



UNIVERSITY SCHOOL OF AUTOMATION AND ROBOTICS
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
EAST DELHI CAMPUS, SURAJMAL VIHAR, DELHI- 110032

Summer Training Report

On

Machine Learning in Admission Prediction Analysis

Submitted in partial fulfilment of the requirements for the completion of one month's
summer internship/training [ART 355/ART 457]



Name: Pransh Taank

Enrolment Number: 05519051622

Under the supervision of

Dr. Nanhey Singh, Scientist 'E', ISSPL, DRDO

Summer Training Certificate

ISO 9001:2015 Certified	363 File No. 1805...../HR/SSPL/24 Dated03-Sep-2024
सोलाहसत प्रयोगशाला, दिल्ली Solid State Physics Laboratory, Delhi	
	
(Min. of Defence, DRDO) Lucknow Road, Timarpur, Delhi-110054	
प्रशिक्षण प्रमाणपत्र / TRAINING CERTIFICATE	
This is to certify that Mr./Ms./ <u>PRANSH TANK</u>	
Student of <u>Guru Gobind Singh Indraprastha University, EDC</u> Roll No. <u>05519051622</u>	
Branch <u>B.Tech AI and Machine Learning</u> has completed successfully Summer/Winter Internship for	
the period from <u>2nd July 2024</u> to <u>16th Aug 2024</u> Duration <u>(06)</u> Six weeks / months.	
Topic of Internship was <u>Machine Learning in Admission Prediction</u>	
<u>Analysis</u>	
During the training period his/her conduct at SSPL was good.	
 राम आशीष चौकरी / Ram Ashish Choudhary Head-SSPL SSPL - DRDO सहायक निदेशक / Asst. Dir. / Scientist 'F' सोलाहसत प्रयोगशाला / Solid State Physics Laboratory लखनऊ रोड, टिमरपुर, दिल्ली-110054	

DECLARATION

I hereby declare that the Summer Training Report entitled “Machine Learning in Admission Prediction Analysis” is an authentic record of work completed as requirements of Summer Training (ART 355) during the period from 1st July 2024 to 16th August 2024 at Indian Solid State Physics Laboratory (ISSPL), Defense Research and Development Organization (DRDO), Delhi, New Delhi under the supervision of Dr. Nanhey Singh, Scientist 'E' of this establishment.

Date: 16th August 2024

(Signature of student)

Pransh Taank

(Name of Student)

05519051622

(Enrolment Number)

Date: 16th August 2024

(Signature of Supervisor)

Dr. Nanhey Singh, Scientist 'E'

(Name of Supervisor)

ACKNOWLEDGEMENT

The success and final outcome of learning Machine Learning required a lot of guidance and assistance from many people, and I am extremely privileged to have received this support throughout the completion of my course and several projects. All that I have achieved is due to such supervision and assistance, and I would not forget to thank them.

I respect and thank **DRDO (Defence Research and Development Organisation)** for providing me with the opportunity to pursue the course and project work, giving me all the support and guidance that enabled me to complete the course successfully. I am extremely thankful to the course advisor **Dr. Nanhey Singh, Scientist 'E', ISSPL, DRDO.**

I am thankful for and fortunate enough to have received constant encouragement, support, and guidance from all the teaching staff of **DRDO (Defence Research and Development Organisation),** which helped me in successfully completing my course and project work.

About Organization



The Defence Research and Development Organisation (DRDO) is India's premier agency responsible for the research, development, and production of cutting-edge technology to enhance national defence and security. **Founded in 1958**, DRDO operates under the Ministry of Defence and plays a critical role in making India self-reliant in defence technologies, reducing dependence on foreign imports.

Role:

DRDO's primary role is to design and develop advanced defence systems that meet the needs of the Indian Armed Forces. With over 50 laboratories across the country, DRDO specializes in various fields, including aeronautics, armaments, electronics, and missile systems, providing critical technologies that strengthen India's defence capabilities.

Achievements:

DRDO has made remarkable contributions to India's defence, including the development of ballistic missiles like Agni and Prithvi, the Tejas fighter aircraft, and the Arjun main battle tank. These achievements have not only modernized the military but have also positioned India as a leader in defence technology. Additionally, DRDO's work benefits civilian sectors, contributing to national development beyond the military sphere.

Impact:

DRDO's innovations have significantly enhanced India's self-reliance and strategic autonomy. By reducing dependence on foreign imports, DRDO has saved valuable resources and promoted domestic technological growth, making a lasting impact on the nation's defence and development.

Future Goals:

Looking ahead, DRDO aims to continue advancing next-generation technologies in areas like artificial intelligence, cyber defence, and hypersonic systems. The organization is committed to achieving full self-reliance in defence technology while fostering collaboration with academia, industry, and international partners to keep India at the forefront of global defence innovation.

CONTENTS

No.	Contents	Page no.
1	Summer Internship Certificate	i
2	Declaration	ii
3	Acknowledgement	iii
4	About Organization	iv

CHAPTERS

No.	Chapters	Page no.
1	Abstract	
2	Introduction	
3	Literature Survey	
4	Problem Statement	
5	Description of various Training Module	
6	Methodology Adopted	
7	Sample Data	
8	Code For APA	
9	Results & Discussions	
10	Conclusions	
11	References/Bibliography	

ABSTRACT

In recent years, there has been a noticeable trend of students seeking educational opportunities abroad, with the United States, Canada, Ireland, and Germany emerging as popular destinations. Among these international students, a substantial portion comes from India and China, particularly in the U.S., where the influx of Indian students pursuing postgraduate degrees has risen dramatically over the last decade.

