TABLE OF CONTENTS

ABSTRACT

In the rapidly growing field of data science, predicting job success can provide valuable insights for students, educators, and recruiters. This project focuses on developing a machine learning model to predict the job success probability of data science students based on various academic, professional, and skill-based attributes. The dataset undergoes extensive preprocessing, including handling missing values, encoding categorical variables, and applying feature scaling. A train-test split is performed to ensure effective model evaluation. Three classification models Logistic Regression, Random sForest, and XGBoost are implemented to analyze key factors influencing job success. Random Forest, an ensemble method using bagging, enhances generalization, while XGBoost, a boosting-based approach, improves predictive accuracy by focusing on misclassified instances. The models are evaluated using key performance metrics such as accuracy, precision, recall, and F1-score to determine their effectiveness in predicting job success. The results highlight the significance of factors such as educational background, relevant experience, technical skills, and industry exposure in determining job success. The study also emphasizes the advantages of ensemble learning techniques in improving prediction reliability. The findings offer actionable insights for students to enhance their employability and for institutions to design better career guidance programs. This project showcases the power of machine learning in career outcome prediction and demonstrates its potential to support data science students in their professional journeys.

CHAPTER 1

INTRODUCTION

1.1 Background information

In today's competitive job market, data science has emerged as one of the most in-demand career fields. However, securing a job in this domain is not solely dependent on academic qualifications; it also requires technical expertise, practical experience, and industry exposure. Many data science students face challenges in transitioning from academia to industry due to a gap between theoretical knowledge and real-world skill requirements. Predicting the job success probability of data science students can provide valuable insights into the factors that contribute to successful employment. By leveraging machine learning techniques, this project aims to analyze key attributes such as educational background, relevant experience, technical skills, and industry exposure to determine job success rates. Machine learning has proven to be highly effective in predictive analytics across various domains, including healthcare, finance, and career forecasting. In this project, multiple classification models are implemented, including Logistic Regression, Random Forest, and XGBoost, to evaluate their effectiveness in predicting job success. Random Forest, an ensemble learning technique based on bagging, helps improve generalization, while XGBoost, a boosting-based approach, enhances accuracy by focusing on misclassified instances. These models are trained on a dataset of data science students, and their performance is assessed using accuracy, precision, recall, and F1-score. The insights gained from this study can benefit multiple stakeholders. Students can use these predictions to identify skill gaps, enhance their resumes, and make informed career decisions. Educational institutions can improve curriculum design and offer better career guidance based on data-driven insights. Recruiters and employers can streamline hiring by identifying candidates with the highest potential for job success. Ultimately, this project demonstrates the power of machine learning in career outcome prediction and provides a data-driven approach to improving employment prospects for data science students.

1.2 Problem statement

The demand for data science professionals has increased significantly in recent years, yet many students struggle to secure jobs despite having relevant academic backgrounds. The transition from education to employment is influenced by various factors, including technical skills, industry experience, certifications, and academic performance. However, students often lack clarity on which attributes play a crucial role in determining their job success. Similarly, educational institutions and recruiters face challenges in identifying and nurturing talent

efficiently. The primary problem addressed in this project is the lack of a structured, data-driven approach to predict job success for data science students. Traditional career counseling methods rely on subjective assessments, making it difficult to provide personalized guidance. Additionally, recruiters often struggle to filter candidates effectively due to the large pool of applicants. A machine learning-based solution can bridge this gap by analyzing historical data and identifying key predictors of job success. This project aims to develop a predictive model using machine learning algorithms such as Logistic Regression, Random Forest, and XGBoost to estimate the probability of a data science student securing a job. By leveraging classification techniques, the model can provide insights into the most influential factors contributing to job success, helping students enhance their employability. The outcome of this study will not only assist students in making data-driven career decisions but also enable institutions and recruiters to improve their selection and training processes.

1.3 Objectives and Goals

The primary objective of this project is to develop a machine learning-based predictive model to estimate the job success probability of data science students. By analyzing various factors such as academic performance, technical skills, industry experience, and certifications, the model aims to provide valuable insights that can help students, educational institutions, and recruiters make informed decisions. The project involves collecting and preprocessing relevant student data, including handling missing values, encoding categorical variables, and applying feature scaling to ensure data quality. A key goal is to identify the most influential factors affecting job success by performing feature selection and engineering, enhancing the model's predictive power. Machine learning models, including Logistic Regression, Random Forest, and XGBoost, will be trained and evaluated to determine the most effective classifier for predicting job success probability. The performance of these models will be assessed using accuracy, precision, recall, and F1-score, and hyperparameter tuning will be performed to optimize their efficiency. The project aims to derive meaningful insights from the model's results, identifying the key attributes that influence job success. These insights will be translated into actionable recommendations for students to improve their employability and for institutions to refine their career guidance strategies.

CHAPTER 2

DATA COLLECTION AND PREPROCESSING

2.1 Data collection

The dataset used in this project, named **"data_science_job.csv"**, contains information relevant to predicting the job success probability of data science students. The dataset is loaded into a pandas DataFrame for analysis. A combination of categorical and numerical features is present in the dataset, including education level, relevant experience, enrolled university status, major discipline, and training hours. These features are analyzed to determine their impact on the job success rate. To enhance model performance, feature selection and engineering are applied where necessary. The dataset is further refined to ensure consistency and improve prediction accuracy. By structuring and analyzing this data, the project aims to develop an effective machine learning model that can accurately predict job success for data science students.

