EMPLOYEE ACCESS MANAGEMENT USING MACHINE LEARNING

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology
in
Computer Science & Engineering
School of Engineering & Sciences

submitted by

Somasani Chidvila(AP20110010649)

Kataboina Sriharsha(AP20110010656)

Under the Guidance of

Prof. Amit Kumar Mandal

Department of Computer Science & Engineering
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2024

DECLARATION

I undersigned hereby declare that the project report **Employee access management using machine learning** submitted for partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in the Computer Science & Engineering, SRM University-AP, is a bonafide work done by me under supervision of Prof. Amit Kumar Mandal. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree of any other University.

Place	:	Date	: May 2, 2024
Name of student	: Somasani Chidvila	Signature	: ....
Name of student	: Kataboina Sriharsha	Signature	: ....

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRM University-AP

Neerukonda, Mangalgiri, Guntur

Andhra Pradesh - 522 240

CERTIFICATE

This is to certify that the report entitled **Employee access management using machine learning** submitted by **Somasani Chidvila, Kataboina Sriharsha** to the SRM University-AP in partial fulfillment of the requirements for the award of the Degree of Bachelors of Technology in Computer Science and Engineering is a bonafide record of the project work carried out under our guidance and supervision. This report, in any form, has not been submitted to any other University or Institute for any purpose.

Project Guide	Head of Department

Name : Prof. Amit Kumar Mandal

Signature: ....

Name : Prof. Niraj Upadhayaya

Signature: ....

ACKNOWLEDGMENT

I wish to record my indebtedness and thankfulness to all who helped me prepare this Project Report titled **Employee access management using machine learning** and present it satisfactorily.

I am especially thankful for my guide and supervisor Prof. Amit Kumar Mandal guide in the Department of Computer Science & Engineering for giving me valuable suggestions and critical inputs in the preparation of this report. I am also thankful to Prof. Niraj Upadhayaya, Head of Department of Computer Science & Engineering for encouragement.

My friends in my class have always been helpful and I am grateful to them for patiently listening to my presentations on my work related to the Project.

Somasani Chidvila,kataboina Sriharsha
(Reg. No. AP20110010649,AP20110010656)
B. Tech.
Department of Computer Science & Engineering
SRM University-AP

ABSTRACT

The ability to effectively and efficiently manage access permissions inside an organization is vital to ensuring that personnel can execute their duties without additional delays or hurdles. In general, access can be done manually. The procedure begins when an employee makes a request and must submit the appropriate information. The request would then be manually reviewed to determine if access was granted or refused. This is a lengthy and time-consuming process. There is also the risk of errors. The paper aims to compare the effectiveness of Q-learning and supervised learning in automating access management within organizations. Reinforcement learning is a subfield of machine learning in which an agent learns to interact with its environment to maximize some concept of cumulative reward. Specifically, a Q-learning algorithm has been proposed as a solution to streamline and optimize the access management procedure. We aim to develop and deploy a Q-learning model and a supervised machine-learning model that can effectively compare and determine whether to approve or deny access based on various attributes of the employee's request.These attributes include the employee's role, department, and other relevant information that typically influences access rights decisions. Supervised learning that involves training the model using a labeled dataset. The model learns to predict outcomes based on this data. Usage of current organizational data, such as personnel responsibilities, departmental connections, and other relevant criteria, was trained to automatically assess and accurately classify each access request as either 'approve' or 'deny' based on the employee's categorical attributes. The major purpose of using such a model is to greatly decrease manual work and speed up the process of allocating access permissions

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The major goal of this project is to enhance employee access rights management by reducing manual intervention and operational complexity. This paper undertakes an analysis of machine learning techniques such as supervised learning and reinforcement learning for access management.

This project is a binary classification problem in which the goal is to determine if an employee should have access to a given resource. Previous research primarily utilized machine learning approaches to overcome this difficulty. However, this study explores the use of Q-learning as an alternate technique.

This project provides a chance to experiment with innovative approaches to addressing a variety of important concerns. It entails investigating new issues, notably in the context of supervised learning and Q-learning, which is an application of reinforcement learning

Throughout the study, the emphasis was on improving data analysis, model selection, and research on multiple models. The efficiency of several models was examined by comparing their results of accuracy. This research includes a comprehensive review of Q-learning's efficacy compared to traditional learning techniques.

Chapter 2

MOTIVATION

2.1 IT HELPS TO IDENTIFY A REAL-TIME PROBLEM AND PROVIDE A SOLUTION

Identifying a project concept begins with addressing real-time challenges or circumstances and finding appropriate solutions. In the framework of the Employee Access Challenge project, we focused on addressing the difficulties of employee access rights to eliminate manual effort. This project provided a chance for us to apply our machine-learning knowledge and investigate alternate methodologies, such as Q-learning. Our goal in solving this real-world challenge was to develop novel solutions and explore learning. [1]

2.1.1 It helps to choose diversified research topics.

Research papers and articles are extremely helpful in assisting in selecting a variety of study subjects. By examining several articles, can gain in-depth information about a broad range of topics and find out how different techniques and methods are used in real-world circumstances.

Like we did within our situation, we looked at research papers to gain insight into how reinforcement learning works in the decision-making process. These publications gave extensive insights into our project's needs and assisted us in understanding how supervised learning approaches, such as data analysis, feature engineering, and machine learning algorithms like Random Forest, SVM, Logistic Regression, and KNN are used.

Furthermore, our exploration of reinforcement learning through real-world scenario research publications expanded our grasp of the decision-making process, particularly how well it works when applying approaches such as Q-learning. These studies carefully described predicted actions, allowing for a better understanding of current patterns and techniques in the context.[2]

The experience gained from research papers and publications enabled us to choose diverse study topics, offering significant insights into technology breakthroughs and approaches. We improved our learning experience by creating project portfolios based on new concepts.

2.1.2 It helps to choose appropriate project topics and mentor carefully.

Collaborative efforts, including group discussions and brainstorming sessions, were critical to the progress of our project. We improved our problem-solving, management, and creative thinking skills by working together.

Our mentor guided us through the project planning process, identifying our areas of interest and reviewing the numerous project alternatives accessible to us.

