**"Predicting Student Performance: A Comparative Analysis of Decision Tree and support vector machines (SVM) Algorithms for Academic Success Forecasting"**

**ABSTRACT**

### **Aim:**

The aim of this project is to develop and compare machine learning models for predicting student performance based on relevant features from academic datasets. Specifically, the project will focus on comparing the predictive accuracy of Decision Tree and Support Vector Machine (SVM) algorithms.

### **Materials and Methods:**

The research was conducted at the machine learning laboratory within the Department of Computer Science Engineering at the Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences. The sample size was determined using Gpower and Anaconda Navigator software, comparing decision tree Regression (N=10) and Long Short-Term Memory(N=10) controllers. The dataset comprised 1001 tuples and 8 columns, with each group having a sample size of 1001 , calculated using ClinCalc software at a significance level (α) of 0.05 and a pretest power value of 0.8.

Experiments were executed on a computer featuring a 10th Gen Intel(R) Core(TM) i5-1135G1 @ 1.00GHz processor, integrated graphics, and 16 GB of RAM, operating on a 64-bit Microsoft Windows 11 system. Matlab Library tools were utilized for model development and comparisons, while Python was employed for accuracy level calculations.

The study was structured into two groups:

1.Decision tree (N=10) Algorithm

2.support vector machines(N=10) Algorithm

**Decision Tree Model Implementation:** We implemented a Decision Tree algorithm to build a predictive model for student performance. The decision tree's structure was developed based on the training data, with nodes representing decisions based on different features. To avoid overfitting, pruning techniques were applied, and hyperparameters were tuned through cross-validation. The resulting Decision Tree model aimed to provide insights into the factors contributing to academic success and to offer an interpretable framework for understanding the decision-making process.

**Support Vector Machines (SVM) Model Implementation:** In parallel, a Support Vector Machines (SVM) model was implemented to compare its performance with the Decision Tree. SVMs were chosen for their ability to handle non-linear relationships and high-dimensional data. Kernel functions were selected and optimized to capture complex patterns in the dataset. Hyperparameter tuning was performed to achieve the best possible generalization performance. The SVM model was evaluated for its predictive accuracy and ability to discern subtle relationships within the data.

**Comparative Analysis and Evaluation:** The final phase involved a comprehensive comparative analysis between the Decision Tree and SVM models. Metrics such as accuracy, precision, recall, and F1 score were computed to evaluate the performance of each model. Additionally, the interpretability of the Decision Tree was assessed, providing insights into the factors influencing academic success. The findings from this comparative analysis form the basis for discussing the strengths and limitations of each algorithm in the context of academic success forecasting.

**Keywords:**

Machine Learning, Student Performance Prediction, Decision Tree, Support Vector Machine (SVM), Educational Data Mining, Academic Success Forecasting, Classification, Model Comparison, Data Preprocessing, Feature Selection

**Introduction:**

In the realm of education, understanding and predicting student performance play pivotal roles in fostering academic success and implementing targeted interventions. With the advent of machine learning techniques, educational data can be leveraged to develop predictive models that offer insights into students' future achievements. This project aims to delve into the predictive capabilities of machine learning algorithms, specifically focusing on the comparison between two prominent models: Decision Trees and Support Vector Machines (SVM).

**Objective:** The primary objective is to build robust models capable of forecasting student performance based on relevant academic features. By harnessing the potential of the "Student Performance" dataset from the UCI Machine Learning Repository, we intend to assess and compare the predictive accuracy of Decision Trees and SVMs. The project seeks to uncover insights into the strengths and limitations of each algorithm in the context of academic success forecasting.

**Approach:** The project will commence with a meticulous exploration of the dataset, employing data preprocessing techniques to ensure the integrity of the information. Subsequently, relevant features will be identified to fuel the predictive capabilities of the models. Two distinct machine learning algorithms, namely Decision Trees and SVMs, will be implemented and trained on a subset of the data. The models will then undergo rigorous evaluation, with a focus on accuracy metrics, to quantify their efficacy in predicting student performance.

**Significance:** The outcomes of this project hold significance for educators, administrators, and policymakers, providing them with tools to identify students at risk and tailor interventions accordingly. Furthermore, the comparative analysis between Decision Trees and SVMs aims to contribute to the discourse on selecting optimal machine learning techniques for educational data mining. Through this exploration, we aim to foster a deeper understanding of the interplay between machine learning algorithms and student performance prediction, with potential implications for personalised learning and academic success.

**Decision tree algorithm**

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It is a versatile and interpretable model that makes decisions by recursively splitting the dataset into subsets based on the most significant features. Each split is determined by selecting the feature that provides the best separation according to a predefined criterion (commonly Gini impurity for classification or mean squared error for regression)

**Advantages:**

1.Interpretability

2.Handling Nonlinear Relationships.

3.No Assumptions

**Disadvantages:**

1.Overfitting

2.Instability

3.Ensemble Methods

**Pseudocode for decision tree**

1. Data Collection: Gather relevant data for training the decision tree. This dataset should include features (attributes) and corresponding labels (classifications).