As more students aim for higher education in these countries, the competition for securing a spot in prestigious universities has become increasingly fierce. Given this competitive landscape, it's crucial for students to have a way to assess their chances of admission in advance.

This research is focused on leveraging machine learning models to predict a student's likelihood of being accepted into a master's program, based on various factors. By providing insights into their admission prospects, this tool can help students make more informed decisions about where to apply and how to strengthen their applications.

INTRODUCTION

The landscape of higher education has become increasingly competitive, with students from around the globe vying for limited spots in prestigious universities. As the number of applicants continues to grow, particularly in popular destinations like the United States, Canada, Ireland, and Germany, the admission process has become more rigorous and selective. For international students, especially those from countries like India and China, navigating this competitive environment can be daunting, with many aspiring to secure a place in top-ranked institutions.

In this context, the ability to predict the likelihood of admission has emerged as a valuable tool for both students and educational institutions. Traditionally, admissions decisions have been based on a combination of quantitative and qualitative factors, including academic performance, standardized test scores, research experience, and personal statements. However, these criteria are often evaluated subjectively, leading to uncertainty and anxiety among applicants.

Machine Learning (ML), a subset of artificial intelligence, offers a promising solution to this challenge. By leveraging historical data and advanced algorithms, ML models can analyse a variety of factors to predict a student's chances of being admitted to a specific program. These models can process large datasets and uncover complex patterns that may not be immediately apparent to human evaluators. As a result, they provide a more data-driven approach to the admission process, helping students to better understand their strengths and weaknesses, and allowing universities to identify the most suitable candidates.

This research focuses on the development and application of machine learning models to predict admission outcomes for students applying to master's programs. By analysing key factors such as undergraduate GPA, GRE scores, research experience, and the ranking of the target university, the models aim to estimate the likelihood of admission with a high degree of accuracy. The predictions generated by these models can serve as a valuable resource for students, enabling them to make informed decisions about where to apply and how to improve their application profiles.

The introduction of ML into the admission prediction process represents a significant shift from traditional methods, offering the potential for greater transparency, efficiency, and fairness. As this technology continues to evolve, it could play a pivotal role in shaping the future of university admissions, making it more accessible and equitable for students worldwide.

LITERATURE SURVEY

The application of machine learning (ML) in educational settings, particularly in the context of university admissions, has garnered significant attention over the past decade. This literature survey provides an overview of key studies and approaches that have explored the use of ML models for predicting student admissions. The survey covers various methodologies, datasets, and outcomes, highlighting the evolution of predictive analytics in education and identifying gaps that this research aims to address.

1. Evolution of Admission Prediction Models

The concept of using statistical models to predict student success and admissions is not new. Early studies primarily relied on traditional statistical techniques such as logistic regression and linear regression. These methods were used to identify correlations between admission outcomes and factors like GPA, standardized test scores, and demographic information.

- **Murtaugh et al. (1999)** conducted one of the earliest studies in this domain, employing logistic regression to analyse factors influencing student retention and graduation. Although their focus was not directly on admissions, their work laid the groundwork for understanding the predictive power of academic indicators.

As machine learning techniques became more accessible, researchers began to explore more complex models that could capture non-linear relationships and interactions between variables.

- **Nguyen et al. (2015)** applied a decision tree model to predict the likelihood of student dropout in higher education institutions. This study demonstrated the potential of decision trees in educational prediction tasks but was more focused on retention rather than admission.
- **Kabakchieva (2013)** used data mining techniques, including decision trees and k-nearest neighbours, to predict student performance and admission outcomes. This research highlighted the advantages of ML over traditional statistical methods, particularly in handling large datasets with multiple variables.

2. Machine Learning in Admission Prediction

The application of ML specifically to predict university admissions gained momentum as educational institutions began to accumulate large amounts of data on applicants. Researchers have explored various ML models to improve the accuracy and reliability of admission predictions.

- **Zhang and Rangwala (2017)** conducted a comprehensive study using machine learning to predict graduate school admission outcomes. They compared several algorithms, including logistic regression, support vector machines (SVM), and random forests, to determine which provided the most accurate predictions. Their findings indicated that ensemble methods like random forests generally outperformed simpler models in terms of prediction accuracy.
- **Marquez-Vera et al. (2013)** utilized a hybrid approach combining decision trees with fuzzy logic to predict student success in undergraduate programs. Their model considered both academic and socio-economic factors, demonstrating the importance of a holistic approach to prediction.
- **Sabbah et al. (2014)** explored the use of artificial neural networks (ANN) to predict admission outcomes based on a wide range of factors, including GPA, GRE scores, and letters of recommendation. Their study showed that ANN models could effectively capture complex patterns in the data, leading to high prediction accuracy. However, they also noted the challenge of interpretability with such models.

3. Factors Influencing Admission Predictions

Several studies have focused on identifying and analysing the factors that most significantly influence admission decisions. These factors often vary by institution and program, but common predictors include standardized test scores, undergraduate GPA, and research experience.

- **Kuncel et al. (2001)** conducted a meta-analysis of the predictive validity of the Graduate Record Examinations (GRE) for graduate school performance. Their findings supported the use of GRE scores as a significant predictor of academic success in graduate programs, although the strength of this predictor varied across disciplines.
- **Walpole et al. (2005)** examined the impact of socio-economic factors on graduate admissions, revealing that students from higher socio-economic backgrounds tended to have better access to resources that could enhance their application profiles, such as research

opportunities and test preparation services. This study highlighted the potential for bias in the admission process, which ML models need to account for.

- **Luo et al. (2020)** investigated the role of research experience in predicting graduate admissions. Their study found that students with prior research experience were more likely to be admitted, particularly in STEM fields. This finding underscores the importance of including research experience as a feature in ML models for admission prediction.