We focused on machine learning and reinforcement learning projects since they matched our interests and provided enough opportunities for research and development.

Chapter 3

LITERATURE SURVEY

In this paper, we compare supervised learning and reinforcement learning approaches to address the difficulty of managing employee access. We explore the accuracy of multiple algorithms in predicting employee access permissions for the employee in authorizations.

Previous research has mostly focused on binary classification with machine learning methods including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Regression, and Random Forest. While these classifiers have demonstrated promising results, we propose using reinforcement learning, especially Q-learning, to increase the accuracy and efficiency of employee access prediction. In the next sections, evaluation and results will determine which learning model is appropriate.

This outlines a complete method for employee access management, which includes data pre-processing, feature extraction, and categorization. We compare the performance of traditional machine learning algorithms to Q-learning, highlighting the benefits and drawbacks of each method.

We also discuss the issues connected with employee access management, the requirement for real-time decision-making. By integrating Q-learning, we hope to address these issues by allowing the system to learn and adapt to changing access patterns over time.

Chapter 4

DESIGN AND METHODOLOGY

We methodically developed our models, starting with data pre-processing and moving on to model implementation and training.[3]

- Data pre-processing involves cleaning the dataset, identifying missing or duplicate values, and detecting outliers.
- Exploratory Data Analysis (EDA) involves visually analyzing a dataset to understand its distribution and structure.
- Feature Selection:Extracting features from raw data improves algorithm performance by providing useful input.
- Data Segregation: The separation of datasets into training and testing sets,suitable for model implementation.
- Compare models to identify the best fit for the challenge based on feature selection accuracy.
- Model Selection: Build a machine learning model and q learning model based on accuracy.
- Training Process: Divide the dataset into training and testing sets, train models, and tune hyperparameters.
- Evaluation metrics include accuracy, precision, confusion matrix to assess model performance.

Chapter 5

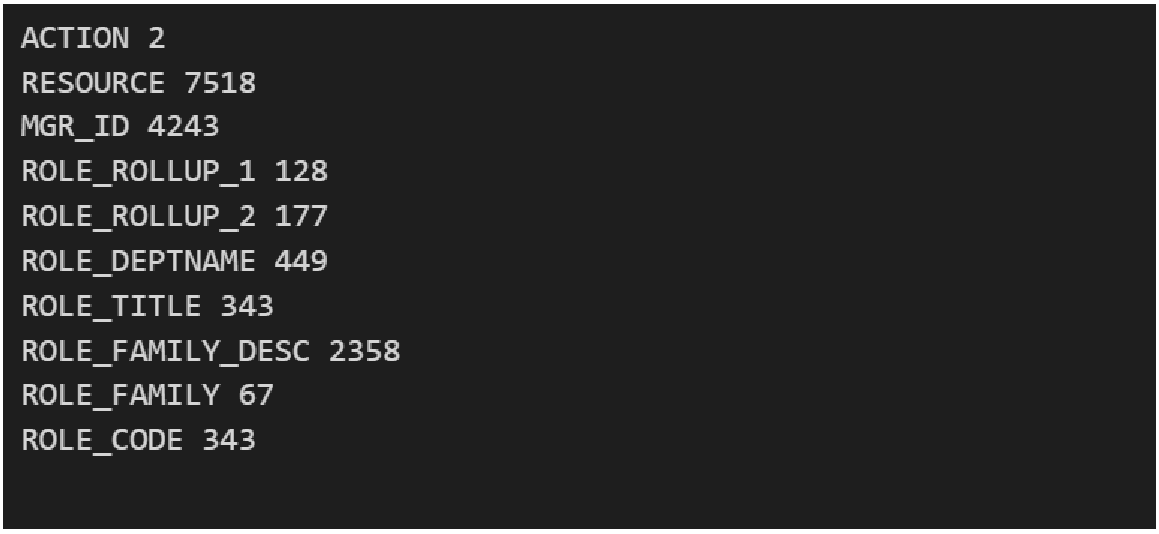
IMPLEMENTATION

The dataset has 32,769 rows and 10 columns. The variable representing the amount of unique values in each column denotes the variety of information provided by each characteristic, and the dataset is free of duplicate rows and missing values. The goal of the supervised learning issue posed by this dataset is to predict, given certain criteria, the appropriate level of access that an employee should have.

The dataset contains unique characteristics about an ACTION -If the action value is 1 the resource was approved, 0 if the resource was not. RESOURCE-(An ID for each resource) MGR ID -(The EMPLOYEE ID of the manager of the current) ROLE ROLLUP 1-(Company role grouping category id 1 (e.g. US Engineering)) ROLE ROLLUP 2-(Company role grouping category id 2 (e.g. US Retail)) ROLE DEPTNAME-(Company role department description (e.g. Retail)) ROLE TITLE-(Company role business title description (e.g. Senior Engineering Retail Manager)) ROLE FAMILY DESC-(Company role family extended description (e.g. Retail Manager, Software Engineering)) ROLE FAMILY-(Company role family description (e.g. Retail Manager)), ROLE CODE-(Company role code, role (e.g. Manager)). Every row represents a separate observation, and each column corresponds to a particular feature.[4]

5.1 DATA PRE-PROCESSING

During this step, duplicates and missing values are checked to verify the integrity of the data. The data set is divided into action categories. Our project comprises of two 1/0 (approved or refused) actions to gain insights into the data set's features and analysis. The data counts for each of their columns are shown below.



```
ACTION 2
RESOURCE 7518
MGR_ID 4243
ROLE_ROLLUP_1 128
ROLE_ROLLUP_2 177
ROLE_DEPTNAME 449
ROLE_TITLE 343
ROLE_FAMILY_DESC 2358
ROLE_FAMILY 67
ROLE_CODE 343
```

Figure 5.1: Number of unique values of each column.

However, that ROLE CODE and ROLE TITLE have the same number of unique values (343),suggests that each title corresponds to a unique code.According to the feature specifications, ROLE CODE is a unique code allocated to each role, whereas ROLE TITLE reflects the title connected with each role.The fact that both features have the same amount of unique values indicates that each title is assigned a unique code. As a result, for each unique role title, there is a matching unique role code, resulting in an equal number of unique values for both characteristics.