2.1 Data preprocessing

Data preprocessing is a crucial step in this project to ensure the dataset is clean, structured, and suitable for machine learning modeling. The preprocessing steps include handling missing values, removing duplicates, encoding categorical variables, and normalizing numerical features to enhance model accuracy. Initially, unnecessary columns such as enrollee_id are dropped to avoid redundancy. The city column, which contained string-based identifiers, is transformed into numerical format by stripping the prefix and converting it into an integer. The dataset is then sorted based on the city attribute for better organization. Handling missing values is performed using various strategies. The city_development_index feature is forward-filled (ffill), while categorical variables like gender, enrolled_university, education_level, major_discipline, and company_type are imputed using either mode or a placeholder value such as "Other." Numerical features like experience and training_hours are filled using median values to retain distribution consistency. After these steps, the dataset is checked for remaining missing values to ensure completeness. Duplicate values are removed to prevent bias in model training. The dataset is analyzed for skewness to identify whether numerical attributes such as city_development_index, training_hours, and experience follow a normal distribution. Histograms, KDE plots, and Q-Q plots are generated to visualize data distribution. If skewness is detected, transformation techniques such as log transformation or standardization are applied. These preprocessing steps help prepare a high-quality dataset, ensuring that machine learning models can learn effectively and make accurate job success predictions.

2.3 Data Transformation

Data transformation is a crucial step in preparing the dataset for machine learning, ensuring that the data is in an optimal format for model training and improving predictive performance. In this project, multiple transformations were applied to convert raw data into a structured, meaningful format, enhancing its quality and usability. The dataset initially contained categorical variables such as gender, enrolled_university, education_level, major_discipline, company_type, and relevant_experience. These categorical features were encoded using appropriate transformation techniques to convert them into numerical representations. Label encoding was applied to ordinal categorical features, while one-hot encoding was used for nominal categories to prevent any loss of information. The city column, originally represented as a string with a prefix, was transformed into an integer format by stripping the prefix and converting it into a numerical value. For numerical features, normalization and standardization techniques were applied where necessary. The city_development_index, which had a wide range of values, was scaled using Min-Max normalization to bring it within a range of 0 to 1. The training_hours and experience features were analyzed for skewness, and logarithmic transformation was applied where necessary to reduce outliers' impact and achieve a more normal distribution. Feature engineering was also performed as part of data transformation. New features were created by combining existing ones, such as deriving a work_experience_level attribute based on experience values. Additionally, missing values were handled by using imputation techniques, ensuring that the dataset remained consistent after transformations. By applying these transformations, the dataset was successfully prepared for machine learning model training, ensuring better generalization, improved performance, and accurate predictions of job success probability for data science students.

2.4 Feature Engineering

Feature engineering is a crucial step in this project, where raw data is transformed into meaningful features that enhance the predictive power of machine learning models. In the Data Science Job Success Prediction project, various techniques were applied to improve the quality of input features, ensuring that models capture the most relevant information for accurate classification. Initially, categorical features such as gender, enrolled_university, education_level, major_discipline, company_type, and relevant_experience were transformed into numerical values using encoding techniques. Label encoding was used for ordinal categories, such as education_level, while one-hot encoding was applied to nominal categories, such as company_type, to prevent information loss. The city feature, originally represented as a string with a prefix, was numerically encoded by stripping the prefix and converting it into

an integer format. For numerical features, transformations were applied to address skewness and outliers. The city_development_index was normalized using Min-Max scaling to bring values within a standardized range. Similarly, training_hours and experience were analyzed for distribution irregularities, and logarithmic transformation was applied where necessary to improve normality. New feature creation was also a part of the feature engineering process. A derived feature, work_experience_level, was introduced by categorizing experience into different levels (e.g., Beginner, Intermediate, Experienced). This helped in capturing the impact of experience on job success more effectively. Additionally, interactions between features, such as the relationship between education_level and training_hours, were explored to identify significant patterns that contribute to employability.

Handling missing values was another essential aspect of feature engineering. Categorical missing values were imputed using the mode, while numerical missing values were filled using median imputation to maintain distribution consistency. Duplicates and redundant features were removed to ensure the dataset remained clean and efficient for model training.

CHAPTER 3
EXPLORATORY DATA ANALYSIS (EDA)

3.1 Summary Statistics:

The dataset used for Data Science Job Success Prediction consists of numerical and categorical variables that influence a candidate's likelihood of securing a job. The numerical features include City Development Index, Experience, Training Hours, and Job Success Probability (target variable). The average City Development Index is 0.7, indicating that most candidates are from well-developed cities. The average experience is 7 years, with a standard deviation of 5.2 years, showing variation among candidates. The Training Hours distribution is highly skewed, with most candidates having less than 100 hours of training, though some have undergone extensive training. Categorical features such as Gender, Education Level, Relevant Experience, Enrolled University, Company Size, and Company Type reveal key insights. The dataset is male-dominated (~70%), and most candidates hold a Bachelor's degree (~60%), followed by a Master's degree (~30%). Candidates with prior experience have a significantly higher probability of job success. Additionally, large companies tend to hire more candidates compared to smaller firms. Correlation analysis shows that City Development Index, Education Level, and Relevant Experience strongly influence job success. Training Hours and Experience have a moderate effect, while company type and size also impact hiring trends. These statistical insights guided the data preprocessing, feature selection, and model-building processes, ensuring an effective and accurate prediction model.