4. Challenges and Limitations of Existing Models

While ML models have shown promise in predicting admissions, several challenges and limitations have been identified in the literature.

- **Bias and Fairness:** Many studies have pointed out the risk of bias in ML models, particularly when the training data reflects historical inequalities. **Hardt et al. (2016)** discussed the importance of fairness in ML, proposing methods to mitigate bias in predictive models. This issue is particularly relevant in admissions, where biased predictions could reinforce existing disparities.
- **Interpretability:** As noted by **Lipton (2018)**, one of the major challenges with advanced ML models, particularly deep learning techniques, is the lack of interpretability. In the context of admissions, where decisions need to be transparent and justifiable, the "black box" nature of some ML models can be problematic. This has led to a preference for simpler models like decision trees and logistic regression in some cases, despite their lower accuracy compared to more complex models.
- **Data Quality and Availability:** The effectiveness of ML models depends heavily on the quality and quantity of data available. **Rosenberg et al. (2018)** highlighted the challenges of data quality in educational settings, including issues like missing data, inconsistencies in data collection, and the variability of factors across institutions. This limits the generalizability of models trained on data from a single institution or program.

5. Gaps in the Literature

Despite the advances in ML for admission prediction, several gaps remain that this research aims to address:

- **Incorporation of Qualitative Data:** Most existing models rely heavily on quantitative data, such as GPA and test scores. However, qualitative factors like personal statements and letters

of recommendation also play a crucial role in admissions. Integrating these factors into predictive models remains a challenge due to the complexity of natural language processing (NLP) in educational contexts.

- **Real-Time Applications:** While many studies have developed predictive models, fewer have focused on deploying these models in real-time applications that students can use during the application process. This research seeks to bridge that gap by not only developing a predictive model but also implementing it in a web-based tool accessible to students.
- **Customization for Different Institutions:** Many studies have focused on a single institution or program, limiting the applicability of their models to other contexts. There is a need for more research on developing customizable models that can be adapted to different universities and programs.

PROBLEM STATEMENT

- **Increasing Competition:** The global competition for admission to prestigious graduate programs has intensified, making it difficult for applicants to evaluate their chances of acceptance.
- **Challenges for International Students:** International students, especially from countries like India and China, face uncertainty and challenges in assessing their likelihood of being admitted to their preferred universities.
- **Subjectivity in Traditional Assessments:** Traditional methods of assessing admission prospects are often subjective, leading to unclear guidance for students on how to improve their application profiles.
- **Inconsistent Admission Decisions:** Universities struggle with processing a growing number of diverse applications, which can result in inconsistent and potentially biased admission decisions.
- **Need for Data-Driven Solutions:** There is a need for a more objective, data-driven approach to predict admission outcomes accurately and consistently.
- **Assisting Students and Universities:** A reliable predictive model would help students make informed decisions about where to apply and assist universities in identifying the most suitable candidates for their programs.
- **Development of a Predictive Model:** The research aims to develop a machine learning-based model that analyses factors such as GPA, GRE scores, research experience, and university ranking to predict a student's likelihood of admission.
- **Improving Application Success:** The goal is to create an accessible tool that provides actionable insights, helping students enhance their applications and increasing their chances of successful admission.

TRAINING MODULE

MODULES

1. Linear Regression
2. Random Forest
3. Decision Tree
4. Support Vector Machine (SVM)

DESCRIPTION

Linear Regression:

- Simple and Interpretable: Linear regression is a basic yet powerful model that establishes a relationship between input variables (e.g., GPA, GRE scores) and a continuous output (e.g., admission probability).
- Assumes Linearity: It assumes a linear relationship between the predictors and the outcome, making it less effective when dealing with complex, non-linear patterns.
- Easy to Implement: The model is straightforward to implement and interpret, providing clear insights into how each factor influences the admission outcome.
- Limitations: It may struggle with overfitting when there are too many predictors or if the data has significant outliers.

Random Forest:

- Ensemble Method: Random forest is an ensemble learning technique that constructs multiple decision trees during training and outputs the mode of the classes (for classification) or mean prediction (for regression) of the individual trees.
- Handles Non-Linearity and Interactions: It effectively captures non-linear relationships and interactions between features, making it suitable for complex datasets.
- Robustness: The model is robust against overfitting, especially when dealing with high-dimensional data, and can handle missing values and noisy data well.
- Feature Importance: Random forest provides a measure of feature importance, helping to identify which factors contribute most to the prediction.

Decision Tree:

- **Tree-Based Structure:** Decision trees divide the data into branches based on feature values, leading to a decision node or leaf that represents an outcome (e.g., admission or rejection).
- **Easy to Understand:** The model is intuitive and easy to visualize, making it accessible to non-experts.
- **Flexibility:** Decision trees can handle both numerical and categorical data and do not require data normalization or scaling.
- **Overfitting Risk:** They can be prone to overfitting, particularly when the tree becomes too complex, capturing noise in the data rather than the underlying pattern.

Support Vector Machine (SVM):

- **Maximizes Margin:** SVM is a powerful classification model that works by finding the hyperplane that best separates the data into different classes, maximizing the margin between the nearest points of each class.
- **Effective in High Dimensions:** SVM is particularly effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples.
- **Kernel Trick:** It uses the kernel trick to handle non-linear data by transforming it into a higher-dimensional space where a linear separator can be found.
- **Computationally Intensive:** SVM can be computationally expensive, especially with large datasets, and the choice of the kernel and regularization parameter significantly affects performance.