The figure 5.2 observation suggests that the dataset is imbalanced, with many more accessed requests than denied requests.

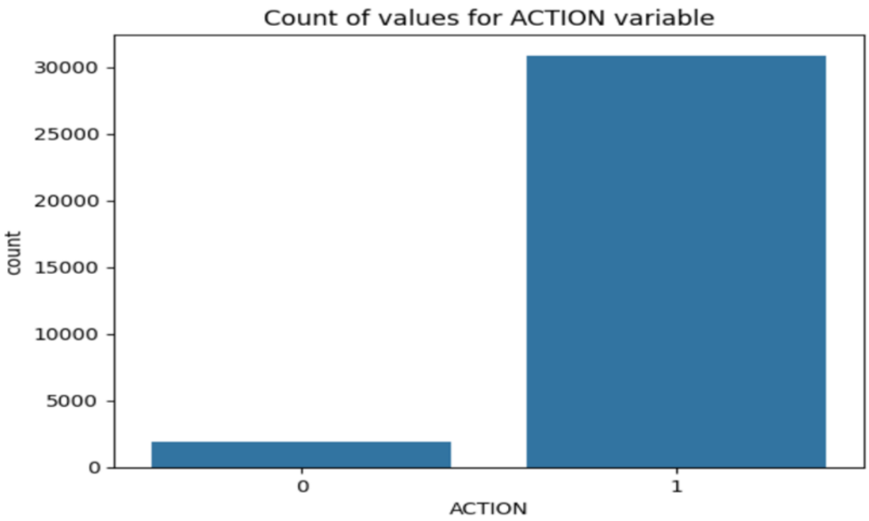


Figure 5.2: Visual representation of the action variable.

5.2 EDA(EXPLORATORY DATA ANALYSIS)

The visualization approach for data exploration. The figures show the distribution of various characteristics inside the feature variable and probability densities. Kernel Density Estimation (KDE) plots are used to depict the distribution of various variables in both approved and denied requests.

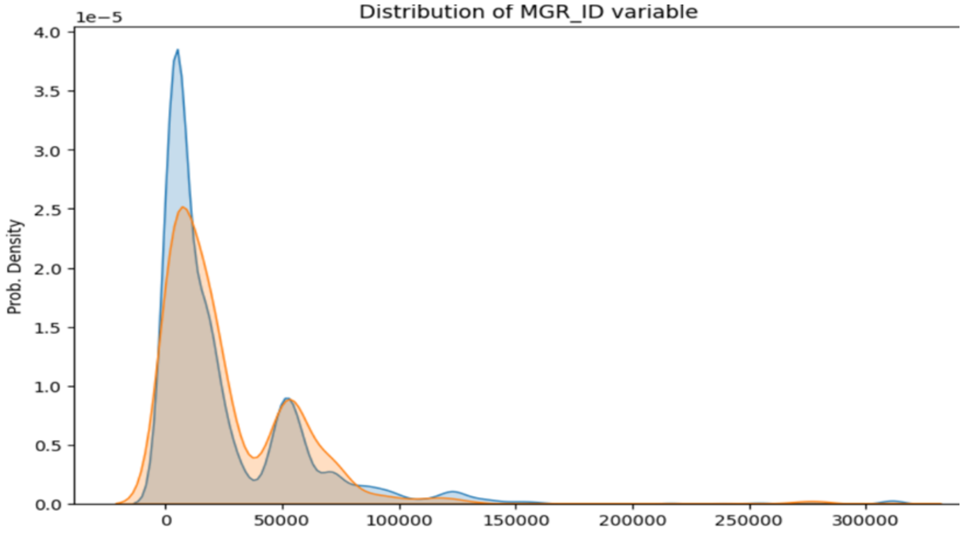


Figure 5.3: MGR ID Distribution

The figure indicates that across the range of 0 to 20,000, the density of authorized requests is greater than that of rejected requests.

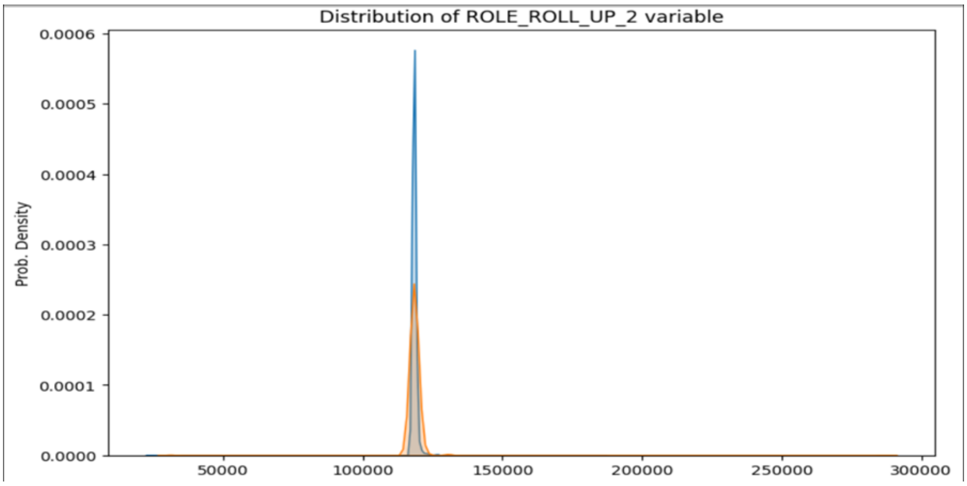


Figure 5.4: ROLE ROLL UP 2 Distribution

According to this figure, the large spike implies that the majority of the values happened within the range (100000 to 150000). And the densities of approved requests are larger.

