WQD7006 MACHINE LEARNING FOR DATA SCIENCE

PROJECT REPORT

SEMESTER 2, SESSION 2023/2024

FRAUDULENT JOB PREDICTION

GROUP 15: RL

SUPERVISOR : DR. RIYAZ AHAMED

NAME	MATRIC NUMBER
LIM SZE SING	22109557
GAN JING WEN	22065433
LAW JIA JIN	22071390
NUR SHAFIQAH BINTI MOHAMAD JOHARI	22119564
NUR NAZIFA BINTI ZHAMRI	22088840

Table Of Contents

CHAPTER 1: INTRODUCTION

1.1 PROBLEM STATEMENT

The rising cost of living in Malaysia has led to an increase in online fraud cases, particularly job recruitment scams. Many Malaysians, seeking additional income opportunities, fall victim to these scams due to difficulty in distinguishing genuine job openings from fraudulent ones. Current literature compares various machine learning models such as logistic regression, decision trees, random forests, and MLP for detecting fake job postings. However, the trade-off between precision and recall suggests that further refinement and enhancement of these algorithms are necessary to ensure a safer and more reliable job market (Pablo, Guillermo and Alberto, 2023).

1.2 OBJECTIVES

- To evaluate the performance of different models.
- To identify the most suitable model for detecting fake job postings.
- To enhance the overall performance of classification results, focusing on precision and recall metrics.

CHAPTER 2: DATASET DESCRIPTION AND PRE-PROCESSING

2.1 DATASET IMPORT & DESCRIPTION

Google Colab was used throughout the project under the link : https://colab.research.google.com/drive/1G4f7dc2f2I-A0aOzdf_sQgkmAATIw1WN?usp=sharing

The dataset named " fake_job_postings.csv " was imported using the pandas library in Python. The pd.read_csv() function was utilized to read the CSV file containing the dataset. The dataset was obtained from Kaggle and stored in a DataFrame named df.

The dataset comprises 17 features (attributes) and one target variable (classification attribute). Here is a brief description of each feature and overall dataset information:

List of original features in the dataset:

- job_id : Unique Job ID
- title : The title of the job ad entry
- location : Geographical location of the job ad
- department : Corporate department (e.g. sales)
- salary_range : Indicative salary range (e.g. $50,000-60,000)
- company_profile : A brief company description
- description : The details description of the job ad
- requirements : Enlisted requirements for the job opening
- benefits : Enlisted offered benefits by the employer
- telecommuting : True for telecommuting positions
- has_company_logo : True if company logo is present
- has_questions : True if screening questions are present
- employment_type : Full-type, Contract, etc
- required_experience : Executive, Intern, etc
- required_education : Doctorate, Bachelor, etc
- industry : Automotive, Healthcare, Real estate, etc
- function : Consulting, Research, Sales, etc
- fraudulent : target (Classification attribute)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15184 non-null  object
 8   benefits             10668 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

Figure 1: Data information of dataset

The dataset consists of 17,880 entries. It primarily consists of categorical and boolean features, with a few numerical features such as job_id. However, some features contain missing values, as indicated by the "Non-Null Count" in the dataset information. Notably, the location, department, salary_range, employment_type, required_experience, required_education, industry, and function features have missing values.

2.2 DATA PRE-PROCESSING

This stage is divided into five subsections as the flowchart below:

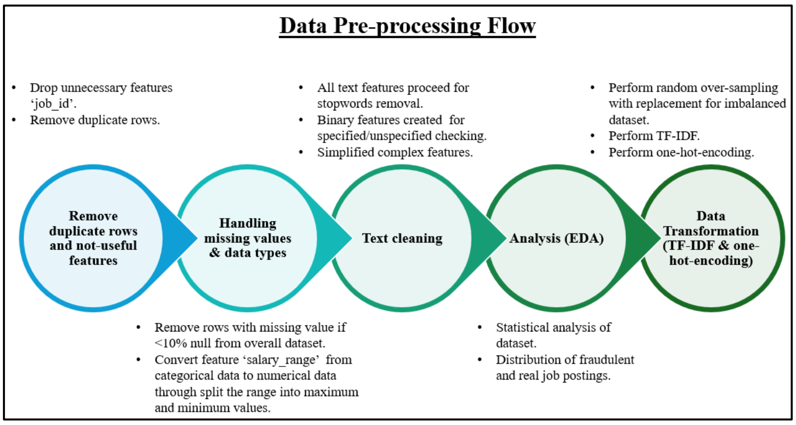


Figure 2: Data Pre-processing Flow

After removing rows with missing value (<10% of overall dataset) and give 'unspecified' for filling up the missing cells, the dataset have no missing value. The details of dataset after missing value cleaning are shown in figure below:

```
<class 'pandas.core.frame.DataFrame'>
Index: 17533 entries, 0 to 17879
Data columns (total 24 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   title                      17533 non-null  object
 1   department                 17533 non-null  object
 2   company_profile            17533 non-null  object
 3   description                17533 non-null  object
 4   requirements               17533 non-null  object
 5   benefits                   17533 non-null  object
 6   telecommuting              17533 non-null  int64
 7   has_company_logo           17533 non-null  int64
 8   has_questions              17533 non-null  int64
 9   employment_type            17533 non-null  object
 10  required_experience        17533 non-null  object
 11  required_education         17533 non-null  object
 12  industry                   17533 non-null  object
 13  function                   17533 non-null  object
 14  fraudulent                 17533 non-null  int64
 15  average_salary             17533 non-null  float64
 16  salary_specified           17533 non-null  int64
 17  company_profile_specified  17533 non-null  int64
 18  description_specified      17533 non-null  int64
 19  requirements_specified     17533 non-null  int64
 20  benefits_specified         17533 non-null  int64
 21  country                    17533 non-null  object
 22  state                      17533 non-null  object
 23  city                       17533 non-null  object
dtypes: float64(1), int64(9), object(14)
memory usage: 3.3+ MB
```