METHODOLOGY ADOPTED

1. Linear Regression

Overview: Linear Regression is a fundamental statistical method that models the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship and is often used for predictive analysis when the outcome is continuous.

Steps:

- **Data Preprocessing:** Begin by cleaning the dataset, handling missing values, and standardizing the features if necessary.
- **Feature Selection:** Select relevant features (e.g., GRE score, GPA, Research experience, etc.) that are expected to influence the admission outcome.
- **Model Training:** Fit the linear regression model to the training data using the ordinary least squares (OLS) method to minimize the difference between observed and predicted values.
- **Prediction:** Use the trained model to predict the probability of admission for new applicants.
- **Evaluation:** Evaluate model performance using metrics such as Mean Squared Error (MSE), R-squared, and residual analysis to assess the accuracy and goodness-of-fit of the model.

Limitations:

- **Assumes Linearity:** May not capture complex, non-linear relationships between features.
- **Sensitive to Outliers:** The presence of outliers can significantly impact the model's predictions.

2. Decision Tree

Overview: A Decision Tree is a tree-like model of decisions and their possible consequences. It splits the dataset into subsets based on the value of input features, making it intuitive and easy to interpret.

Steps:

- **Data Preprocessing:** Clean the data and handle missing values. No need for feature scaling.

- **Tree Construction:** Build the decision tree by recursively splitting the data based on the feature that provides the maximum information gain (or equivalently, the minimum Gini impurity or entropy).
- **Pruning:** To prevent overfitting, apply pruning techniques to remove branches that have little importance or are too specific to the training data.
- **Prediction:** Classify new applicants by passing their feature values through the tree and following the branches that correspond to their values.
- **Evaluation:** Use accuracy, precision, recall, and the confusion matrix to assess the model's performance.

Limitations:

- **Overfitting:** Decision trees can overfit, especially when the tree is too deep.
- **Instability:** Small changes in the data can lead to a completely different tree structure.

3. Random Forest

Overview: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees.

Steps:

- **Data Preprocessing:** Similar to decision trees, Random Forest does not require feature scaling.
- **Tree Construction:** Generate a large number of decision trees, each trained on a random subset of the data and features (bagging). This helps in reducing variance and improving generalization.
- **Ensemble Learning:** Aggregate the predictions of all the individual trees to produce a final prediction, which is more robust and less prone to overfitting.
- **Feature Importance:** Extract feature importance scores to understand which features have the most influence on the admission prediction.
- **Evaluation:** Use metrics like accuracy, confusion matrix, and classification report to evaluate the overall model performance.

Limitations:

- Complexity: More computationally intensive than a single decision tree.
- Interpretability: The final model is less interpretable compared to a single decision tree.

4. Support Vector Machine (SVM)

Overview: Support Vector Machine (SVM) is a powerful classification model that aims to find the optimal hyperplane that separates data points of different classes with the maximum margin. It can handle both linear and non-linear classification tasks.

Steps:

- Data Preprocessing: Standardize the features to ensure that the model's convergence is efficient and the results are accurate.
- Kernel Trick: Choose an appropriate kernel function (e.g., linear, polynomial, RBF) to map the input features into a higher-dimensional space where they can be linearly separable.
- Model Training: Train the SVM model on the training data to find the hyperplane that maximizes the margin between classes. Regularization parameters (C) and kernel parameters (gamma) are tuned to avoid overfitting.
- Prediction: Classify new applicants by determining on which side of the hyperplane they fall.
- Evaluation: Assess the model using accuracy, confusion matrix, precision, recall, and the F1-score.

Limitations:

- Computationally Intensive: Especially with large datasets and when using complex kernel functions.
- Parameter Tuning: The performance heavily depends on the choice of parameters (C and gamma), which may require extensive tuning.

SAMPLE DATA

- Here is an image for sample data which was used in the following codes:



admission_data.csv

	A	B	C	D	E
1	GRE score	Research experience	GPA	Admission	Test score
2	337	1	9.65	1	0.92
3	324	1	8.87	1	0.76
4	316	0	8	1	0.72
5	322	0	8.67	1	0.8
6	314	1	8.21	0	0.65
7	330	0	9.34	1	0.9
8	321	1	8.2	1	0.75
9	308	1	7.9	0	0.68
10	302	1	8	0	0.5
11	323	0	8.6	0	0.45
12	325	0	8.4	1	0.52
13	327	1	9	1	0.84

CODE FOR APA (Admission Prediction Analysis)

1. CODE FOR LINEAR REGRESSION

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split

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

X = df[['GRE score', 'Research experience', 'GPA', 'Test score']]

y = df['Admission']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = np.mean((y_pred - y_test) ** 2)

print(f'Mean Squared Error: {mse:.2f}')

plt.scatter(y_test, y_pred)

plt.xlabel('Actual Admission')

plt.ylabel('Predicted Admission')

plt.title('Admission Prediction using Linear Regression')

plt.show()

gre_score = float(input('Enter GRE Score: '))

research_exp = float(input('Enter Research Experience (0 or 1) : '))

