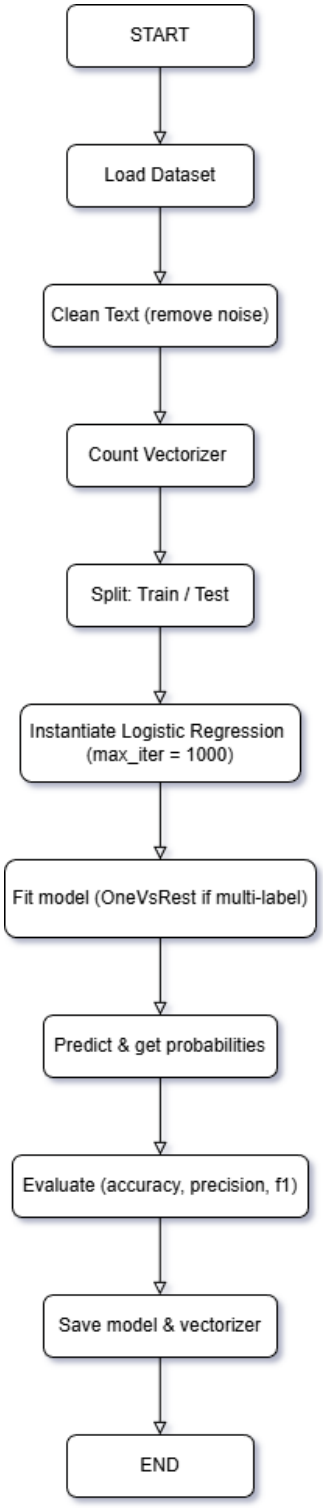


Figure 8: Logistic Regression Flowchart

12.4. Random Forest Flowchart

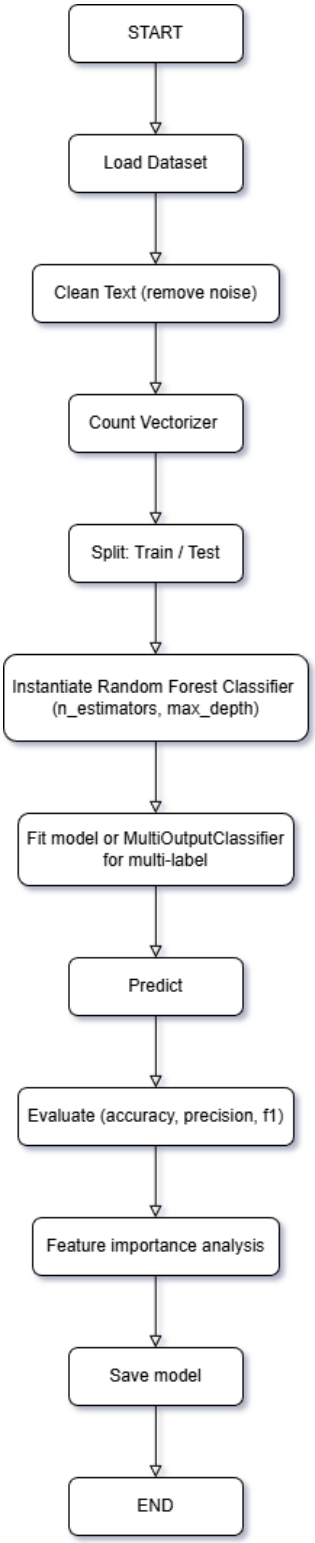


Figure 9: Random Forest Flowchart

13. TOOLS AND TECHNOLOGIES USED

To create the Job Recommendation System, Python-based technology stack was chosen because it has a wide range of data science and machine learning libraries. The following tools and libraries were used:

Table 4: Tools & Technologies Used

TOOL / LIBRARY	ROLE IN PROJECT	JUSTIFICATION FOR SELECTION
Python	Core programming language	The standard in the industry in terms of AI because it is easy to read and has an enormous collection of ready-to-use data science packages.
Pandas	Data Manipulation	Used to load the dataset into a structured Data-Frame. It enabled the processing of data with efficient data cleaning (dropping of nulls, string operations).
Scikit-Learn (sklearn)	Machine Learning Framework	The library was used for: 1. Vectorization (TF-IDF): Text to numbers.