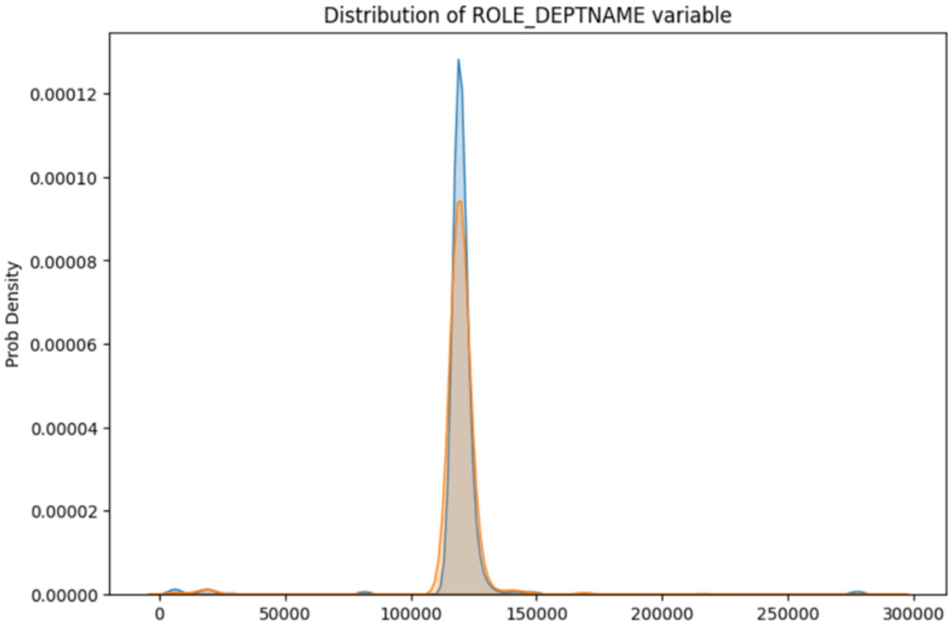


Figure 5.5: ROLE DEPT NAME Distribution

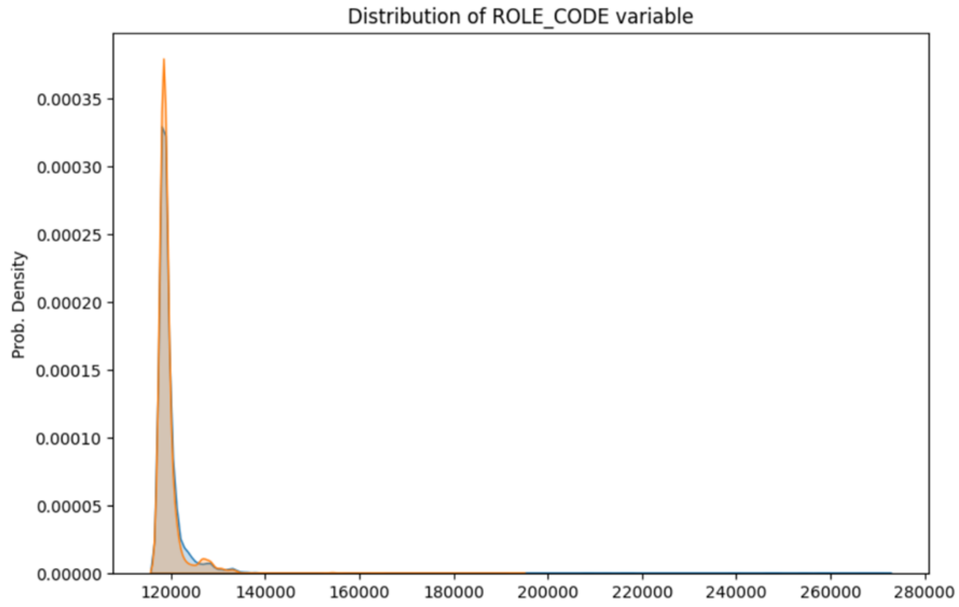


Figure 5.6: ROLE CODE Distribution

The observation suggests that for points lying between 110,000 and 130,000, the density of authorized requests is larger than rejected requests.

5.2.1 HEAT MAP

Heatmaps are a great technique for visualizing correlation matrices. In a correlation matrix, a number closer to 1 implies a high positive correlation whereas a value closer to -1 shows significant negative correlation. A value of zero indicates no association between characteristics. The heatmap shows that most of the correlation values are close to zero. The correlation coefficient between ROLE TITLE and ROLE FAMILY DESC is 0.17, whereas ROLE TITLE and ROLE CODE are 0.16. Although the majority of the correlation coefficients are close to zero, there appears to be a linear link between ROLE CODE and ROLE TITLE. Because each title has a distinct ROLE CODE, there may be some correlation between these two variables.