Figure 3: Data information of dataset after missing value cleaning

Then, the number of fraudulent and real job posts are counted and plotted in bar plot as below:

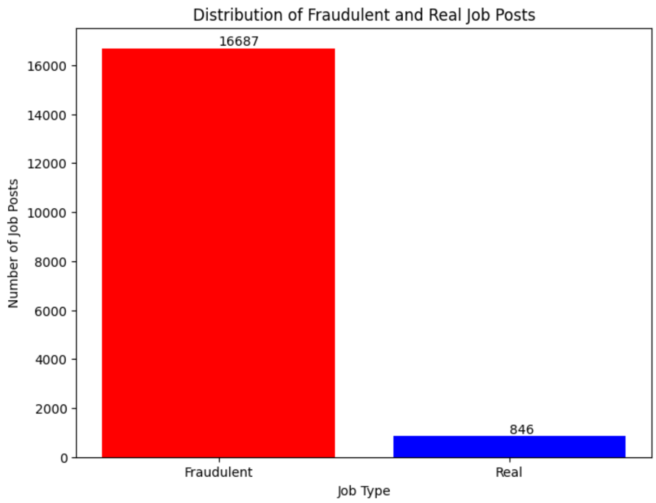


Figure 4: Statistical analysis of dataset

Then, distribution of fraudulent and real job postings are plotted to observe the relationship of each feature on target features.

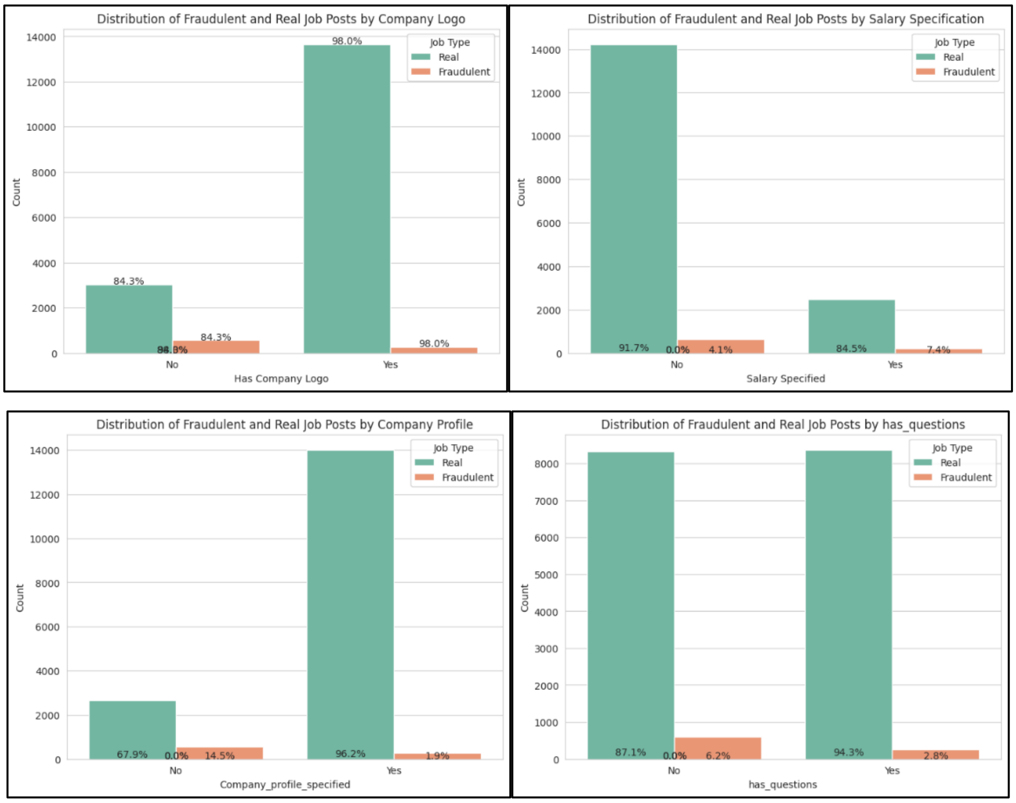


Figure 5: Distribution of fraudulent and real job postings

The statistical tests for 'company_profile' word count and ‘requirements’ word count are performed with P-values calculation. P-value for 'company_profile' is 1.4e-100 and P-value for 'requirements' is 5.8e-12. Based on these p-values (two-tailed test), which is much smaller than the typical significance level of 0.05, both 'company_profile' and 'requirements' word counts appear to be statistically significant predictors of fraudulent job posts. Next, random over-sampling with replacement is used to address imbalanced dataset to avoid machine learning model biased towards the majority class. After that, TF-IDF (Term Frequency-Inverse Document Frequency) is used to evaluate the importance of a word in a document relative to a collection of documents (corpus). Then, one-hot-encoding is used to convert categorical data into a numerical format that can be provided to machine learning algorithms to improve their performance. Each category value is converted into a new categorical column and assigned a binary value of 1 or 0. Lastly, the final dataset after preprocessing are consists of only numerical data to use for model development.

CHAPTER 3: MODEL DEVELOPMENT

In this project, classification supervised learning algorithms are utilized. This is because the dataset, which contains fake job postings, has a target outcome is to determine whether a posting is fraudulent or not. By utilizing classification machine learning algorithms, it will be possible to develop a model that can be trained to predict and learn from this target outcome, generating a function that maps inputs to desired outputs (Nasteski, 2017).

Five models have been selected which are Random Forest (RF), Decision Tree, K-Nearest Neighbour (KNN), Linear Support Vector Machine (SVC) and Stochastic Gradient Descent (SGD). These models were chosen because they represent a diverse set of approaches to classification. The random decision tree algorithm, introduced in 1995 by Bell Labs' Ho, aggregates multiple classifiers like bagging and random space to enhance prediction accuracy through voting-based decision-making (Muhammad et al., 2020). RF and Decision Tree models are powerful for handling complex datasets with multiple features and interactions, offering robustness against overfitting through ensemble learning and simplicity in decision-making processes, respectively.