		2. Modelling: applying Naive Bayes, Logistic Regression and Random Forest. 3. Assessment: Accuracy measure and division of training data.
Hugging Face (dataset)	Data source	Facilitated the access to the dataset of the "Job Skill Set".
NumPy	Numerical Operations	Utilized for calculating the maximum probability scores from the model's prediction output to determine match confidence.
Regex (re)	Text Normalization	Utilized to generate a cleaning operation that removes punctuations and non-alphanumeric characters, which provide the AI with high-quality input.

14. CONCLUSION

A viable job recommendation pipeline using the batuhanmtl/job-skill-set dataset was effectively created and tested in this project. The project was transformed to a simple keyword-matching system to a context-sensitive machine learning system. The frequency threshold of 10 was used to remove noise caused by singleton classes and this gave a more consistent training environment.

The suggested solution will solve the key issues of the contemporary recruitment environment by creating actionable hints to align the candidates with the possible positions. The application uses the probability output of the model (predict_proba) to give a score of the Match Confidence, which can be ranked by top-K and saves time that would otherwise be spent on screening resumes manually. Moreover, the tool provides a more comprehensive skill-to-job mapping by combining job titles and descriptions, and this is more holistic and considers the context in which a skill is applied, hence, making it possible to make better hiring and career development decisions.

Future work should be aimed at improving robustness and semantic depth. Although the existing threshold-based method achieved much better accuracy, the next round can be based on hyperparameter optimization with stratified cross-validation to make sure that the 65.38% accuracy is stable on all folds of data. Also, the incorporation of Deep Learning embeddings (e.g., BERT or Word2Vec) might enhance the level of semantic comprehension, particularly in the case of low-confidence inputs, such as the one in the case of the salary survey, which was noted during testing. Lastly, the system must be audited in terms of fairness and A/B tested to the user to guarantee that any offline metric gains are reflected in the production environment as unbiased and effective career recommendations.

15. BIBLIOGRAPHY

Alam, M., 2024. *Data Science Dojo.* [Online] Available at: https://datasciencedojo.com/blog/random-forest-algorithm/ [Accessed 16 December 2025].

Anon., 2022. *machine learning plus.* [Online] Available at: https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/ [Accessed 16 December 2025].

Batuhanmtl, 2024. *Hugging Face.* [Online] Available at: https://huggingface.co/datasets/batuhanmtl/job-skill-set/viewer/default/train?views%5B%5D=train [Accessed 14 December 2025].

Chandak, 2024. *Journal of emerging technologies and innovative research.* [Online] Available at: https://www.jetir.org/papers/JETIR2404C74.pdf [Accessed 19 January 2026].

Humaidi, 2023. *E3S Web of Conferences.* [Online] Available at: https://www.e3s-conferences.org/ [Accessed 19 January 2026].

IEEE, 2024. *IEEE Explore.* [Online] Available at: https://ieeexplore.ieee.org/document/10714763 [Accessed 16 December 2025].

Jain, 2024. *Research Gate.* [Online] Available at: https://www.researchgate.net/publication/363891251_Recommendation_System_using_Machine_Learning_Techniques [Accessed 19 January 2026].

Kabir, 2025. *Arxiv.* [Online] Available at: https://arxiv.org/abs/2111.00215 [Accessed 19 January 2026].

Lee, F., n.d. *IBM.* [Online] Available at: https://www.ibm.com/think/topics/logistic-regression [Accessed 16 December 2025].

Martinez, J., 2018. *Research Gate.* [Online] Available at: https://www.researchgate.net/publication/324652918_Recommendation_of_Job_Offers_Using_Random_Forests_and_Support_Vector_Machines [Accessed 16 December 2025].

Sahu, 2023. *INTERNATIONAL RESEARCH JOURNAL OF MODERNIZATION IN ENGINEERING TECHNOLOGY AND SCIENCE.* [Online] Available at: https://www.irjmets.com/uploadedfiles/paper/volume5/issue_5_may_2023/39902/final/fin_irjmets1684824844.pdf [Accessed 19 January 2026].

Zhang, M., n.d. *Tsinghua.edu.cn.* [Online] Available at: https://keg.cs.tsinghua.edu.cn/jietang/publications/SIGIR14-Zhang-et-al-cold-start-recommendation.pdf [Accessed 16 December 2025].