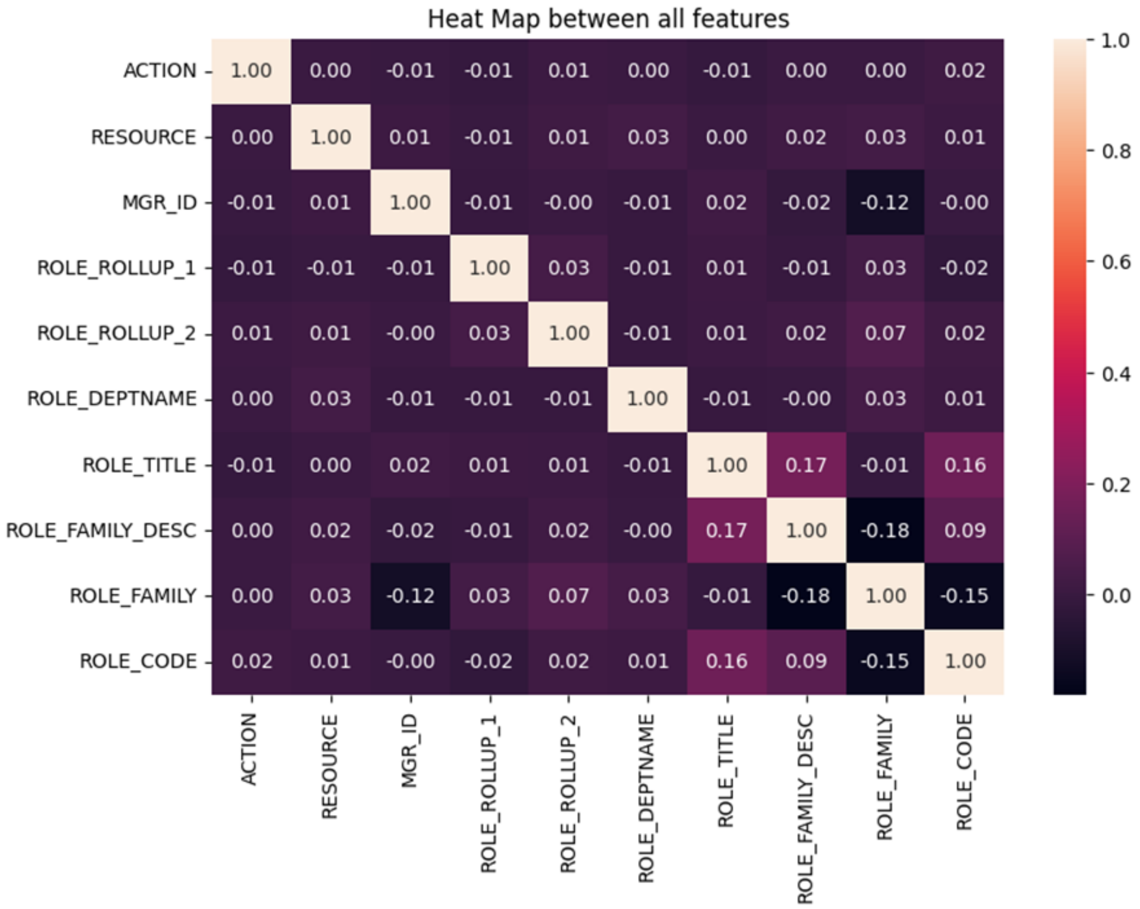


Figure 5.7: Heat map

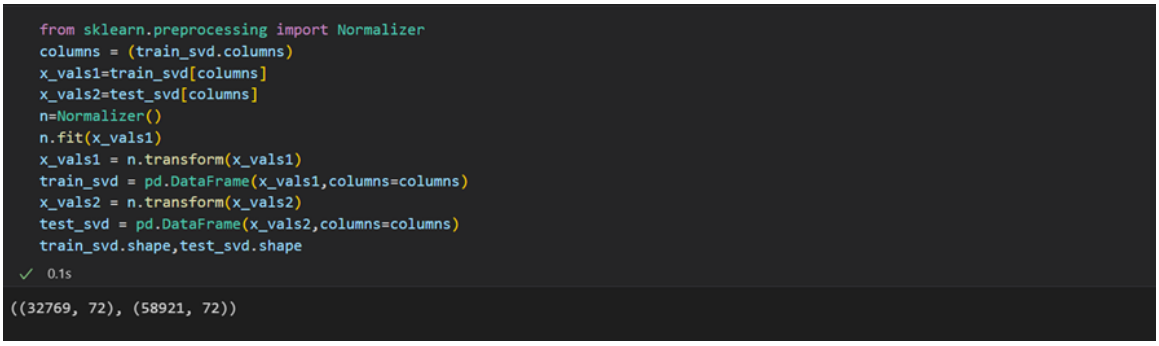
5.3 FEATURE ENGINEERING

Feature Engineering is a vital phase in the machine learning process that converts raw data into features that machine learning algorithms can better understand, hence enhancing performance.

This process entails developing new features, choosing the most relevant ones, and improving existing ones to make them more helpful. Feature engineering seeks to describe data in a manner that improves the algorithm's ability to identify patterns and generate correct predictions.[5]

5.3.1 SVD (Singular value decomposition

Singular Value Decomposition (SVD) is a technique for reducing dimensionality, data compression, and noise, all of which contribute to data complexity reduction.

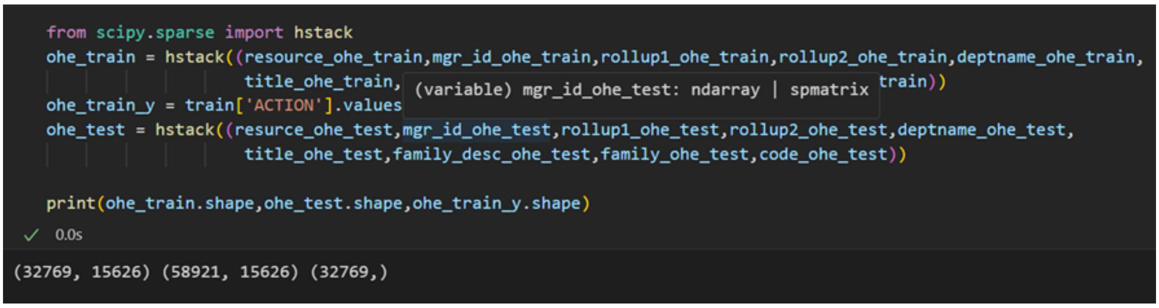


```
from sklearn.preprocessing import Normalizer
columns = (train_svd.columns)
x_vals1=train_svd[columns]
x_vals2=test_svd[columns]
n=Normalizer()
n.fit(x_vals1)
x_vals1 = n.transform(x_vals1)
train_svd = pd.DataFrame(x_vals1,columns=columns)
x_vals2 = n.transform(x_vals2)
test_svd = pd.DataFrame(x_vals2,columns=columns)
train_svd.shape,test_svd.shape
✓ 0.1s
((32769, 72), (58921, 72))
```

Figure 5.8: Singular value decomposition

5.3.2 One hot encoding

One hot encoding is a technique for converting categorical data into numerical representation. It enables the model to learn the correlations between categorical variables independently. The data is represented as a binary vector. A one hot encoding represents category variables as binary vectors.First, the categorical values must be converted to integers.Then, each integer value is represented as a binary vector containing all zero values except the integer's index, which is denoted by a 1.[6]