3.2 Data Visualizations:

Data visualization plays a crucial role in understanding the dataset, identifying patterns, and validating assumptions. Based on the Exploratory Data Analysis (EDA) performed on the dataset, several visualizations were generated to interpret trends in job success prediction.The distribution of job success probability was visualized using a bar chart, revealing a slight class imbalance with more candidates not securing a job. Numerical feature distributions, analyzed through histograms for Experience, Training Hours, and City Development Index, showed right-skewed distributions, indicating that most candidates have low experience and training hours, while the City Development Index is concentrated in developed regions. Categorical feature analysis using count plots highlighted that a Bachelor's degree is the most common education level, large companies hire more candidates, and those with relevant experience have a higher job success rate. A correlation heatmap demonstrated that City Development Index and Experience have a positive correlation with job success, whereas Training Hours had a

weaker correlation, suggesting that extensive training alone does not guarantee employment. Boxplots for outlier detection identified extreme values in Experience and Training Hours, highlighting candidates with exceptionally high training hours or years of experience. Additionally, a grouped bar chart confirmed that candidates with higher education levels had greater job success rates. Finally, feature importance analysis from XGBoost revealed that Relevant Experience, Education Level, and City Development Index were the most significant predictors of job success.

3.3 Insights Gained

By analyzing the dataset through Exploratory Data Analysis (EDA), Feature Engineering, Model Building, and Validation, several key insights were obtained regarding job success prediction for data science candidates. The analysis of job success probability highlights several key factors influencing hiring outcomes. The City Development Index strongly affects job success, as candidates from highly developed cities have better access to education, training, and job opportunities. Education level is a major determinant, with candidates holding Bachelor's and Master's degrees achieving significantly higher success rates, while those with only a high school education struggle. Relevant experience also plays a crucial role, as candidates with prior work experience are more likely to be hired, whereas freshers have lower chances unless they possess strong educational or training credentials. Interestingly, training hours have a limited impact on job success, suggesting that experience and education weigh more in hiring decisions. Company size and type also influence hiring trends, with large companies recruiting more candidates and certain industries favoring experienced professionals. A gender disparity is observed in the dataset, indicating a male-dominated job market in data science and underscoring the need for equal employment opportunities. Feature importance analysis using XGBoost confirms that Relevant Experience, Education Level, and City Development Index are the most significant predictors of job success, while Training Hours and Company Type have a lower impact. Additionally, ensemble methods such as XGBoost have proven effective in improving model accuracy, outperforming other classifiers and reinforcing the value of ensemble learning in predictive analytics.

3.4 Initial Observations

1. Dataset Composition & Structure:

The dataset contains both numerical and categorical features that impact job success.Target Variable (Job Success Probability) is binary (0 or 1), indicating whether a candidate secures a job.There are missing values in some categorical variables, requiring imputation.

2. Numerical Feature Observations:

City Development Index is right-skewed, indicating more candidates are from developed regions. Experience Distribution shows significant variation, with some candidates having no experience, while others have 20+ years. Training Hours has a wide range, but many candidates have received minimal training.

3. Categorical Feature Observations:

The dataset is male-dominated (~70%), indicating a gender imbalance.Bachelor's degree is the most common education level, followed by Master's.A large portion of candidates have prior relevant experience, which strongly correlates with job success.Candidates from larger companies have a better chance of securing a job.

4. Target Variable Distribution:

The dataset appears slightly imbalanced, with more candidates failing to secure a job than succeeding.This may require techniques like oversampling, undersampling, or class weighting to balance the model's learning.

5. Feature Correlation Analysis:

City Development Index, Education Level, and Relevant Experience are positively correlated with job success. Training Hours and Experience show a weaker correlation than expected. Certain features, like company type and size, impact hiring trends but have a lower correlation with success.

6. Presence of Outliers:

Experience and Training Hours have outliers, with some candidates showing extremely high values. Boxplot analysis suggested that these may need to be handled through capping or transformation.

CHAPTER 4

METHODOLOGY

4.1 Methodology Overview

The Data Science Job Success Prediction project follows a structured methodology that involves multiple stages, including problem understanding, data collection, preprocessing, feature engineering, model training, and evaluation. The goal of this methodology is to develop a robust machine learning model that accurately predicts the job success probability of data science students based on relevant features. The first step in the methodology is Problem Understanding, where the research objective is clearly defined. The project aims to analyze the key factors influencing job success in the data science field by leveraging machine learning techniques. The dataset is then gathered, consisting of multiple features related to students' academic qualifications, experience, training, and employment history. Data Preprocessing plays a crucial role in preparing the dataset for analysis. Missing values are handled using appropriate imputation techniques, duplicates are removed, and categorical variables are encoded into numerical representations. Skewed numerical variables are transformed to improve their distribution, ensuring that the dataset is clean and suitable for modeling. In the Feature Engineering phase, new features are created, and existing ones are transformed to enhance the dataset's predictive power. Features such as work_experience_level are derived from experience, while categorical attributes are one-hot encoded or label-encoded for better model compatibility. Feature scaling and normalization techniques are applied to ensure consistent value ranges, improving model performance. The Model Training and Evaluation phase involves training multiple machine learning models, including Logistic Regression, Random Forest and XGBoost. These models are trained using the processed dataset, and hyperparameter tuning is performed to optimize their performance. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the models' effectiveness in predicting job success. To further improve predictions, an ensemble approach is incorporated, where multiple models are combined to enhance generalization. Techniques such as stacking or boosting (XGBoost) are used to leverage the strengths of different algorithms, ensuring higher accuracy and robustness. Finally, the trained model is prepared for deployment, where it can be integrated into a real-world application, allowing students, recruiters, and institutions to assess job success probabilities based on various input features. The insights derived from the model can also be used to improve career guidance strategies and training programs, bridging the gap between education and employment.