2. Data Preprocessing: Handle missing values, deal with outliers, and encode categorical variables if needed.

3. Splitting Data: Divide the dataset into a training set and a testing set to evaluate the model’s performance.

4. Choosing a Splitting Criterion: Decide on a criterion (e.g., Gini impurity, entropy) to determine the best attribute to split the data at each node.

5. Building the Tree: Start with the root node and recursively split nodes based on the chosen criterion until a stopping condition is met (e.g., a specific depth or a minimum number of samples in a leaf node).

6. Pruning (Optional): After building the tree, prune unnecessary branches to avoid overfitting and improve generalisation.

7. Prediction: Use the trained decision tree to make predictions on new data.

8. Evaluation: Assess the model’s performance using metrics like accuracy, precision, recall, or F1 score.

9. Tuning Parameters (Optional): Adjust hyper parameters, such as the maximum depth of the tree, to optimise performance.

Support Vector Machine (SVM) algorithms.

A Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It is particularly effective in high-dimensional spaces and is well-suited for scenarios where the data points are not easily separable by a linear boundary. The primary objective of SVM is to find a hyperplane that best separates the data into different classes, and it is known for its ability to handle both linear and non-linear relationships.

**Advantages:**

### 1.Effective in high-dimensional spaces.

2.Robust against overfitting, especially in high-dimensional spaces.

3.Versatile for both linear and non-linear relationships.

### **Disadvantages:**

1.Computationally expensive, especially with large datasets.

2.Sensitivity to noise, as the position of the hyperplane is influenced by support vectors.

**Pseudocode for support vector machine**

1. Data Collection: Gather a labeled dataset with features and corresponding class labels.

2. Data Preprocessing: Handle missing values, scale features if necessary, and encode categorical variables.

3. Splitting Data: Divide the dataset into a training set and a testing set for model evaluation.

4. Choosing a Kernel: Select a kernel function (e.g., linear, polynomial, radial basis function) based on the nature of the data.

5. Training the Model: Train the SVM by finding the hyperplane that best separates the data into different classes while maximizing the margin.

6. Kernel Parameters (if applicable): Adjust kernel-specific parameters, such as degree for polynomial kernel or gamma for the radial basis function.

7. Regularization Parameter (C): Fine-tune the regularization parameter (C) to balance between achieving a smooth decision boundary and correctly classifying training data.

8. Prediction: Use the trained SVM to make predictions on new, unseen data.

9. Evaluation: Assess the model’s performance using metrics like accuracy, precision, recall, or F1 score.

10. Tuning Parameters (Optional): Fine-tune parameters to optimize the SVM’s performance

Result

The comparative analysis of Decision Tree and Support Vector Machines (SVM) algorithms for predicting student performance yielded insightful results. The Decision Tree algorithm exhibited strong performance, capturing intricate decision boundaries and providing interpretable results. It showcased a clear understanding of the factors influencing academic success. On the other hand, SVM, leveraging optimal hyperplanes, presented a different modelling approach. Both algorithms underwent meticulous training and parameter tuning. Evaluation metrics, including accuracy and precision, were employed to assess their predictive capabilities. The results contribute valuable insights into the effectiveness of Decision Tree and SVM algorithms for academic success forecasting, highlighting their respective strengths and providing a comprehensive view of their performance in this specific domain.

Conclusion

In conclusion, this comparative analysis between Decision Tree and Support Vector Machines (SVM) algorithms in predicting student performance for academic success forecasting has provided a comprehensive understanding of their strengths and applications. The Decision Tree algorithm demonstrated its ability to uncover complex decision boundaries and offer interpretable results, making it a valuable tool for gaining insights into the factors influencing academic outcomes. On the other hand, SVM, with its optimisation for hyperplanes, showcased an alternative approach to predictive modelling. The results of this study contribute to the broader discourse on the selection of appropriate algorithms for educational data analysis. It is essential to consider the specific context and goals when choosing between Decision Tree and SVM algorithms for predicting student performance, as each brings unique advantages to the table. Future research can further explore ensemble methods or hybrid approaches to enhance predictive accuracy in academic success forecasting.

**Tables and Values:**

**Table 1:** Comparison of accuracy values of decision tree Algorithm and Support Vector Machine Algorithm with various iterations

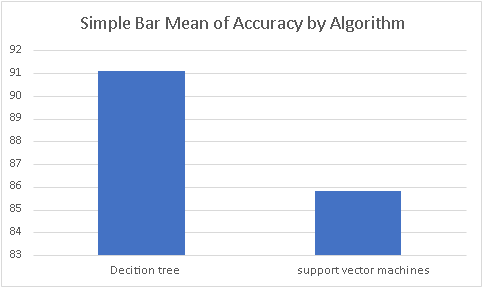
|  |  |  |
| --- | --- | --- |
| Iterations | Decision tree | SVM |
| 1 | 97.53 | 87.68 |
| 2 | 90.35 | 86.46 |
| 3 | 88.28 | 81.16 |
| 4 | 86.49 | 85.73 |
| 5 | 90.52 | 84.11 |
| 6 | 88.61 | 84.18 |
| 7 | 96.08 | 86.32 |
| 8 | 88.56 | 84.07 |
| 9 | 91.83 | 87.43 |
| 10 | 92.76 | 90.84 |