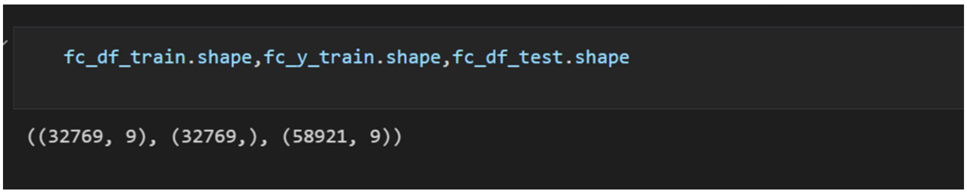
```
from scipy.sparse import hstack
ohe_train = hstack((resource_ohe_train,mgr_id_ohe_train,rollup1_ohe_train,rollup2_ohe_train,deptname_ohe_train,
                    title_ohe_train, (variable) mgr_id_ohe_test: ndarray | spmatrix train))
ohe_train_y = train['ACTION'].values
ohe_test = hstack((resurce_ohe_test,mgr_id_ohe_test,rollup1_ohe_test,rollup2_ohe_test,deptname_ohe_test,
                    title_ohe_test,family_desc_ohe_test,family_ohe_test,code_ohe_test))

print(ohe_train.shape,ohe_test.shape,ohe_train_y.shape)
✓ 0.0s
(32769, 15626) (58921, 15626) (32769,)
```

Figure 5.9: One hot encoding

5.3.3 Frequency coding

Frequency coding is a technique for encoding categorical values by substituting each category's frequency in the dataset. This is important for understanding the distribution of the categories. Frequency coding is a basic yet efficient method for representing categorical data based on the frequency of its values in the collection. Counting Frequencies: For each unique value in a column, determine how many times it appears in the dataset. Encoding with Frequencies: Replace the original values with the appropriate frequencies. For example, if a specific value shows 10 times in the dataset, you would replace it with the number 10.



```
fc_df_train.shape,fc_y_train.shape,fc_df_test.shape
```

```
((32769, 9), (32769,), (58921, 9))
```

Figure 5.10: Frequency coding

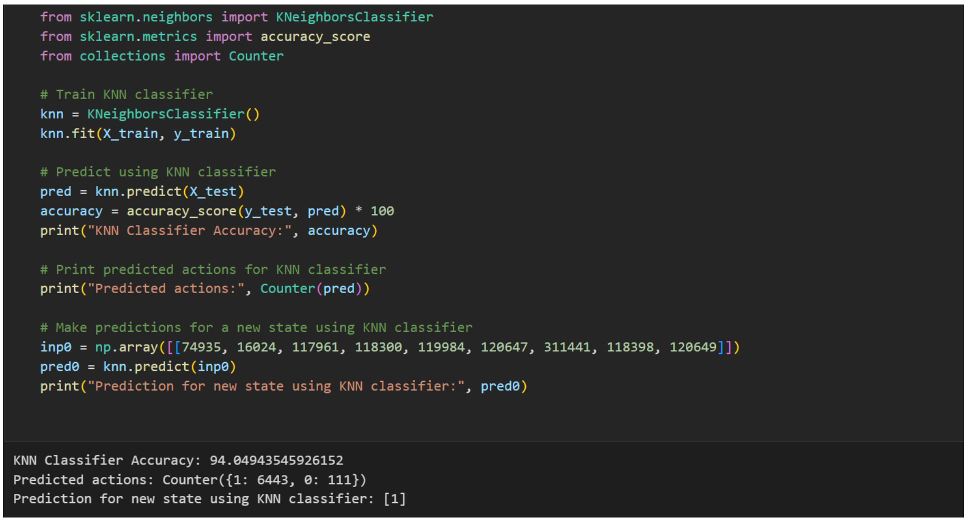
5.3.4 Response encoding

Response encoding, also known as target encoding, is a technique for encoding categorical variables that involves replacing each category with the mean of the target variable in that category.This encoding approach captures the link between the category feature and the target variable, making it ideal for classification tasks.In target encoding, each category is represented by the likelihood that a given data item belongs to a specific class.

5.4 MACHINE LEARNING MODELS

5.4.1 KNN

KNN (K closest neighbor): This technique is applicable to both classification and regression applications. This guesses the class label for a new datapoint using the class labels of its nearest neighbors. Given a new data point, KNN computes the distances to all training data points and chooses the K closest neighbour. The anticipated class label for the new data point is chosen by a majority vote of its K closest neighbors.[7]



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from collections import Counter

# Train KNN classifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)

# Predict using KNN classifier
pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, pred) * 100
print("KNN Classifier Accuracy:", accuracy)

# Print predicted actions for KNN classifier
print("Predicted actions:", Counter(pred))

# Make predictions for a new state using KNN classifier
inp0 = np.array([[74935, 16024, 117961, 118300, 119984, 120647, 311441, 118398, 120649]])
pred0 = knn.predict(inp0)
print("Prediction for new state using KNN classifier:", pred0)
```

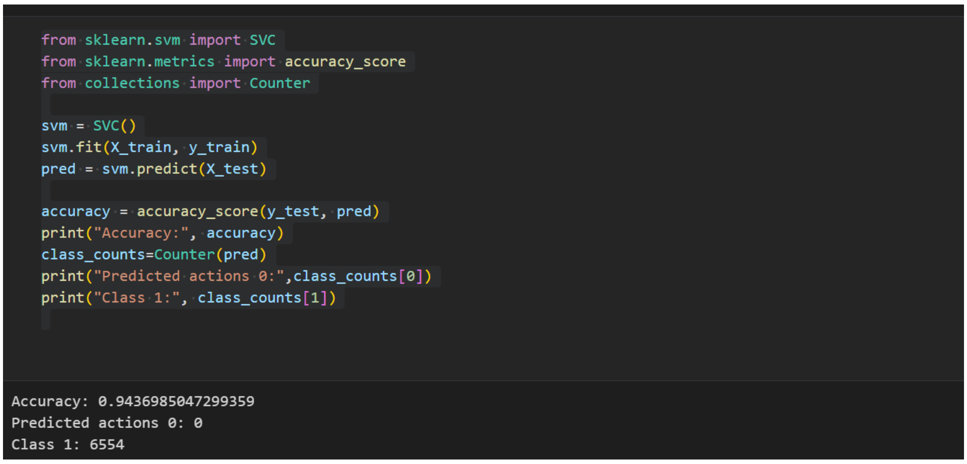
```
KNN Classifier Accuracy: 94.04943545926152
Predicted actions: Counter({1: 6443, 0: 111})
Prediction for new state using KNN classifier: [1]
```

Figure 5.11: Accuracy of KNN

5.4.2 SVM

This algorithm is used for binary classification and works well with high-dimensional datasets. This may not be helpful for the skewed dataset. Given a set of labeled training data, SVM determines the best hyperplane that divides the data into two classes with the greatest margin.[8]

The support vectors, or data points that are closest to the decision boundary, define the hyperplane. SVM optimizes the margin, or the distance