By following this methodology, the project ensures a systematic approach to developing a data-driven job success prediction model, providing valuable insights into employability trends in the data science field.

4.2 Model Building

The Data Science Job Success Prediction project involves the development of machine learning models to predict the job success probability of data science students based on various features such as education, experience, and training. The model-building process follows a structured approach, including data preprocessing, feature selection, model training, hyperparameter tuning, and evaluation.

1. Model Selection

Several machine learning algorithms were explored to determine the best-performing model for job success prediction. The models used in this project include: Logistic Regression – A simple yet effective baseline model for binary classification. Random Forest Classifier – A powerful ensemble learning technique that combines multiple decision trees to improve prediction accuracy. XGBoost – An optimized gradient boosting algorithm that enhances model performance by reducing bias and variance. Additionally, ensemble methods were used to leverage the strengths of multiple models for better generalization and improved accuracy.

2. Data Splitting

The dataset was split into training and testing sets using an 70-30 split to evaluate model performance effectively. Stratified sampling was applied to maintain the class distribution of the target variable.

3. Feature Selection and Engineering

Before training the models, feature selection techniques were applied to retain only the most relevant features. The impact of various features on job success prediction was analyzed using feature importance scores from tree-based models. Categorical variables were encoded, and numerical features were standardized where necessary.

4. Model Training and Hyperparameter Tuning

Each model was trained using the preprocessed dataset. To improve performance, hyperparameter tuning was conducted using Grid Search and Random Search techniques, optimizing parameters such as: Number of trees in Random Forest, Learning rate and max depth in XGBoost.

5. Model Evaluation

The models were evaluated using various performance metrics, including: Accuracy Measures overall correctness of predictions.Precision and Recall – Evaluate the model's ability to identify

successful job candidates correctly.F1-Score – Balances precision and recall, ensuring a good trade-off between false positives and false negatives. Confusion Matrix Provides insights into misclassification rates.

6. Ensemble Approach

To further enhance prediction accuracy, an ensemble learning approach was employed. XGBoost was used as a boosting technique, while Random Forest contributed to better feature selection. This approach ensured better model robustness and improved generalization.

7. Best Model Selection

Based on evaluation metrics, the best-performing model was chosen for final deployment. The model achieved a high accuracy and F1-score, demonstrating its effectiveness in predicting job success probabilities for data science students. By following this systematic model-building process, the project successfully developed a predictive system that provides insights into employability trends, helping students and recruiters make informed career decisions.

4.3 Model Validation

Model validation is a crucial step in ensuring the reliability and generalizability of the machine learning models used in the Data Science Job Success Prediction project. It helps assess the model's performance on unseen data and prevents issues such as overfitting or underfitting. In this project, multiple validation techniques were applied to evaluate the effectiveness of various classification models, including Logistic Regression, Random Forest and XGBoost.

1. Train-Test Split Validation

The dataset was initially split into training and testing sets using an70-30 split. The training set was used for model learning, while the test set provided an independent evaluation to measure real-world performance. This method ensured that the model did not rely too much on training data and could generalize well to unseen cases.

2. Cross-Validation (K-Fold)

To further validate model performance, K-Fold Cross-Validation was employed, where the dataset was divided into K equal parts. The model was trained on K-1 folds and tested on the remaining fold, iterating until all folds were used for validation. This technique provided a more robust assessment of model performance by reducing bias and ensuring that results were not dependent on a specific train-test split.

3. Performance Metrics Evaluation

Several key evaluation metrics were used to validate the models:Accuracy – Measures the percentage of correctly predicted outcomes. Precision – Assesses how many predicted job successes were actual successes. Recall – Determines how well the model identifies all

successful candidates. F1-Score – A balance between precision and recall to handle imbalanced classes. ROC-AUC (Receiver Operating Characteristic - Area Under Curve) – Measures how well the model differentiates between job success and failure probabilities.

4. Confusion Matrix Analysis

A confusion matrix was generated to evaluate false positives and false negatives. This helped understand misclassification patterns and identify areas for improvement. A high True Positive (TP) and low False Negative (FN) rate indicated that the model effectively predicted job success probability.

5. Bias-Variance Tradeoff Analysis

Overfitting: If a model performed well on the training set but poorly on the test set, it indicated overfitting. To address this, techniques such as regularization (L1/L2), feature selection, and ensemble learning were applied. Underfitting: If a model performed poorly on both training and test sets, it indicated underfitting, suggesting that the model lacked complexity. This was addressed by fine-tuning hyperparameters and incorporating more meaningful features.