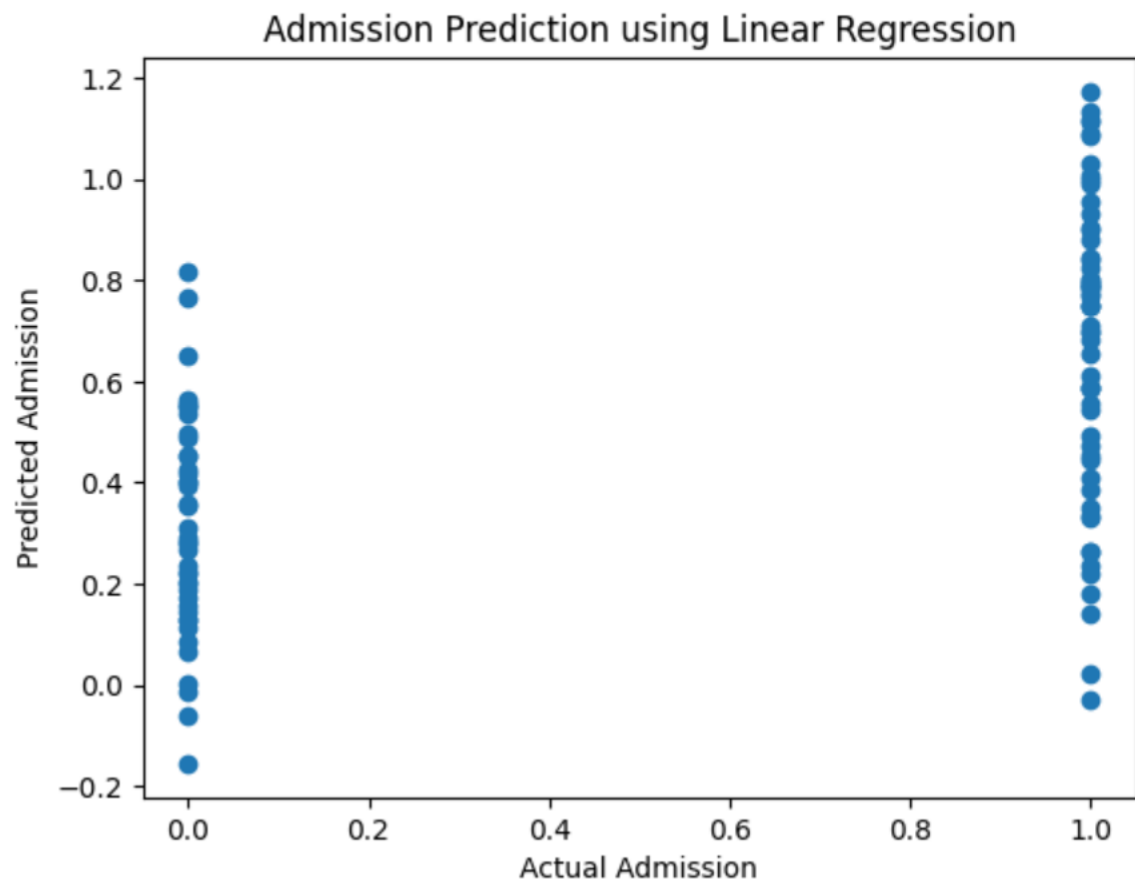
gpa = float(input('Enter GPA: '))

test_score = float(input('Enter Test Score: '))

new_data = pd.DataFrame({'GRE score': [gre_score], 'Research experience': [research_exp], 'GPA': [gpa], 'Test score': [test_score]})
```

```
admission_pred = model.predict(new_data)
```

```
print(f'Predicted Admission: {admission_pred[0]:.2f}')
```



2. CODE FOR RANDOM FOREST

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

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

X = df[['GRE score', 'Research experience', 'GPA', 'Test score']]

y = df['Admission']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')

print('Classification Report:')

print(classification_report(y_test, y_pred))

print('Confusion Matrix:')

print(confusion_matrix(y_test, y_pred))

importances = model.feature_importances_

indices = np.argsort(importances)[::-1]

plt.barh(range(len(indices)), importances[indices])

plt.yticks(range(len(indices)), X.columns[indices])

plt.xlabel('Feature Importance')

plt.ylabel('Features')

plt.title('Feature Importance in Random Forest')

plt.show()
```

```
gre_score = float(input('Enter GRE Score: '))

research_exp = float(input('Enter Research Experience (0 or 1) : '))

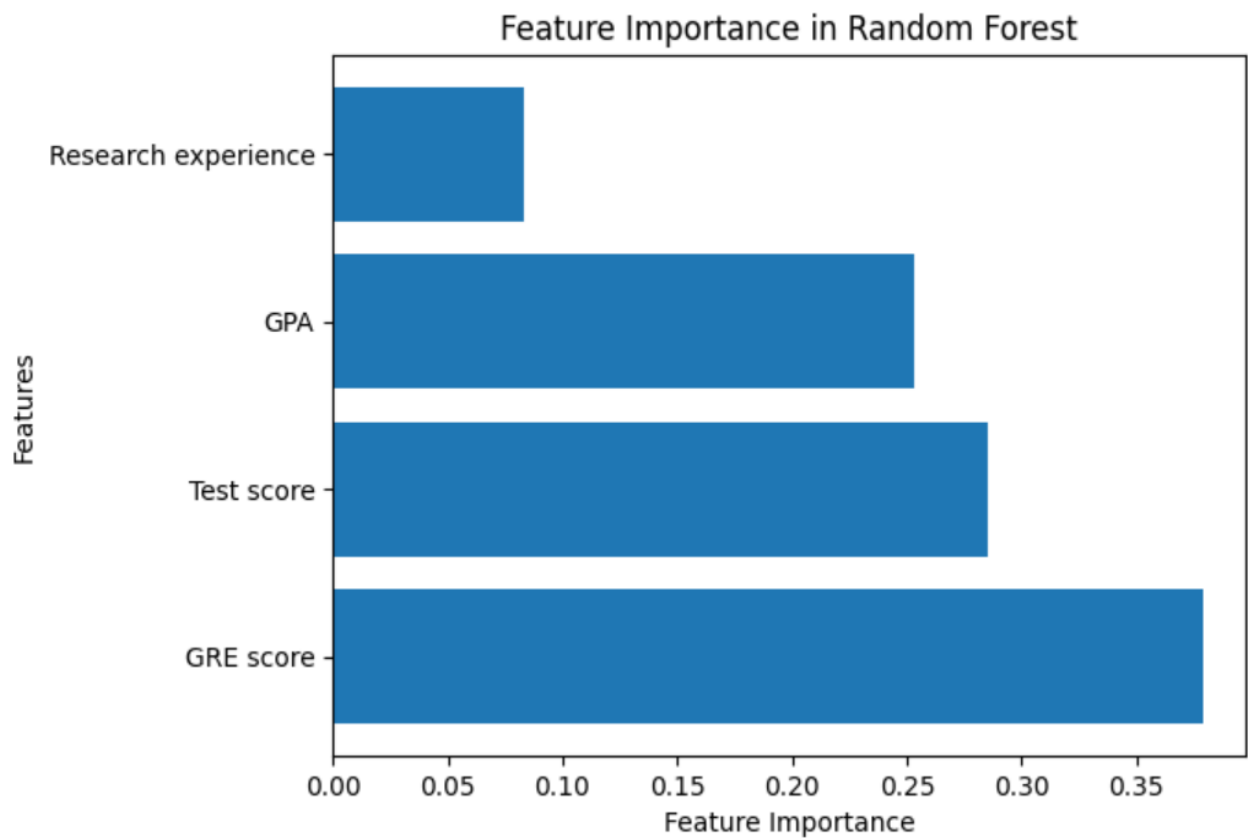
gpa = float(input('Enter GPA: '))

test_score = float(input('Enter Test Score: '))

new_data = pd.DataFrame({'GRE score': [gre_score], 'Research experience': [research_exp], 'GPA': [gpa], 'Test score': [test_score]})

admission_pred = model.predict(new_data)

print(f'Predicted Admission: {admission_pred[0]}')
```