Next, third machine learning selected for this project is K-nearest neighbor (KNN). KNN is a simple supervised ML algorithm based on the hypothesis "things that look alike." It is a non-parametric classifier used for both regression and classification tasks (Muhammad et al., 2020). Linear Support Vector Classifier (SVC) finds the best separating line between two classes and is the simplest form of Support Vector Machine. However, it performs poorly if the data is not linearly separable (Ali et al., 2022). Lastly, the stochastic gradient descent algorithm randomly selects one sample per iteration for training, reducing computation for large-scale data. This approach leads to faster convergence and higher performance compared to other algorithms. (Wang et al., 2021). By employing this variety of models, this project aims to identify the most effective algorithm for detecting fraudulent job postings in dataset chosen.

All of these models are developed using the Python programming language, which is known for its power and versatility. Python provides dedicated libraries for implementing classification algorithms, making it convenient to train and evaluate classification models. Scikit-learn, a widely adopted Python library for machine learning, has been utilized in this project.

Below are the models used along with the corresponding Python libraries imported.

```python
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn import linear_model
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import confusion_matrix, classification_report
```

Figure : List of Imported Python Libraries.**Figure 6:** List of Imported Python Libraries.

Next, a function named `train_and_evaluate` is defined, which takes in a machine learning model and datasets for training and testing.

```python
def train_and_evaluate(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)
    accuracy = model.score(X_train, y_train) * 100
    print(f"{model.__class__.__name__} Training Accuracy: {accuracy:.2f}%")
    accuracy_test = model.score(X_test, y_test) * 100
    print(f"{model.__class__.__name__} Test Accuracy: {accuracy_test:.2f}%")

    # Generate confusion matrix
    y_pred = model.predict(X_test)
    print(f"\nConfusion Matrix for {model.__class__.__name__}:")
    print(confusion_matrix(y_test, y_pred))

    # Generate classification report
    print(f"\nClassification Report for {model.__class__.__name__}:")
    print(classification_report(y_test, y_pred))

    # # Calculate recall score
    # recall = recall_score(y_test, y_pred)
    # print(f"{model.__class__.__name__} Recall: {recall:.2f}")

    return accuracy, accuracy_test
```

Figure 7: Defined Function Name 'train_and_evaluate'.

Figure : Defined Function Name 'train_and_evaluate'.

After defining the function and developing the models, they are run in the same code chunk to use the same training and testing sets during execution. Each model is then evaluated using cross-validation and evaluation metrics.

```python
# List of models to train
models = [
    SGDClassifier(max_iter=5, tol=None),
    RandomForestClassifier(n_estimators=100),
    KNeighborsClassifier(n_neighbors=3),
    LinearSVC(),
    DecisionTreeClassifier()
]
# X contains all columns except 'fraudulent'
X = df5.drop('fraudulent', axis=1)

# y contains only the 'fraudulent' column
y = df5['fraudulent']

# Split the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Train and evaluate each model
train_accuracies = []
test_accuracies = []
for model in models:
    train_acc, test_acc = train_and_evaluate(model, X_train, y_train, X_test, y_test)
    train_accuracies.append(train_acc)
    test_accuracies.append(test_acc)
```

Figure 8: List of Models to Train and Evaluate.

Figure : List of Models to Train and Evaluate.

Diagram below showed the workflow of the project, starting with Data Collection, followed by EDA, Data Preprocessing, Model Development, Model Evaluation, and end with Cross Validation.

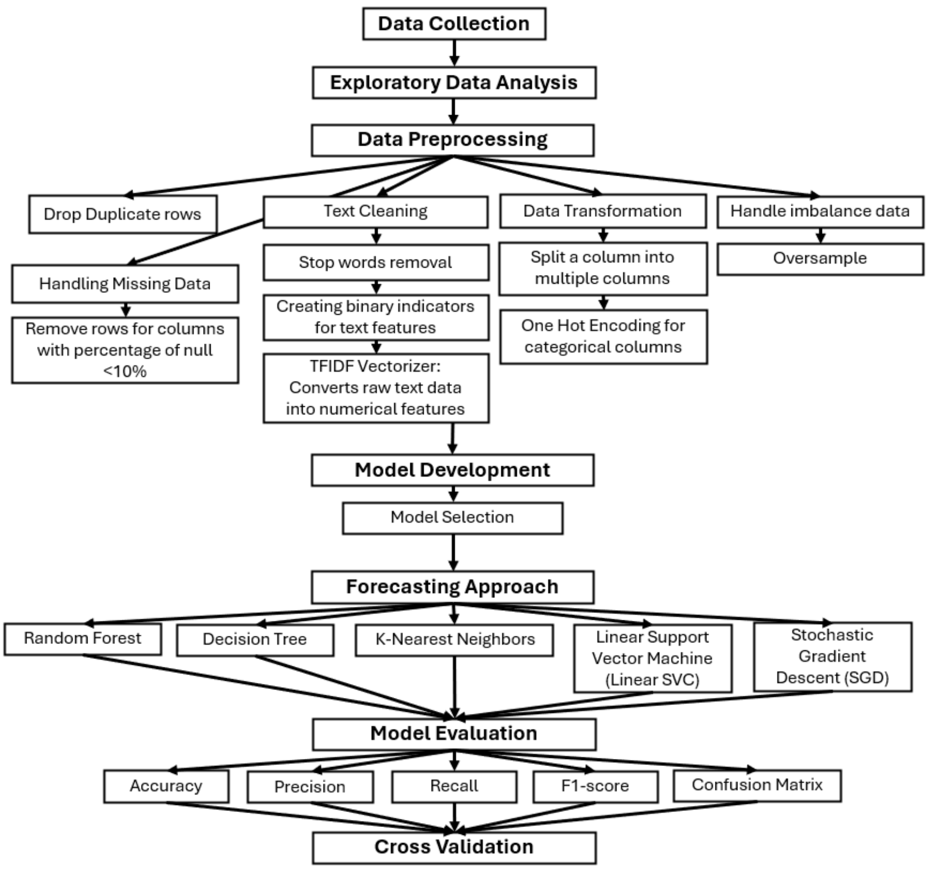


Figure 9: Project architecture diagram

CHAPTER 4: MODEL PRACTICAL IMPLEMENTATION AND COMPARISON

This section provides a detailed comparison of five machine learning algorithms beyond just accuracy. We evaluate each model based on recall, precision, F1-score, computational efficiency, interpretability, robustness, hyperparameter sensitivity, scalability, deployment complexity, and cost.

	Random Forest	Decision Tree	KNN	Linear SVC	SGD
Accuracy	99.95%	98.81%	96.71%	92.43%	73.60%
Recall	99.95%	98.81%	96.72%	92.43%	73.60%
Precision	99.95%	98.83%	92.79%	94%	74.49%
F1 Score	99.95%	98.81%	92.42%	94%	73.36%

Table 1: Models' Accuracy, Recall, Precision & F1 Score Performance

The top 3 model that have highest performance are Random Forest, Decision Tree, and KNN. Among these 3 models, Random Forest stands out as the most balanced and robust model with the highest performance across multiple metrics.

CHAPTER 5: MODEL EVALUATION

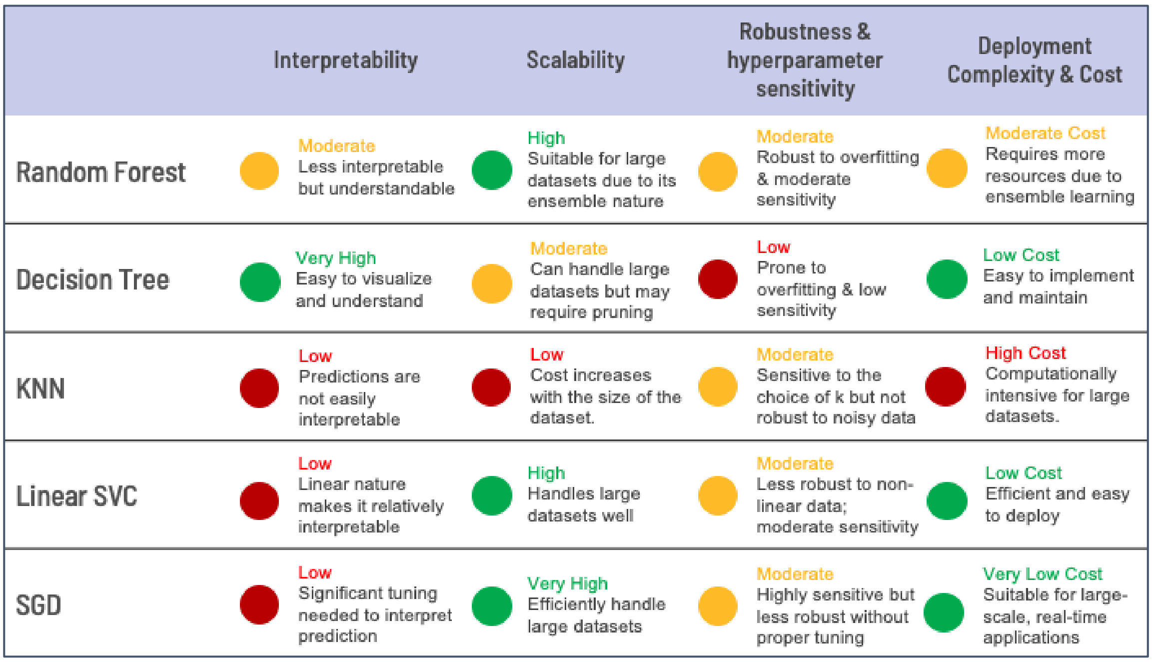
In the pursuit of predicting fraudulent job postings and deploying the appropriate model, it is crucial to find balance in each model's performances. While accuracy metrics are important to get a precise prediction, understanding the computational, efficiency, memory consumption, robustness, scalability aspects etc. of these models are also crucial as it directly impacts the user retention and practically of our predictive system.

Using Python's 'psutil' library or '%%time' command, we can compute and monitor CPU and RAM usage during model training, helping us to evaluate the intensity during deployment. Below is the comparison of each model's computational performance :

	Random Forest	Decision Tree	KNN	Linear SVC	SGD
Time	25.6 secs	20.22 secs	57.63 secs	25.78 secs	3.67 secs
Memory	49.1%	51.4%	51.4%	51.5%	49%

Table 2: Models' Computational Comparison

Considering the overall practicality, below is the comparison of each models :



	Interpretability	Scalability	Robustness & hyperparameter sensitivity	Deployment Complexity & Cost
Random Forest	Moderate Less interpretable but understandable	High Suitable for large datasets due to its ensemble nature	Moderate Robust to overfitting & moderate sensitivity	Moderate Cost Requires more resources due to ensemble learning
Decision Tree	Very High Easy to visualize and understand	Moderate Can handle large datasets but may require pruning	Low Prone to overfitting & low sensitivity	Low Cost Easy to implement and maintain
KNN	Low Predictions are not easily interpretable	Low Cost increases with the size of the dataset.	Moderate Sensitive to the choice of k but not robust to noisy data	High Cost Computationally intensive for large datasets.
Linear SVC	Low Linear nature makes it relatively interpretable	High Handles large datasets well	Moderate Less robust to non-linear data; moderate sensitivity	Low Cost Efficient and easy to deploy
SGD	Low Significant tuning needed to interpret prediction	Very High Efficiently handle large datasets	Moderate Highly sensitive but less robust without proper tuning	Very Low Cost Suitable for large-scale, real-time applications

Figure 10: Model's Practicality Comparison

Decision Tree offers simplicity and interpretability but requires careful handling to avoid overfitting. KNN, while accurate, it's computationally expensive and less scalable. LinearSVC and SGDClassifier are efficient and scalable but may underperform on complex, non-linear data.

Based on the overall comparison, Random Forest is recommended for its overall performance, but simpler models like Decision Tree and LinearSVC might be preferred for their ease of use and efficiency in specific contexts.

CHAPTER 6: MODEL DEPLOYMENT

Once a model is trained and evaluated, deployment of models are typically done as an endpoint to provide real-world practical tools. While organizations can deploy models on a large scale to leverage its full potential, it often incurs significant cost.

In this study, we will focus on deploying the product using tools that offers minimum to no cost such as Google Colab, Github and Streamlit App.

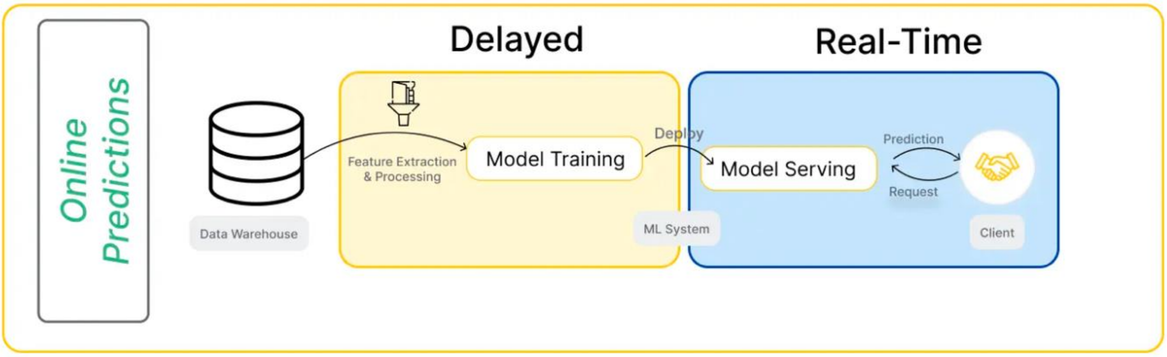


Figure 11: Model Deployment Architecture Diagram (Klushin, P., 2024)

6.1 DEVELOP & STORE MODEL

As the best-performing model was identified as the Random Forest model, the deployment code was designed to focus on data pre-processing step and Random Forest modelling.

```python
# Load data from the CSV file
data = pd.read_csv("fake_job_postings.csv")

# Extract job descriptions and labels from the data
train_data = data['description'].astype(str).tolist()  # 'description' is the column containing job descriptions
train_labels = data['fraudulent'].tolist()  # 'fraudulent' is the column containing labels (1 for fake, 0 for real)

# Define the model
rf_classifier = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', RandomForestClassifier()),
])

# Fit TF-IDF vectorizer and classifier with training data
rf_classifier.fit(train_data, train_labels)

def predict_fake_job_posting(job_description):
    # Make predictions using the loaded model
    prediction = rf_classifier.predict([job_description])
    return prediction[0]
```

Figure 12: Simplified Code for Client's Data Pre-Processing & Model Training

Code was initially tested using Google Colab to identify and resolve any issues, and then was stored in GitHub for integration to Streamlit App.

6.2 TESTING & OPTIMIZATION

Streamlit App's localhost feature were utilized to gauge the draft outcome of deployment. Throughout the process, codes to generate buttons, title, header image, colour formatting etc. was added to enhance the final product's user interface and experience

```python
def main():
    job_description = st.text_area('Enter job description here:')
    if st.button('Predict'):
        prediction = predict_fake_job_posting(job_description)
        if prediction == 1:
            st.write('Prediction: ', unsafe_allow_html=True)
            st.write('<span style="color:red;">Fake</span>', unsafe_allow_html=True)
        else:
            st.write('Prediction: ', unsafe_allow_html=True)
            st.write('<span style="color:green;">Real</span>', unsafe_allow_html=True)

if __name__ == '__main__':
    main()
```

Figure 13: User Interface Code

After few pilot testing, the final draft can be seen as below :

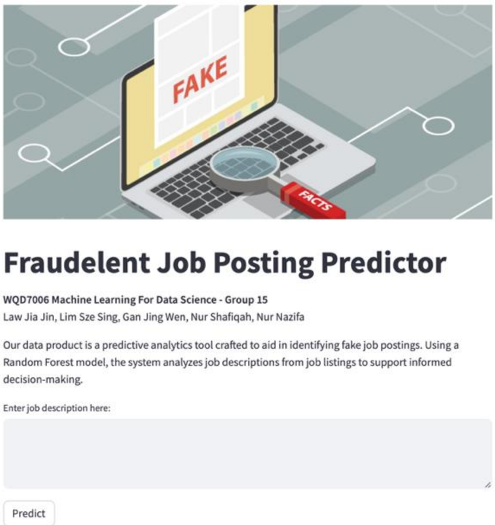


Figure 14: Model Deployment Final Draft

6.3 DEPLOYMENT & MONITORING

After completing the final draft, the next step of this process is to deploy the web application and monitor its performance. The product is deployed at: https://fakejobsdeploy-wqd7006-group15.streamlit.app Given previous evaluation of the Random Forest's model, clients are expected to get the prediction outcome within 30 seconds and 99.95% accuracy.

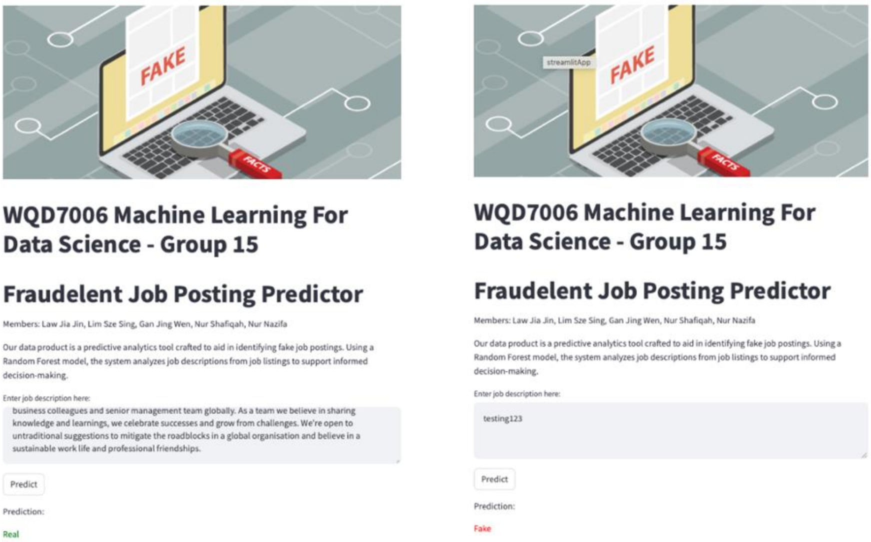


Figure 15: Model Deployment Prediction Simulation

Post simulation, it was observed that the application performs as expected with an average execution time of 30 seconds and moderately accurate prediction.

CHAPTER 7: APPROACH AND INNOVATION

The approach and innovation to fraud detection in this project encompass aspects such as data preprocessing, model evaluation, deployment strategy, and performance monitoring & user experience as shown in the table below:

Table : List of Aspects for Innovative Approach.

Aspect	Description
Data Preprocessing	Advanced data preprocessing techniques improve how imbalanced datasets, text features, and categories are handled. By using adaptive oversampling, context-aware TF-IDF, and hierarchical one-hot encoding, machine learning models become more accurate and robust. These methods go beyond traditional techniques, enhancing fraud detection and other applications significantly.
Model Evaluation	Evaluate models using a evaluation metrics. This approach simplifies comparing and choosing the best model by considering all important factors equally. It ensures a balanced assessment to identify the most suitable model overall.
Deployment Strategy	The deployment strategy prioritizes platforms such as Google Colab, GitHub, and Streamlit for deploying machine learning models. These platforms are cost-effective or free, which helps minimize costs for smaller businesses or projects with limited budgets.
Performance Monitoring & User Experience	The deployment includes monitoring the web application's performance to maintain high standards of service. By ensuring predictions are delivered within 30 seconds with 99.95% accuracy, the deployment not only meets but enhancing user satisfaction and trust in the application.

Table 3: List of Aspects for Innovative Approach.

CHAPTER 8: COMMERCIALIZATION

The job post fraud detection system is a niche data product that can address critical challenges for various businesses and organizations. The business model focuses on licensing technology and offer customization services, allowing for tailored solutions at an additional fee. For instance, personalized models can be developed tailored to a specific industry, geographic location, or job categories. Alternatively, integration partnerships with existing job posting platforms will further extend the system's reach and impact. By targeting popular job posting platforms in Malaysia such as JobStreet, Indeed, and Maukerja.my, the pain points related to time-consuming human screening can be addressed through the implementation of the fraud detection model. There are customers who may have established HR management systems or applicant tracking systems in place. By providing customization services, the fraud detection system can be seamlessly integrated with their existing infrastructure, ensuring smooth data flow and minimal disruption to their workflow.

The go-to-market strategy is crucial in ensuring the successful recognition of service by potential clients. By partnering with respected job posting platforms, the business can gain credibility and access to a broader network of potential clients. Additionally, offering free trials or pilot programs to potential customers allows them to experience the benefits firsthand, fostering trust and encouraging adoption. These strategic initiatives not only drive awareness and credibility but also promote meaningful engagement and ultimately position the business for long-term success in the market.

To ensure the necessary startup funds, it is essential to explore various avenues. One such avenue is applying for the Malaysia Digital X-port Grant (MDXG), which specifically supports Malaysian technology companies in both product development and commercialization endeavours. Additionally, seeking potential investors who align with the business's vision and goals will provide the necessary capital for growth and expansion.

In summary, setting go-to-market strategy to offer customizable solutions for enterprise customers, securing startup funds and investor partnerships is crucial for executing the strategy effectively and ensuring long-term success in the market. Together these efforts position the business for long-term success and market leadership in the dynamic landscape of fraud detection and prevention.

CHAPTER 9: MARKET VALIDATION

The Market Validation Plan has three main goals. First, it seeks to identify market demand by assessing the need for a fake job posting prediction tool among target users. Second, it aims to evaluate product-market fit by determining if the solution meets the specific needs of users. Third, it will determine how much potential customers are willing to pay for this service.