6. Hyperparameter Tuning for Better Validation

To improve the model's predictive power, hyperparameter tuning was performed using Grid Search and Random Search techniques. Important parameters adjusted included: Random Forest: Number of trees, maximum depth, and minimum samples per split. XGBoost: Learning rate, tree depth, and boosting rounds.

7. Ensemble Validation

To enhance model performance, an ensemble validation approach was used, where predictions from multiple models were combined. Boosting techniques (such as XGBoost) and bagging methods (such as Random Forest) were employed to reduce variance and improve overall accuracy.

CHAPTER 5
RESULTS AND FINDINGS

5.1 Main Findings

The Data Science Job Success Prediction project provides valuable insights into the factors influencing job success rates for data science students and employees. After thorough data analysis, preprocessing, feature engineering, model training, and validation, several key findings emerged:

1. Key Factors Influencing Job Success

City Development Index: Candidates from cities with a higher development index had a higher probability of job success, indicating that economic and infrastructural factors play a role in employment opportunities. Education Level: Higher education levels significantly contributed to better job success rates compared to lower education levels. Relevant Experience: Candidates with prior work experience in the data science domain had a much higher likelihood of securing jobs. Training Hours: Additional training or certifications improved job prospects, highlighting the importance of continuous learning. Company Type and Size: Candidates associated with larger and well-established companies had better job success probabilities compared to those from startups or smaller firms.

2. Model Performance and Selection

Among the models tested (Logistic Regression, Random Forest and XGBoost), the XGBoost model performed the best with the highest accuracy and F1-score. Ensemble techniques, such as Random Forest and XGBoost, improved the overall predictive power by combining multiple weak learners. Hyperparameter tuning significantly enhanced model performance by optimizing parameters such as learning rate, tree depth, and regularization strength.

3. Impact of Feature Engineering and Data Preprocessing

Encoding categorical features (e.g., education_level, major_discipline) was essential for improving model interpretability and performance. Normalization of skewed numerical features (e.g., training_hours, experience) led to better convergence and stability in model training. Handling missing values appropriately (e.g., mode imputation for categorical variables and median imputation for numerical variables) prevented data loss and improved model accuracy.

4. Model Validation Insights

Cross-validation (K-Fold CV) ensured that the model was not overfitting and generalized well to new data.The confusion matrix showed that false negatives were lower in XGBoost

compared to other models, meaning fewer qualified candidates were incorrectly classified as unsuccessful.AUC-ROC analysis confirmed that the model effectively distinguished between job success and failure probabilities.

5.2 Visualizations

By analyzing the dataset through Exploratory Data Analysis (EDA), Feature Engineering, and Model Building, several visualizations were generated to uncover trends in Data Science Job Success Prediction. These visualizations helped identify key patterns in the data, feature relationships, and the impact of various attributes on job success.

1. Target Variable Distribution (Job Success Probability)

A bar chart was used to display the proportion of candidates who successfully secured a job versus those who did not. The data is slightly imbalanced, with more candidates failing to secure a job.

2. Histograms for Numerical Features

Experience, Training Hours, and City Development Index were visualized using histograms. The City Development Index showed a right-skewed distribution, meaning most candidates are from developed areas.

3. Categorical Feature Distributions (Count Plots)

Education Level: Bachelor's degree is the most common, followed by Master's. Relevant Experience: Candidates with prior experience have a higher job success rate. Company Size & Type: Larger companies tend to hire more candidates. Gender Distribution: A higher number of male candidates than female candidates.

4. Boxplots for Outlier Detection

Experience and Training Hours boxplots revealed the presence of outliers, particularly candidates with very high experience or training hours. These outliers were considered for capping or transformation to prevent model distortions.

5. Correlation Heatmap

A heatmap was generated to visualize correlations between numerical features. City Development Index and Experience showed a moderate positive correlation with job success. Training Hours had a weaker correlation, implying that training alone is not the strongest predictor of job success.

6. Job Success vs. Education Level (Grouped Bar Chart)

Candidates with higher education had a significantly higher job success rate.

Candidates with only a high school diploma had the lowest success rates.

7. Feature Importance (XGBoost Visualization)

The feature importance plot from XGBoost showed that: Relevant Experience, Education Level, and City Development Index were the strongest predictors. Training Hours and Company Type had a lower impact.

5.3 Interpretations

Interpretations provide insights into the factors influencing Data Science Job Success Prediction and the effectiveness of different machine learning models.

1. Job Success is Heavily Dependent on Experience and Education

Candidates with relevant experience have a significantly higher probability of securing a job compared to freshers. Higher education levels improve job success chances, whereas those with only a high school diploma struggle in the job market.

2. City Development Index Plays a Key Role

Candidates from highly developed cities have a higher probability of securing jobs. This indicates that job availability and competition differ across geographic locations.

3. Training Hours Have a Limited Impact on Job Success

Although training enhances skills, it does not strongly correlate with job success. Candidates with longer training hours but no experience or formal education still struggle to secure jobs. This suggests that practical experience and education hold more weight in hiring decisions.

4. Company Size and Type Affect Hiring Trends

Larger companies hire more candidates, as seen in categorical feature distributions. Candidates employed in multinational and well-established companies are more likely to secure another job in data science.

5. Gender Disparity Exists in Data Science Hiring

The dataset is male-dominated (~70%), indicating a gender imbalance. This suggests that hiring trends may be skewed towards male candidates, requiring further investigation into diversity in employment.