```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from collections import Counter

svm = SVC()
svm.fit(X_train, y_train)
pred = svm.predict(X_test)

accuracy = accuracy_score(y_test, pred)
print("Accuracy:", accuracy)
class_counts=Counter(pred)
print("Predicted actions 0:",class_counts[0])
print("Class 1:", class_counts[1])
```

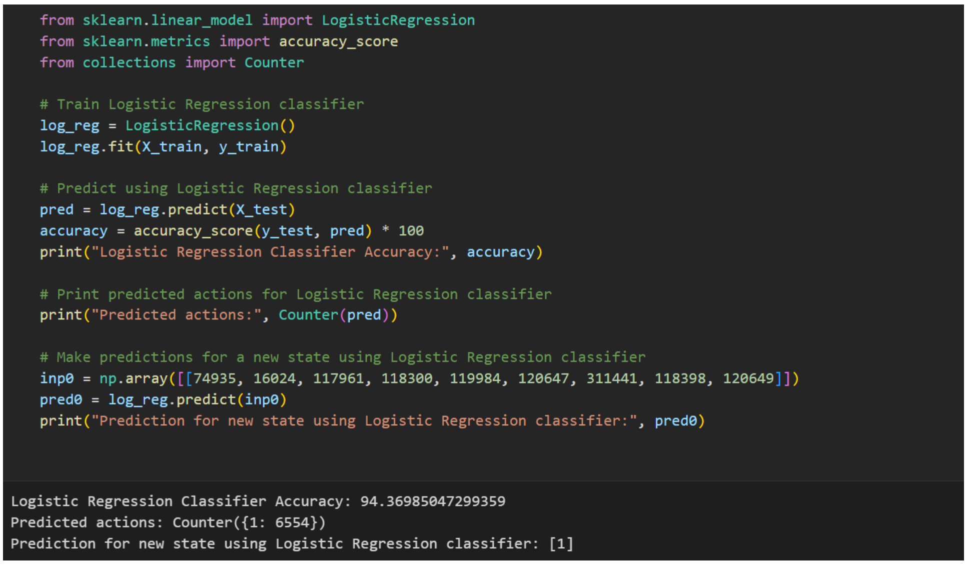
```
Accuracy: 0.9436985047299359
Predicted actions 0: 0
Class 1: 6554
```

Figure 5.12: Accuracy of SVM

between the hyperplane and the nearest data points (support vectors).

5.4.3 Logistic regression

Logistic regression is a statistical approach for estimating the likelihood of a binary outcome. It is a sort of regression analysis in which the dependent variable is categorical. The name ”logistic” is employed because this sort of regression employs the logistic function to simulate the likelihood of a specific class or event occurring.



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from collections import Counter

# Train Logistic Regression classifier
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)

# Predict using Logistic Regression classifier
pred = log_reg.predict(X_test)
accuracy = accuracy_score(y_test, pred) * 100
print("Logistic Regression Classifier Accuracy:", accuracy)

# Print predicted actions for Logistic Regression classifier
print("Predicted actions:", Counter(pred))

# Make predictions for a new state using Logistic Regression classifier
inp0 = np.array([[74935, 16024, 117961, 118300, 119984, 120647, 311441, 118398, 120649]])
pred0 = log_reg.predict(inp0)
print("Prediction for new state using Logistic Regression classifier:", pred0)
```

```
Logistic Regression Classifier Accuracy: 94.36985047299359
Predicted actions: Counter({1: 6554})
Prediction for new state using Logistic Regression classifier: [1]
```

Figure 5.13: Accuracy of Logistic regression

5.4.4 Randomforest

Random Forest is a frequent machine learning technique for classification and regression problems. It is an ensemble learning approach that works by building a large number of decision trees during training and then outputs the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

5.5 FEATURE ENGINEERING COMPARISONS FOR EACH MODEL

Let us compare the models with each feature engineering to choose which is best for this project by testing AUC score.

KNN	Test AUC Score
KNN	0.6814
KNN using (one hot encoding)	0.8172
KNN using frequency coding	0.7912
KNN using SVD	0.821
KNN using response coding	0.8219

Table 5.1: KNN Test AUC Score table.

Random forest	Test AUC Score
Random forest	0.876
Random forest using (one hot encoding)	0.85072
Random forest using frequency coding	0.88561
Random forest using SVD	0.86712
Random forest using response coding	0.80399

Table 5.2: Random forest Test AUC Score table.

As seen in the tables, One-Hot Encoding (OHE) outperforms all other models tested. As a result, we have opted to use OHE as a feature engineering approach in our research. OHE is successful in transforming categorical data into a format suited for machine learning algorithms, thus improving the performance and accuracy of our model for employee access control.

Logistic Regression	Test AUC Score
Logistic Regression	0.5303
Logistic Regression(one hot encoding)	0.8816
Logistic Regression using frequency coding	0.5917221
Logistic Regression SVD	0.634669
Logistic Regression using response coding	0.84046

Table 5.3: Logistic Regression Test AUC Score table.

SVM	Test AUC Score
SVM	0.496
SVM(one hot encoding)	0.8786
SVM using frequency coding	0.5957
SVM using SVD	0.63795
SVM response coding	0.8392

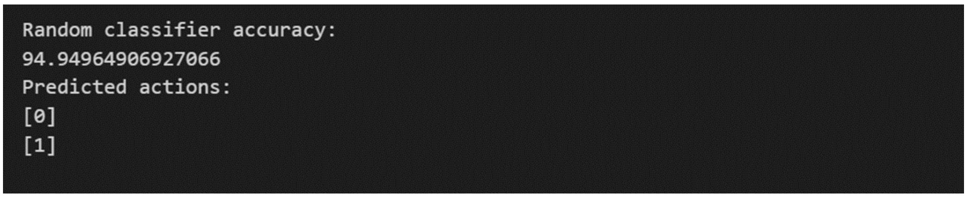
Table 5.4: SVM Test AUC Score table.