9.1 MILESTONES

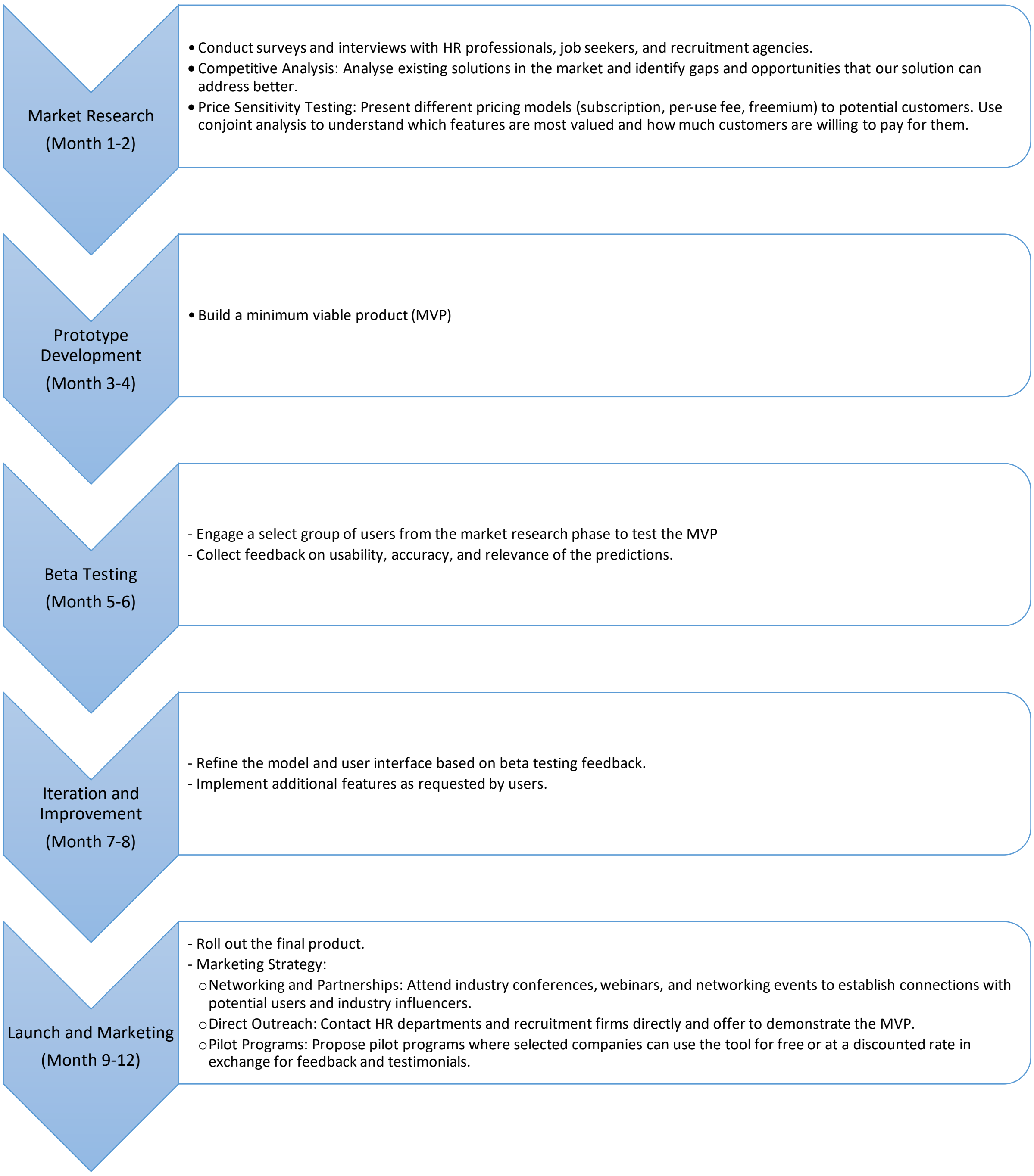


Figure 16: Milestones

9.2 SUCESS METRICS

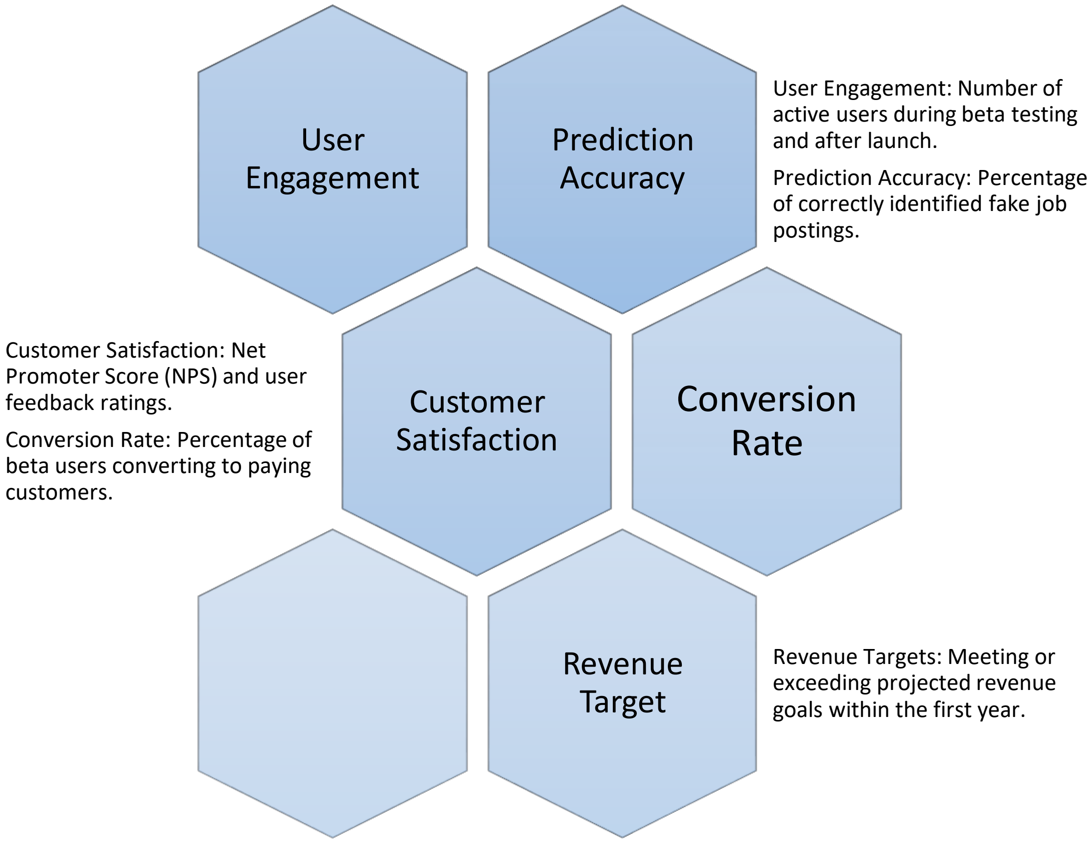


Figure 17: Sucess Metrics

By following this comprehensive market validation plan, we will be able to accurately gauge the market demand for our fake job posting prediction model, determine the optimal pricing strategy, and build a strong foundation of early adopters and advocates for our solution.

CHAPTER 10: COST ANALYSIS

As a startup company, the initial cost analysis for developing, deploying, and maintaining a machine learning system for fraud detection of fake job postings are as table below:

Cost Breakdown	Roles	Function	Estimated Salaries (MYR)
Personnel Structure (Glassdoor, 2024)	Data Scientist	Data collection, preprocessing and model development.	RM84,000/year
	Software Developer	Implementing ML models into necessary software infrastructure.	RM60,000/year
	Project Manager	Maintain relationship with stakeholders and tracking the project progress.	RM72,000/year
Technology & Tools & Infrastructure (Calculator, 2024)	Software subscription	Budget for essential software subscriptions and tools.	RM0/year (use open-source tools such as Python, Jupyter, etc.)
	Deployment tools	Streamline development process.	RM15,715.68/year
	Cloud computing	For model training and deployment.	
	Data storage	Cloud-based storage solutions.	
	Hardware	Purchase of basic hardware for development and testing, such as PC.	RM40,000/times (estimate RM8,000 per headcount)
Operational Costs	Utilities	For utilities service.	RM20,000/year
	Internet (Unifi, 2024)	For internet service.	RM3,828/year
	Working Space (WeWork, 2024)	Office space. At No. 3, Jalan Bangsar, KL Eco City Kuala Lumpur, Federal Territory of Kuala Lumpur 59200.	RM5,508/year
		Total Cost =	RM301,051.68/year

Table 4: Cost Analysis

name	quantity	region	service_id	sku	total_price, USD	notes
N1 Predefined Instance Core running in Americas	2920	us-central1	6F81-5844-456A	2E27-4F75-95CD	92.30412	
N1 Predefined Instance Ram running in Americas	10950	us-central1	6F81-5844-456A	6C71-E844-38BC	46.39515	
Storage PD Capacity	20	us-central1	6F81-5844-456A	D973-5D65-BAB2	0	
Network Data Transfer GCP Inter Region within Asia	500		95FF-2EF5-5EA1	A115-9CB2-F553	37.2529	
Standard Storage Asia Multi-region	1	asia	95FF-2EF5-5EA1	E653-0A40-3B69	24.21439	
Network Data Transfer GCP Replication within Asia	1	asia		CA61-E18A-B2D6	74.50581	
				Total Price:	274.67237	

Prices are in US dollars, effective date is 2024-05-21T08:14:39.000Z

The estimated fees provided by Google Cloud Pricing Calculator are for discussion purposes only and are not binding on either you or Google. Your actual fees may be higher or lower than the estimate.

Url to the estimate: https://cloud.google.com/calculator?dl=CiQ3NmE5NDA0ZS1mODM2LTQ3YjEtOTVmYS0wNzY4OTA1MjgwZWU=

Figure 18: Reference of Technology & Tools & Infrastructure

CHAPTER 11: CONCLUSION AND FUTURE WORK

This analysis has demonstrated that the Random Forest algorithm provides the best overall performance for detecting fraudulent job postings, excelling in accuracy, precision, recall, and F1-score, while also maintaining a balance between computational efficiency and scalability. Decision Tree and Linear SVC offer advantages in interpretability and deployment simplicity, making them suitable alternatives in contexts where these factors are prioritized. KNN, although accurate, is hindered by high computational costs and scalability issues. SGD Classifier, with its low complexity and high scalability, can be useful in large-scale applications despite lower performance on complex data.

Moving forward, the future work can focus on optimizing the selected models to enhance their robustness and efficiency further. This will include experimenting with hyperparameter tuning, feature engineering, and integrating additional data sources to improve model accuracy and generalizability. Also, the future work may explore advanced machine learning techniques such as deep learning and ensemble methods to capture more complex patterns in the data. Additionally, implementing real-time model monitoring and automated retraining pipelines will help maintain the model's performance and adapt to new types of fraudulent activities as they emerge. By continuously refining our approach, a highly reliable and scalable fraud detection system that can effectively protect users from fraudulent job postings is highly possible to be deployed in the market.

REFERENCES

Ali, I., Mughal, N., Khand, Z. H., Ahmed, J., & Mujtaba, G. (2022). Resume classification system using natural language processing and machine learning techniques. Mehran University Research Journal Of Engineering & Technology, 41(1), 65-79.

Bhd, T. D. (2024). Glassdoor. Retrieved from https://www.glassdoor.com/Job/malaysia-software-developer-jobs-SRCH_IL.0,8_IN170_KO9,27.htm

Calculator, G. C. (2024). Google Cloud. Retrieved from https://cloud.google.com/products/calculator?_gl=1*132v8qp*_up*MQ..&gclid=8de111347c1718f904b877782af9be2c&gclsrc=3p.ds&hl=en&dl=CiQyMjNmNTBkNC11NDQwLTQxYjgtYTg0Mi1iMWE0ZjM1YmEzZTAQCBokQTQ0RjM5OEYtNzQyRi00MjAwLThCN0YtQkYyNjdFOUM4RUJF

Glassdoor. (2024). Glassdoor. Retrieved from https://www.glassdoor.com/index.htm

Klushin, P. (2024, January 29). *What does it take to deploy ML models in production? | Qwak's Blog*. Www.qwak.com. https://www.qwak.com/post/what-does-it-take-to-deploy-ml-models-in-production

Muhammad, L. J., Algehyne, E. A., & Usman, S. S. (2020). Predictive supervised machine learning models for diabetes mellitus. SN Computer Science, 1(5), 240.

Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4(51-62), 56.

NodeFlair. (2024). Glassdoor. Retrieved from https://www.glassdoor.com/Job/malaysia-data-scientist-jobs-SRCH_IL.0,8_IN170_KO9,23.htm

Pablo, Guillermo and Alberto. (2023). Fake Job Detection with Machine Learning: A Comparison. Monterrey, Mexico: ResearchGate. Retrieved from https://www.researchgate.net/publication/371508732_Fake_Job_Detection_with_Machine_Learning_A_Comparison

Unifi. (2024). Unifi. Retrieved from https://unifi.com.my/all-in-one

Wang, X., Yan, L., & Zhang, Q. (2021, September). Research on the application of gradient descent algorithm in machine learning. In 2021 International Conference on Computer Network, Electronic and Automation (ICCNEA) (pp. 11-15). IEEE.

WeWork. (2024). WeWork. Retrieved from https://www.wework.com/search?capacity=20&workspace=privateOffice