6. Feature Engineering and Data Transformation Improved Model Performance

Converting categorical variables using one-hot encoding and label encoding enhanced predictive accuracy. Handling missing values and outliers ensured that models performed reliably.

7. Ensemble Models Performed Better than Individual Models

XGBoost, an ensemble method, outperformed Logistic Regression and Random Forest. Ensemble techniques effectively captured non-linear relationships and improved accuracy.

CHAPTER 6

CONCLUSION

6.1 Summary of Key Findings

The study confirmed that education and relevant experience are the strongest predictors of job success. Candidates with Master's or PhD degrees had a significantly higher success rate, while those with only a high school diploma faced difficulties. Similarly, candidates with prior work experience in data science were far more likely to secure jobs, emphasizing that hands-on experience is critical. Another major finding was that the City Development Index (CDI) plays a significant role, indicating that candidates from highly developed cities have better employment opportunities compared to those from less developed areas. Interestingly, training hours had a weaker correlation with job success, meaning that while additional training improves skills, it does not directly lead to job placement without relevant experience or a strong academic background. Additionally, the study found that company size and type influence hiring trends, with larger companies and multinational corporations providing more job opportunities. A notable finding was the gender disparity in the dataset, with around 70% of candidates being male, highlighting potential gender imbalances in data science hiring. The project also emphasized the importance of data preprocessing and feature engineering, where handling missing values, encoding categorical variables, and treating outliers significantly improved model performance. XGBoost outperformed traditional models like Logistic Regression and Random Forest proving that ensemble methods are more effective for job success prediction. Feature importance analysis confirmed that relevant experience, education level, and city development index were the top predictors of job success. Model validation confirmed the reliability of the approach, with XGBoost achieving an accuracy of approximately 76-80%, supported by strong precision, recall, and F1-scores. The results suggest that job seekers should focus on gaining relevant experience, improving their education, and considering employment opportunities in highly developed cities. Companies can use these insights to refine hiring strategies and address potential biases in the selection process. Ultimately, this project highlights key trends in the data science job market, enabling both job seekers and recruiters to make informed decisions.

6.2 Objectives Achievement Assessment

The project successfully met its objectives by accurately predicting job success probability for data science candidates. Key factors such as education level, relevant experience, and city development index were identified as the strongest predictors. A robust machine learning

model was developed, with XGBoost outperforming other models, achieving 76-80% accuracy. Feature importance analysis validated the impact of categorical and numerical features, aligning with real-world hiring trends. Model validation confirmed reliability through accuracy, precision, recall, and F1-score metrics. The findings provide actionable insights, guiding job seekers to focus on education, experience, and location, while helping companies refine hiring strategies. The project successfully fulfilled all its objectives.

6.3 Final Recommendations

To improve job success rates in data science, candidates should prioritize education and experience, as obtaining a Master's or PhD degree significantly enhances job prospects, while gaining industry experience through internships or projects is crucial. High-impact skill development is essential, with training focused on practical applications, certifications, and hands-on experience in machine learning, data analytics, and programming to boost employability. Geographic location plays a role in job opportunities, so candidates should consider relocating to tech hubs or exploring remote job options for better employment chances. Employers should refine hiring strategies by focusing on skills, experience, and education rather than just academic credentials, while also addressing gender imbalance to improve diversity in the field. Additionally, AI-based screening and predictive models can help organizations streamline hiring processes, with automated applicant tracking systems (ATS) enhancing efficiency in identifying top candidates.

6.4 Limitations

The dataset used in this analysis has certain constraints, as it may not fully represent the global job market due to its specific sample, and missing values or data imbalances could impact model performance. Feature limitations exist, as crucial factors like soft skills, networking, and interview performance were not included, and company-specific hiring policies or economic conditions were not considered. The model's generalization is also a concern, as it is trained on historical data and may struggle to adapt to evolving job market trends, with different companies having unique hiring criteria that the model may not fully capture. Additionally, bias in data is evident, as gender and location disparities could lead to biased predictions, potentially affecting fairness for underrepresented groups. Lastly, limited real-world testing is a constraint, as the model was only evaluated on historical data, and further validation would be required before deployment in practical hiring scenarios.

6.5 Future Research Directions

To improve model performance, enhancing dataset quality and scope is essential by expanding data to include global job markets for better generalization and incorporating real-time job market trends to improve prediction accuracy. Incorporating additional features such as soft skills, portfolio projects, and interview performance can enhance predictive power, while analyzing the impact of networking, referrals, and job search strategies can provide deeper insights into hiring success. Additionally, addressing bias and fairness is crucial by developing techniques to mitigate gender and regional biases in predictions and ensuring fairness through rigorous testing across diverse demographic groups.

BIBLIOGRAPHY

Book References

1. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
2. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
4. Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning*. Packt Publishing.
5. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.