5.6 MODEL COMPARISION

In this section, we compare machine-learning models for employee access control with q-learning models. Our research aims to build on this by doing a comparative analysis. By doing so, we want to increase our understanding of employee access management by utilizing deep learning approaches.

5.6.1 Random forest model

The random forest model was chosen for its capacity to handle complicated patterns in data. The random forest classifier was used for this task, and it achieved a 94.96 percent accuracy rate. Cross-validation was used to undertake a rigorous evaluation of the model's performance. This examination includes calculating accuracy, precision, recall, and F1 scores. The model's hyperparameters were improved using RandomizedSearchCV and a KNN classifier. This technique assisted in identifying the ideal hyperparameters for the model, resulting in optimal performance.The output models were rigorously tested to show their accuracy in categorizing activities in the challenge.



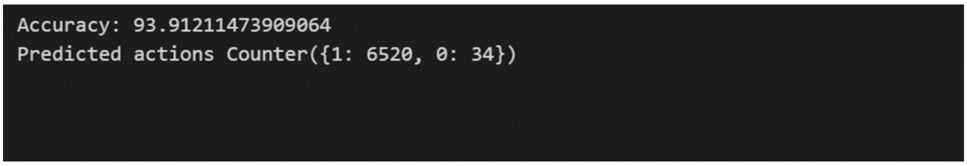
```
Random classifier accuracy:
94.94964906927066
Predicted actions:
[0]
[1]
```

5.6.2 q learning

Q-learning is the reinforcement learning model utilized in this research. One-hot encoding is utilized as a feature selection approach. The Q-learning agent is set up and taught to find the best policy for a binary classification problem.[9]

It maintains a Q-table, with each row representing a state and each column representing a possible action. During training, the agent interacts with the environment by monitoring states and performing actions. In this scenario, each data point in the training set represents a state, and the agent's task is to determine whether the relevant instance belongs to class 0 or class 1.

The Q-value for the current state-action combination is updated using the Q-learning equation, which includes the reward, the maximum Q-value for the next state, and a learning rate. Over time, the agent develops an optimum classification policy, allowing it to forecast the appropriate action (class) for a given condition (input characteristics) to maximize its cumulative reward. However, this approach achieves poorer accuracy than the machine learning model. Despite this, the agent can make judgments based on input attributes and anticipate the right behaviors with some degree of accuracy.[10]



```
Accuracy: 93.91211473909064
Predicted actions Counter({1: 6520, 0: 34})
```

Chapter 6

SOFTWARE TOOLS USED

The machine learning project on employee access control employed the following software tools:

1. Python is the major programming language used for data preparation, model creation, and assessment.

2. **Scikit-learn**: We used Scikit-learn, a Python machine learning framework, to create models like Random Forest Classifiers.

3. **OpenAI Gym**: The Q-learning model was implemented using OpenAI Gym, a tool-set for designing and evaluating reinforcement learning algorithms.

4. **Pandas**: Data was pre-processed and manipulated using Pandas, a Python data manipulation and analysis tool.

5. **NumPy**: Numerical calculations and array manipulation were performed using NumPy, a Python-based numerical computing package.

6. **Matplotlib and Seaborn** : These were utilized for data visualization.Generates plots to the analysis of model performance.

7. Visual Studio Code is used as the primary integrated development environment (IDE) for testing, and debugging the Python code for this project .

Chapter 7

RESULTS & DISCUSSION

The comparison of supervised learning model reveals heterogeneity in their performance, as seen by variances in accuracy. Random Forest predicts more reliably than [insert name of the other model]. This shows that Random Forest, particularly in binary classification issues, is a good fit for machine learning applications. While Q-learning falls within the category of reinforcement learning, its effectiveness in decision-making processes is presently being tested used to learn the optimal policy for making binary decisions. An overall investigation was undertaken to get findings regarding both algorithms.

7.1 DISCUSSION

This paper aimed to assess the performance of two binary classification models, Q-learning and Random Forest Classifier, for employee access management. The Q-learning model optimizes access policies based on staff traits and historical data. It learnt to make judgments by maximizing the cumulative benefits acquired over time. The Random Forest Classifier was trained on a single dataset with employee qualities as features and access rights as the target variable. It learned to classify employees into two categories: granted and denied access.

Chapter 8

CONCLUSION

After extensive experimentation and study, the following results were attained:The performance of both models was compared using measures including accuracy, precision, recall, and F1-score. When compared to the Q-learning model, the Random Forest Classifier outperformed the latter in terms of accuracy and F1. The Random Forest Classifier outperformed the Q-learning model in terms of prediction consistency and reliability. This implies that for the specific job of employee access control, the Random Forest Classifier surpasses Q-learning in terms of binary classification accuracy. Future Considerations: While the Random Forest Classifier performed best in this study, Q-learning's reinforcement learning technique holds potential in scenarios requiring sequential decision-making and exploration.

REFERENCES

[1] **Kaggle Amazon Employee Access Challenge in Kaggle.** https://www.kaggle.com/c/amazon-employee-access-challenge/

[2] **Toward Deep Learning Based Access Control** https://arxiv.org/pdf/2203.15124

[3] **Amazon Employess Access challenege** https://medium.com/analytics-vidhya/amazon-employee-access-challenge-e23fa2a1ddc2

[4] **Implemenetation Refernce of Machine learning project** https://github.com/Shriram016/Amazon-Employee-access-challenge

[5] **Complete guide for Feature enigneering with detailed** https://www.analyticsvidhya.com/blog/2021/09/complete-guide-to-feature-engineering-zero-to-hero

[6] **One hot encoding**

https://www.analyticsvidhya.com/blog/2023/12/how-to-do-one-hot-encoding/

[7] **KNeighborsClassifier**

https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[8] **Support vector machine** https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[9] **Reinforcement learning**

https://www.analyticsvidhya.com/blog/2021/02/introduction-to-reinforcement-learning-for-beginners/

[10] **Q learning**

https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning