MODEL DEVELOPMENT REPORT

Project Overview

Objective

The objective of this project is to develop and evaluate various machine learning models for predicting student dropout risk. The models utilized include Decision Trees, Random Forests, Support Vector Machines (SVM), Gradient Boosting Machines (XGBoost), Logistic Regression, and a Deep Learning model implemented with PyTorch.

Dataset

Source: The dataset was gotten from the 3signet

Description: The dataset contains 37 features, which include both numerical and categorical data. After feature engineering, a total of 17 features were chosen for the model development to reduce redundant features.

Target Variable: The target variable is "Target" where the values indicate whether a student is likely to drop out, enrolled or graduated. Since our focus is on predicting the dropouts, the variable was encoded with Dropout being 1 and 0 for not being a Dropout.

Train-Test Split: The data was split into training (70%) and testing (30%) sets for model evaluation.

Model Development

1. Logistic Regression

Architecture

- Logistic Regression predicts the probability of a student dropping out based on their features using a logistic function.

Implementation

- Hyperparameters used:

Regularization: [`(C = 0.001,max_iter=300, multi_class = 'auto', penalty = None, solver = 'newton-cholesky'`

Performance Metrics

```
Confusion matrix:
 [[198  86]
 [ 36 565]]
Classification report:
              precision    recall  f1-score

           0       0.85      0.70      0.76
           1       0.87      0.94      0.90

    accuracy                           0.86
```

Shows a high performance since we are interested in the high dropout prediction.

Insights

- Logistic Regression provides interpretability, allowing us to understand how features influence dropout risk, though it may not capture non-linear relationships effectively.

---

2. Decision Trees

Architecture

- Decision Trees create a flowchart-like model where decisions based on features determine whether a student is at risk of dropping out.

Implementation

- Hyperparameters used:

```
DecisionTreeClassifier(
 max_features=None,
 max_depth= 4,
 criterion='log_loss',
 class_weight='balanced'
```

Performance Metrics

Insights

- This model is intuitive and easy to visualize but can overfit the training data.

```
Confusion matrix:
 [[223  61]
 [ 70 531]]
Classification report:
              precision    recall  f1-score

           0       0.76      0.79      0.77
           1       0.90      0.88      0.89

    accuracy                           0.85
```

---

3. Random Forests

Architecture

- Random Forests build multiple decision trees and aggregate their predictions to improve accuracy and robustness.

Implementation

- Hyperparameters used:

```
RandomForestClassifier (n_estimators= 300,
 max_features='sqrt',
 max_depth= 9,
 criterion='gini',
 class_weight='balanced')
```

Performance Metrics

Insights

- Random Forests help mitigate overfitting while improving performance, making them suitable for dropout risk prediction.

```
Confusion matrix:
 [[216  68]
 [ 49 552]]
Classification report:
              precision    recall  f1-score

           0       0.82      0.76      0.79
           1       0.89      0.92      0.90
```

. Support Vector Machines (SVM)

Architecture

- SVM finds the optimal hyperplane that separates students at risk of dropping out from those who are not.

Implementation

- Hyperparameters used:

```
SVC(kernel="linear", gamma='auto', C=2, random_state=42)
```

Performance Metrics

Insights

- SVMs can handle high-dimensional data effectively but may require careful tuning of hyperparameters.

```
Confusion matrix:
 [[189  95]
 [ 39 562]]
Classification report:
              precision    recall  f1-score

           0       0.83      0.67      0.74
           1       0.86      0.94      0.89
```

---

5. Gradient Boosting Classifier

Architecture

- GBC builds models sequentially, where each new tree corrects the errors of the previous ones, optimizing for dropout risk prediction.

Implementation

Performance Metrics

```
{'subsample': 0.1, 'n_estimators': 700, 'max_features': 'sqrt', 'max_depth': 1}
```

Insights

- XGBoost often performs exceptionally well, making it a strong candidate for predicting student dropout risk.

```
Confusion matrix:
 [[197  87]
 [ 38 563]]
Classification report:
              precision    recall  f1-score

           0       0.84      0.69      0.76
           1       0.87      0.94      0.90

    accuracy                           0.86
```

---

6. Deep Learning Model (PyTorch)

Architecture

- A neural network with an input layer, multiple hidden layers, and an output layer for predicting dropout risk. Activation functions (e.g., ReLU for hidden layers, Sigmoid for the output) introduce non-linearity.