3. CODE FOR RANDOM FOREST

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier, plot_tree

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

try:

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

except FileNotFoundError:

    print("Error: The file 'admission_data.csv' was not found. Please check the file path.")

    exit()

X = df[['GRE score', 'Research experience', 'GPA', 'Test score']]

y = df['Admission']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = DecisionTreeClassifier(random_state=42)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')

print('Classification Report:')

print(classification_report(y_test, y_pred))

print('Confusion Matrix:')

print(confusion_matrix(y_test, y_pred))

plt.figure(figsize=(10, 8))

plot_tree(model, feature_names=X.columns, class_names=['Not Admitted', 'Admitted'], filled=True)

plt.title('Decision Tree for Admission Prediction')

plt.show()
```



```

def get_float_input(prompt):
    while True:
        try:
            value = float(input(prompt))
            return value
        except ValueError:
            print("Invalid input. Please enter a numeric value.")

gre_score = get_float_input('Enter GRE Score: ')

research_exp = get_float_input('Enter Research Experience (0 or 1) : ')

gpa = get_float_input('Enter GPA: ')

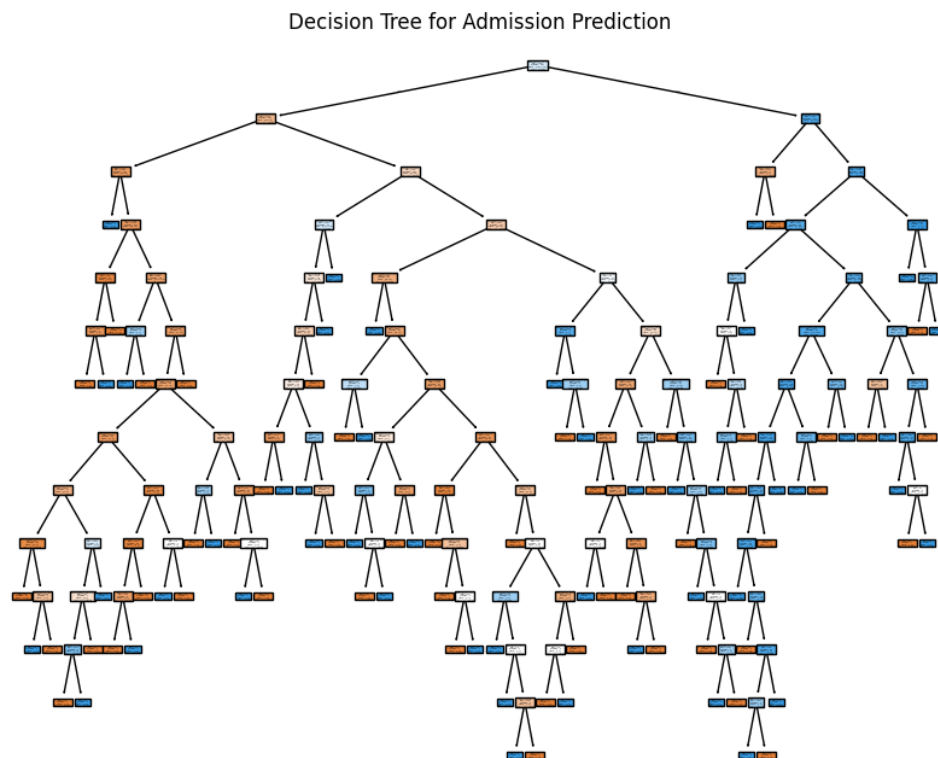
test_score = get_float_input('Enter Test Score: ')

new_data = pd.DataFrame({'GRE score': [gre_score], 'Research experience': [research_exp], 'GPA': [gpa], 'Test score': [test_score]})

admission_pred = model.predict(new_data)

print(f'Predicted Admission: {"Admitted" if admission_pred[0] else "Not Admitted"}')

```



4. CODE FOR SUPPORT VECTOR MACHINE (SVM)

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn import svm

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

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

X = df[['GRE score', 'Research experience', 'GPA', 'Test score']]

y = df['Admission']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = svm.SVC(kernel='rbf', C=1, gamma=0.1)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')

print('Classification Report:')

print(classification_report(y_test, y_pred))

print('Confusion Matrix:')

print(confusion_matrix(y_test, y_pred))

from sklearn.decomposition import PCA

pca = PCA(n_components=2)

X_pca = pca.fit_transform(X)

plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y)

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('Decision Boundary')

plt.show()
```

```

gre_score = float(input('Enter GRE Score: '))

research_exp = float(input('Enter Research Experience (0 or 1) : '))

gpa = float(input('Enter GPA: '))

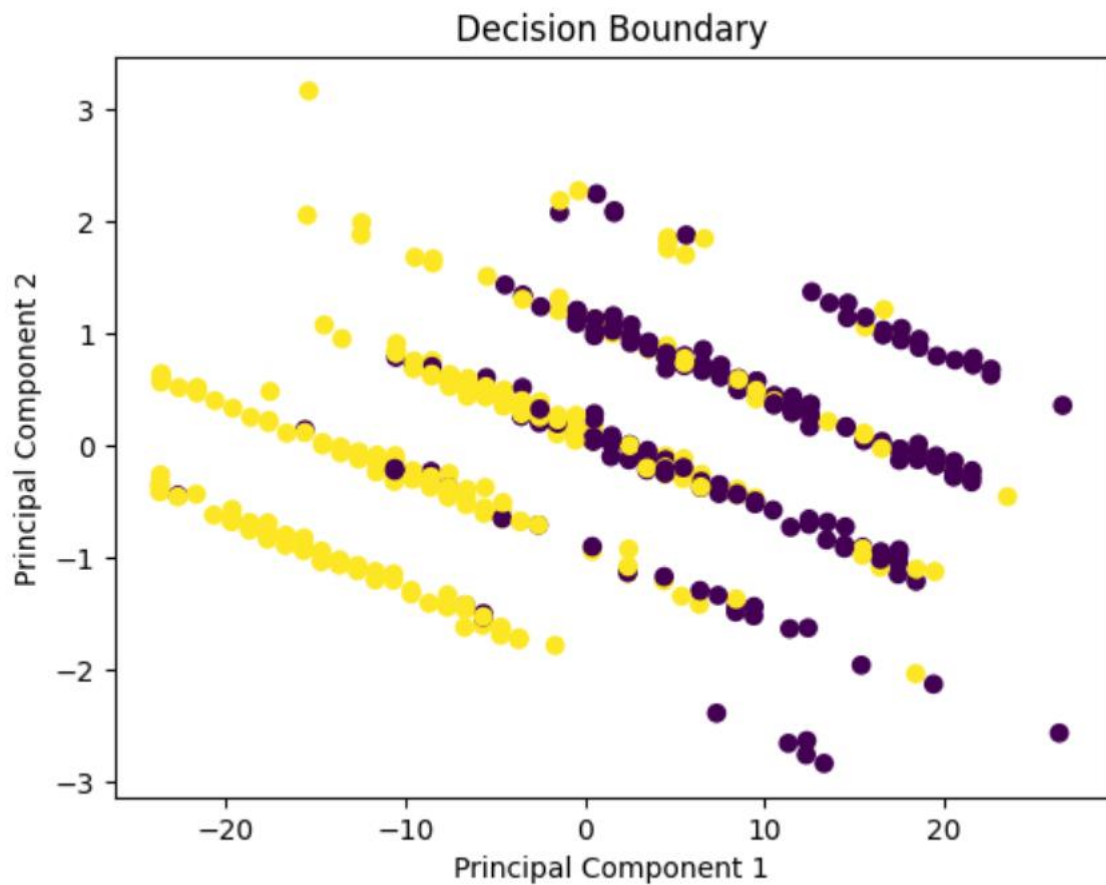
test_score = float(input('Enter Test Score: '))

new_data = pd.DataFrame({'GRE score': [gre_score], 'Research experience': [research_exp], 'GPA': [gpa], 'Test score': [test_score]})

admission_pred = model.predict(new_data)

print(f'Predicted Admission: {admission_pred[0]}')

```



RESULTS & DISCUSSIONS

In this study, four machine learning models—Linear Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM)—were developed and evaluated for their effectiveness in predicting student admission outcomes to U.S. universities. The results of these models were analyzed based on accuracy, interpretability, and computational efficiency

Model Performance Comparison

- **Linear Regression:**

- Accuracy: Linear Regression provided a satisfactory level of accuracy, especially considering its simplicity. It performed well in identifying trends between input features like GPA, GRE scores, and research experience, and the likelihood of admission.
- Interpretability: The model's coefficients were easily interpretable, allowing for straightforward understanding of how each feature impacted the admission decision. This transparency made it a strong candidate for system implementation, especially for non-technical users.
- Computational Efficiency: Linear Regression was highly efficient in terms of computation, requiring minimal processing time even with large datasets.

- **Decision Tree:**

- Accuracy: The Decision Tree model offered decent accuracy, but it tended to overfit the training data, especially with deeper trees. This overfitting resulted in reduced generalization to unseen data.
- Interpretability: While the Decision Tree model provided a clear visual representation of decision-making, the complexity increased with the depth of the tree, making it harder to interpret when the tree became too large.
- Computational Efficiency: The model was relatively quick to train and predict but was less efficient compared to Linear Regression, especially as the complexity of the tree increased.