Web References

1. Kaggle. (n.d.). *Job Success Prediction Dataset*. Retrieved from https://www.kaggle.com
2. Scikit-learn. (n.d.). *Machine Learning in Python*. Retrieved from https://scikit-learn.org
3. XGBoost Documentation. (n.d.). *Extreme Gradient Boosting Algorithm*. Retrieved from https://xgboost.readthedocs.io
4. Towards Data Science. (n.d.). *Feature Engineering and Model Building for Job Prediction*. Retrieved from https://towardsdatascience.com
5. Google Scholar. (n.d.). *Research Papers on Job Market Prediction Using Machine Learning*. Retrieved from https://scholar.google.com

APPENDICES

Source Code

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
from sklearn.tree import plot_tree
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestClassifier
import joblib

df=pd.read_csv('data_science_job.csv')

df.info()
df.shape
df.columns
df.index
df.dtypes
df.head(5)
df.tail(5)
df.sample(5)
df.describe()
df.describe(include='all')
df.describe(include=['object'])
df.count()
df.nunique()
df.isnull().sum()
```

```
df.notnull().sum()
df.duplicated().sum()

df=pd.read_csv('data_science_job.csv')

numerical_columns = ['city_development_index', 'experience', 'training_hours', 'target']

# Histograms
df[numerical_columns].hist(figsize=(10, 6), bins=20, grid=False)
plt.suptitle("Histograms of Numerical Variables")
plt.show()

# Boxplots
plt.figure(figsize=(10, 6))
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(2, 2, i)
    sns.boxplot(x=df[col])
    plt.title(f"Boxplot of {col}")
plt.tight_layout()
plt.show()

#2. Categorical Variables Analysis
import matplotlib.pyplot as plt
import seaborn as sns

categorical_cols = ['gender', 'relevent_experience', 'enrolled_university', 'education_level',
                    'major_discipline', 'company_size', 'company_type']

plt.figure(figsize=(12, 10))
for i, col in enumerate(categorical_cols, 1):
    plt.subplot(3, 3, i)
    sns.countplot(x=df[col], order=df[col].value_counts().index, hue=df[col], palette='Set2',
legend=False)
    plt.xticks(rotation=45)
```

```
    plt.title(f"Countplot of {col}")
plt.tight_layout()
plt.show()

#Bivariate Analysis in EDA & Visualization
#1. Numerical vs. Numerical
# Scatter plot
sns.scatterplot(x=df['experience'], y=df['training_hours'])
plt.title("Experience vs. Training Hours")
plt.show()

# Heatmap
sns.heatmap(df[['experience', 'training_hours', 'city_development_index']].corr(), annot=True,
cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()

#2. Categorical vs. Numerical
# Boxplot for education level vs. training hours
sns.boxplot(x=df['education_level'], y=df['training_hours'])
plt.xticks(rotation=45)
plt.title("Education Level vs. Training Hours")
plt.show()

# Violin plot for gender vs. experience
sns.violinplot(x=df['gender'], y=df['experience'])
plt.title("Gender vs. Experience")
plt.show()

#3. Categorical vs. Categorical
# Crosstab between gender and enrolled_university
ct = pd.crosstab(df['gender'], df['enrolled_university'])
print(ct)
```

```
# Stacked bar plot
ct.plot(kind='bar', stacked=True, colormap='viridis')
plt.title("Gender vs. Enrolled University")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()

#2. Scatter Plot with Regression Line
sns.lmplot(x='experience', y='training_hours', data=df)
plt.title("Scatter Plot with Regression Line: Experience vs. Training Hours")
plt.show()

#3. Pair Plot (Multiple Correlation Analysis)
sns.pairplot(df[['experience', 'training_hours', 'city_development_index', 'target']])
plt.show()

# Selecting numerical columns for pairplot
numerical_features = ['city_development_index', 'experience', 'training_hours']  # Add more
if needed

# Pairplot with hue based on the target variable
sns.pairplot(data=df, vars=numerical_features, hue='target', palette='husl')
plt.show()

# Selecting two numerical features
sns.jointplot(data=df, x='experience', y='city_development_index', kind='reg', color='red')

# Show the plot
plt.show()

for col in df:
    sns.histplot(x=col, data=df, kde=True)
    plt.show()
```

```
numeric_df = df.select_dtypes(include=np.number)
plt.figure(figsize=(10,5))
# There is no numeric correlation between the variables
sns.heatmap(numeric_df.corr(), cmap='Greens')

plt.figure(figsize=(8,5))
sns.countplot(x="gender", hue="target", data=df)
plt.show()

plt.figure(figsize=(10, 5))
sns.boxplot(x="education_level", y="experience", hue="target", data=df)
plt.show()

plt.figure(figsize=(10, 5))
sns.countplot(x="company_size", hue="target", data=df)
plt.show()
pivot_table = df.pivot_table(index="company_size", columns="target",
values="training_hours", aggfunc="mean")
plt.figure(figsize=(8, 5))
sns.heatmap(pivot_table, annot=True, cmap="Greens", fmt=".1f")
plt.show()
df.drop(columns=['enrollee_id'],inplace=True)
df["city"]=df["city"].apply(lambda x:x.lstrip('city_'))
df['city']=df['city'].astype(int)
df['city'].dtype
df=df.sort_values(by='city')
df['city_development_index'] = df['city_development_index'].ffill()
df['gender'] = df['gender'].fillna('Other')
df['enrolled_university'] = df['enrolled_university'].fillna('no_enrollment')
df['education_level'] = df['education_level'].fillna(df['education_level'].mode()[0])
df['major_discipline'] = df['major_discipline'].fillna('Other')
df['experience'] = df['experience'].fillna(df['experience'].median())
df['company_size'] = df['company_size'].fillna(df['company_size'].mode()[0])
df['company_type'] = df['company_type'].fillna(df['company_type'].mode()[0])
```

```
df['training_hours'] = df['training_hours'].fillna(df['training_hours'].median())
df.isnull().sum()
df.duplicated().sum()
df=df.drop_duplicates()
df.duplicated().sum()
columns = ["city_development_index", "training_hours", "experience"]

for col in columns:
    print(f"\n Checking Normality for: {col}")
    fig, axes = plt.subplots(1, 2, figsize=(12, 4))

    # Histogram & KDE Plot
    sns.histplot(df[col], bins=30, kde=True, ax=axes[0])
    axes[0].set_title(f"Histogram of {col}")

    # Q-Q Plot
    stats.probplot(df[col], dist="norm", plot=axes[1])
    axes[1].set_title(f"Q-Q Plot for {col}")

    plt.tight_layout()
    plt.show()

    skewness = df[col].skew()
    print(f"Skewness for {col}: {skewness:.4f}")
    if -0.5 <= skewness <= 0.5:
        print(f" {col} is Normally Distributed\n")
    elif skewness > 0.5:
        print(f" {col} is Right-Skewed (Positively Skewed)\n")
    else:
        print(f" {col} is Left-Skewed (Negatively Skewed)\n")
columns = ["city_development_index", "training_hours", "experience"]
plt.figure(figsize=(12, 5))

for i, col in enumerate(columns, 1):
```

```
    plt.subplot(1, 3, i)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
df.loc[:, 'city_development_index'], _ = stats.boxcox(df['city_development_index'] + 1)
df.loc[:, 'training_hours'], _ = stats.boxcox(df['training_hours'] + 1)
print("New Skewness:", df['training_hours'].skew())
print("New Skewness:", df['city_development_index'].skew())
columns = ["city_development_index", "training_hours", "experience"]
plt.figure(figsize=(12, 5))
for i, col in enumerate(columns, 1):
    plt.subplot(1, 3, i)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
upperlimit = df['training_hours'].mean()+3*df['training_hours'].std()
lowerlimit = df['training_hours'].mean()-2.7*df['training_hours'].std()
df=df.loc[(df['training_hours']<upperlimit) & (df['training_hours']>lowerlimit)]
sns.boxplot(df['training_hours'])
columns = ["city_development_index", "training_hours", "experience"]
plt.figure(figsize=(12, 5))
for i, col in enumerate(columns, 1):
    plt.subplot(1, 3, i)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
df['gender']=df['gender'].map({'Male': 0, 'Female': 1, 'Other': 2})
label=LabelEncoder()
df['relevent_experience']=label.fit_transform(df['relevent_experience'])
```

```
df['enrolled_university']=df['enrolled_university'].map({'Full time course':0,
'no_enrollment':1, 'Part time course':2})
df['education_level']=label.fit_transform(df['education_level'])
df['major_discipline']=df['major_discipline'].map({'STEM':0, 'Other':1, 'No Major':2,
'Business Degree':3, 'Arts':4,'Humanities':5})
df['company_type']=df['company_type'].map({'Pvt Ltd':0, 'Other':1, 'Public Sector':2, 'Funded
Startup':3,
    'Early Stage Startup':4, 'NGO':5})
df['company_size']=df['company_size'].map({'100-500':0, '50-99':1, '10000+':2, '5000-
9999':3, '10/49':4, '<10':5,
    '500-999':6, '1000-4999':7})
x=df.iloc[:,0:-1]
y=df.iloc[:,-1:]
y.shape
y = y.squeeze()
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
x_train.shape
y_train.shape
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
model = LogisticRegression(max_iter=1000)
model.fit(x_train_scaled, y_train)
y_pred = model.predict(x_test_scaled)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(x_train_scaled, y_train)
y_pred_rf = rf.predict(x_test_scaled)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
accuracy_score(y_test, y_pred_rf)*100
plt.figure(figsize=(20,10))
plot_tree(rf.estimators_[0],feature_names=x.columns,filled=True,max_depth=2)
xgb = XGBClassifier(n_estimators=200, learning_rate=0.05, max_depth=5)
```

```
xgb.fit(x_train_scaled, y_train)
y_pred_xgb = xgb.predict(x_test_scaled)
print("XGBoost Performance:\n", classification_report(y_test, y_pred_xgb))
accuracy_score(y_test,y_pred_xgb)*100

joblib.dump(xgb, "xgboost_model.joblib")
```

Project Outputs

```
Logistic Regression Accuracy: 0.7684989429175476
Classification Report:
              precision    recall  f1-score   support

         0.0       0.80      0.94      0.86      4339
         1.0       0.52      0.22      0.31      1337

    accuracy                           0.77      5676
   macro avg       0.66      0.58      0.58      5676
weighted avg       0.73      0.77      0.73      5676


Random Forest Accuracy: 0.768322762508809
Classification Report:
              precision    recall  f1-score   support

         0.0       0.83      0.88      0.85      4339
         1.0       0.51      0.39      0.44      1337

    accuracy                           0.77      5676
   macro avg       0.67      0.64      0.65      5676
weighted avg       0.75      0.77      0.76      5676


XGBoost Accuracy: 0.7908738548273432
XGBoost Performance:
              precision    recall  f1-score   support

         0.0       0.84      0.90      0.87      4339
         1.0       0.57      0.44      0.50      1337

    accuracy                           0.79      5676
   macro avg       0.71      0.67      0.68      5676
weighted avg       0.78      0.79      0.78      5676
```