python
Copy code
import torch
import torch.nn as nn

Insights

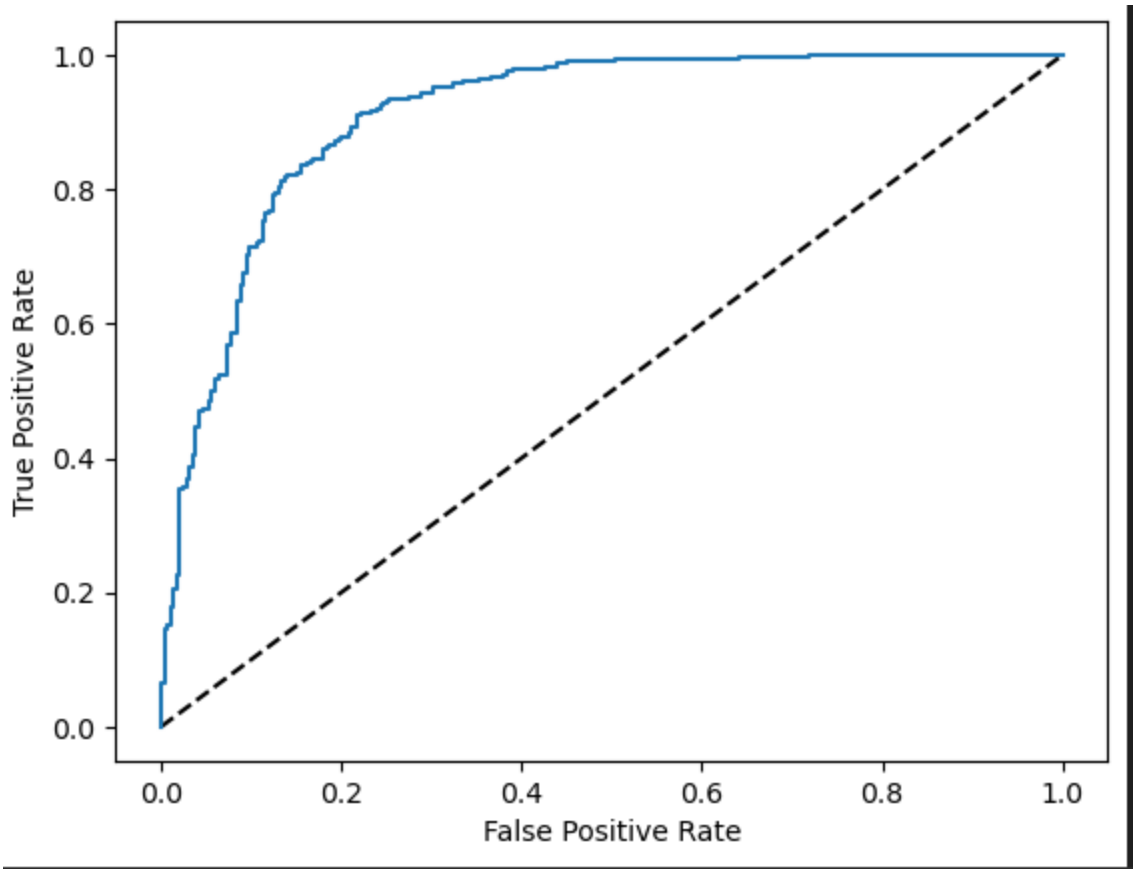
- The Deep Learning model captures complex patterns in dropout risk data but requires significant data and computational resources.

Comparative Analysis of Model Performance

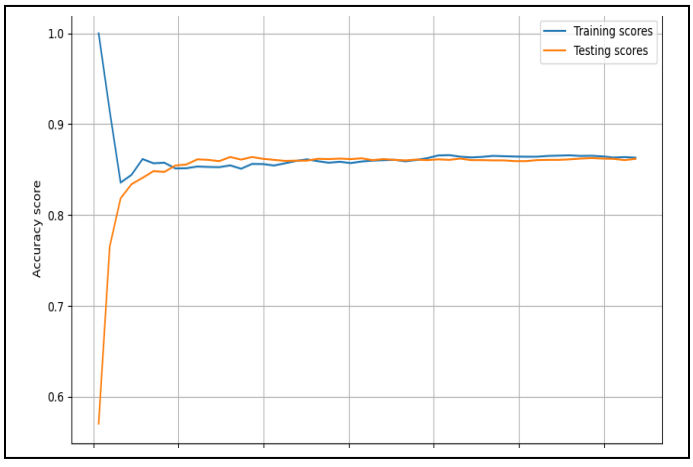
Model	Accuracy	Precision	Recall	F1-Score	Roc_auc_score
Logistic Regression	0.86	0.87	0.94	0.90	0.90
Decision Tree	0.85	0.90	0.88	0.89	0.88
Support Vector Machine	0.85	0.86	0.94	0.89	
Random Forest	0.87	0.89	0.92	0.90	0.91
GradientBoostingClassifier	0.86	0.87	0.94	0.90	0.90
Deep Learning					

Visualizations

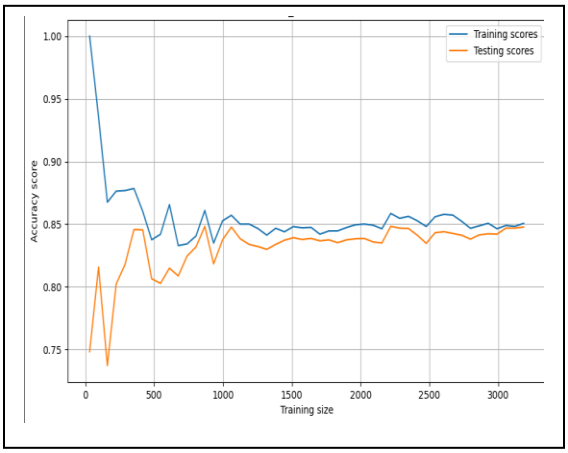
All models roc_curve is above the dotted lines showing the model is better than randomly guessing



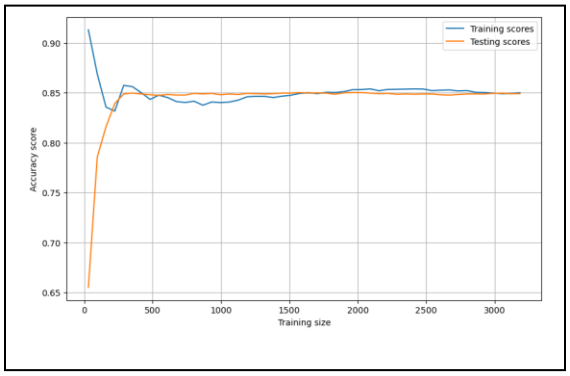
Analysis of Learning Curves



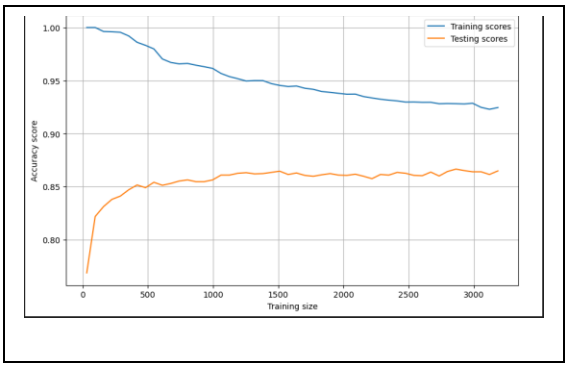
Logistic Regression



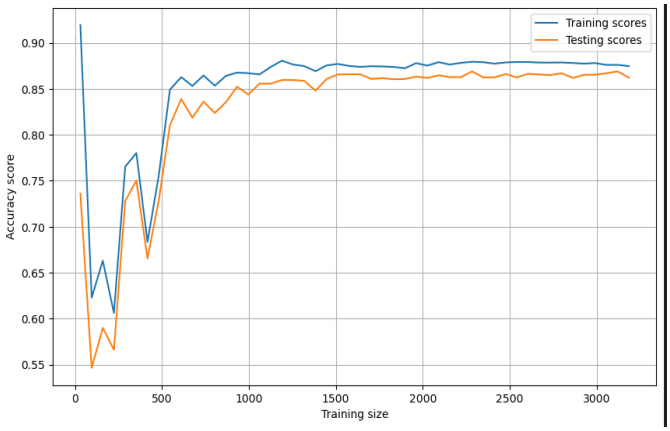
Decision Tree Classifier



Support Vector Machine



Random Forest Classifier



Gradient Boosting Classifier

Observations

- **Bias-Variance Tradeoff**: Models like Random Forest Classifier and decision Tree Classifier showed high variance, while Logistic Regression and Support vector machine showed good model complexity and shows that as training sizes increases, the training scores equals test scores.

- **Recommendations**

1. **Model Selection**:
    - **Best Performing Model**: Support vector classifier is recommended for its superior accuracy and robustness in predicting student dropout risk.
    - **Interpretability**: For a more interpretable model, consider Logistic Regression or Gradient Boosting Classifier, which provide clear insights into feature impacts on dropout risk.
2. **Future Work**:
    - Explore hyperparameter tuning techniques such as Grid Search or Random Search to enhance model performance further.
    - Investigate ensemble methods that combine multiple models for potentially improved predictions.
    - Assess additional features or data sources to enrich the predictive power of the