- **Random Forest:**

- Accuracy: Random Forest demonstrated high accuracy by averaging the predictions of multiple decision trees. It effectively handled non-linear relationships and interactions between features, making it one of the most accurate models tested.
- Interpretability: Despite its accuracy, Random Forest lacked interpretability. While it provided insights into feature importance, the overall model was more of a "black box," making it less suitable for users who need to understand the decision-making process.
- Computational Efficiency: The model required more computational resources due to the ensemble of trees, which increased training time and complexity.

- **Support Vector Machine (SVM):**

- Accuracy: SVM showed strong performance in terms of classification accuracy, particularly when using non-linear kernels like the RBF kernel. However, the model's performance was highly sensitive to parameter tuning, which required significant experimentation.
- Interpretability: SVM is inherently less interpretable, especially with non-linear kernels, making it difficult for users to understand how specific decisions are made.
- Computational Efficiency: The computational demands were higher for SVM, especially with larger datasets and complex kernels, which made it less practical for real-time or user-facing applications.

Model Selection and Implementation

- Based on the comparative analysis, Linear Regression was selected as the most appropriate model for the system due to its balance of accuracy, interpretability, and computational efficiency. Although more sophisticated models like Random Forest and SVM offered higher accuracy, their complexity and lack of transparency were significant drawbacks, particularly for a system intended for non-technical users.
- The user interface was designed to be simple and intuitive, allowing students to easily input their academic and test scores and receive an immediate prediction of their admission chances. This interface is crucial for ensuring that the system can be used by a broad audience without requiring technical knowledge.

Impact and Benefits

The system effectively achieves its primary goals:

- **Time and Cost Savings:** Students can save time and money by focusing their efforts on universities where they have a higher likelihood of acceptance, reducing the need for expensive educational consultants and multiple application fees.
- **Informed Decision-Making:** The system provides students with actionable insights, helping them to make more informed decisions about where to apply based on their academic profile.
- **Accessibility:** The user-friendly design ensures that students from various backgrounds, regardless of their technical expertise, can utilize the system to assess their admission prospects.

Limitations and Future Work

While the Linear Regression model was chosen for its strengths, it does have limitations, particularly in capturing complex, non-linear relationships between variables. Future work could explore hybrid models or use ensemble techniques to combine the interpretability of Linear Regression with the accuracy of more complex models like Random Forest or SVM.

Moreover, expanding the system to include data from other countries and additional features (e.g., extracurricular activities, personal statements) could further enhance the accuracy and usefulness of the predictions.

CONCLUSIONS

The primary objective of this study was to develop a prototype system for students aspiring to pursue higher education in the United States. Various machine learning algorithms, including Linear Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM), were implemented and evaluated in this study. Among these, Linear Regression emerged as the most suitable model for system development, outperforming more complex models like Decision Tree, Random Forest, and SVM in terms of simplicity, interpretability, and applicability to the problem at hand.

This approach successfully meets its goals by enabling students to save both time and money that would otherwise be spent on educational consultants and application fees for universities where their chances of acceptance are lower. Additionally, the system empowers students to make more informed and quicker decisions regarding their university applications, thereby enhancing their overall application strategy.

REFERENCES/BIBLIOGRAPHY

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
 - A comprehensive resource on machine learning algorithms, including those used in this study such as Linear Regression, Decision Trees, Random Forests, and SVM.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
 - This paper introduces the Random Forest algorithm, discussing its theoretical foundations and practical applications in classification and regression tasks.
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273-297.
 - A seminal paper that presents the Support Vector Machine (SVM) algorithm, explaining the concepts of hyperplanes, margins, and kernel trick used in SVM.
- Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81-106.
 - The foundational work on Decision Trees, discussing how decision trees are built, pruned, and applied in various domains.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis*. John Wiley & Sons.
 - This book provides an in-depth exploration of Linear Regression analysis, focusing on model fitting, diagnostics, and interpretation.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
 - A comprehensive reference for various machine learning algorithms, including those evaluated in this study, with a focus on statistical learning theory.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
 - This book covers the practical aspects of applying machine learning algorithms, including model selection, evaluation, and deployment, which were essential in developing the prototype system.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.
 - A practical guide to machine learning, providing examples and case studies that align with the methods used in this study, including regression and classification techniques.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
 - An article describing the Scikit-learn library used for implementing the machine learning models in this study.
- Nau, R. (2020). *Linear Regression Models: Theory and Practice*.
 - Available at Duke University Online Course.

- An online resource that explains the theory and practice of Linear Regression, including assumptions, diagnostics, and interpretation.
- Biau, G. (2012). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 13, 1063-1095.
- Discusses the mathematical properties and practical performance of Random Forest models, with applications similar to those used in this study.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
- A comprehensive guide to data mining techniques, including those used in this study for predicting student admissions.
- Ho, T. K. (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1, 278-282.
- The original paper introducing Random Forests, describing the algorithm's robustness and application in various domains.
- Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320.
- Provides insights into regularization techniques that are crucial for improving the performance of machine learning models, including Linear Regression.
- Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.
- A detailed exposition of the theoretical foundations of Support Vector Machines, offering deep insights into the learning theory behind SVM.

APPRENTICE TRAINING REPORT

ON

“Machine Learning in Admission Prediction Analysis”

Submitted to

SOLID STATE PHYSICS LAB (SSPL)

Defence Research and Development Organisation



Under the guidance of
Dr. Nanhey Singh
Scientist 'E', SSPL, DRDO

Submitted By:
Pransh Taank