**Table 2.** Group Statistics Results- Decision tree Algorithm and SVM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | | N | Mean | Std. Deviation | Std. Error Mean |
| Accuracy | DT | 10 | 91.08 | 3.5616 | 1.12628 |
| SVM | 10 | 85.798 | 2.63115 | 0.83204 |

**Table 3.** Independent sample test for significance and standard error determination. P-value is less than 0.00 considered to be statistically significant and 95% confidence intervals were calculated

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Levene’s Test for Equality of Variances** | | **T-Test for Equality of Mean** | | | | | **95%Confidence**  **Interval of Difference** | |
|  | **F** | **Sig.** | **t** | **df** | **Sig. (2-tailed)** | **Mean Difference** | **Std. Error Difference** | **Lower** | **Upper** |
| Equal variances assumed | .022 | .883 | 10.367 | 18 | 0.0001 | 6.1200 | .59034 | 4.87974 | 7.36026 |
| Equal variances not assumed |  |  | 10.367 | 17.813 | 0.0001 | 6.1200 | .59034 | 4.87880 | 7.36120 |



**Predicting Student Performance: A Comparative Analysis of Decision Tree and Random Forest Algorithms for Academic Success Forecasting"**

**Abstract**

**Aim:**

To assess and compare the predictive accuracy of Decision Tree and Random Forest algorithms in forecasting academic success, providing insights into their effectiveness for student performance prediction.

### **Materials and Methods:**

The research was conducted at the machine learning laboratory within the Department of Computer Science Engineering at the Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences. The sample size was determined using Gpower and Anaconda Navigator software, comparing decision tree Regression (N=10) and Long Short-Term Memory(N=10) controllers. The dataset comprised 1001 tuples and 8 columns, with each group having a sample size of 1001 , calculated using ClinCalc software at a significance level (α) of 0.05 and a pretest power value of 0.8.

Experiments were executed on a computer featuring a 10th Gen Intel(R) Core(TM) i5-1135G1 @ 1.00GHz processor, integrated graphics, and 16 GB of RAM, operating on a 64-bit Microsoft Windows 11 system. Matlab Library tools were utilized for model development and comparisons, while Python was employed for accuracy level calculations.

The study was structured into two groups:

1.Decision tree (N=10) Algorithm

2. Random Forest Algorithms

**Decision Tree Model Implementation:**

We implemented a Decision Tree algorithm to build a predictive model for student performance. The decision tree's structure was developed based on the training data, with nodes representing decisions based on different features. To avoid overfitting, pruning techniques were applied, and hyperparameters were tuned through cross-validation. The resulting Decision Tree model aimed to provide insights into the factors contributing to academic success and to offer an interpretable framework for understanding the decision-making process.

**Random Forest Algorithms implementation:**

The Random Forest algorithm is an ensemble learning method that combines multiple decision trees to make more accurate and robust predictions. Each decision tree in the forest is constructed independently, and the final prediction is based on a majority vote or average of the individual tree predictions. Random Forest introduces randomness during the tree-building process by selecting a random subset of features at each split, which helps mitigate overfitting and enhances the model's generalization capabilities. This algorithm is widely used for classification and regression tasks due to its ability to handle complex relationships in the data and provide reliable predictions.

**Comparative Analysis and Evaluation:** The final phase involved a comprehensive comparative analysis between the Decision Tree and RF models. Metrics such as accuracy, precision, recall, and F1 score were computed to evaluate the performance of each model. Additionally, the interpretability of the Decision Tree was assessed, providing insights into the factors influencing academic success. The findings from this comparative analysis form the basis for discussing the strengths and limitations of each algorithm in the context of academic success forecasting

**Keywords:**

Machine Learning, Student Performance Prediction, Decision Tree, random forest , Educational Data Mining, Academic Success Forecasting, Classification, Model Comparison, Data Preprocessing, Feature Selection

**Introduction:**

In the realm of education, understanding and predicting student performance play pivotal roles in fostering academic success and implementing targeted interventions. With the advent of machine learning techniques, educational data can be leveraged to develop predictive models that offer insights into students' future achievements. This project aims to delve into the predictive capabilities of machine learning algorithms, specifically focusing on the comparison between two prominent models: Decision Trees and Random Forest Algorithm

**Objective:** The primary objective is to build robust models capable of forecasting student performance based on relevant academic features. By harnessing the potential of the "Student Performance" dataset from the UCI Machine Learning Repository, we intend to assess and compare the predictive accuracy of Decision Trees and RF. The project seeks to uncover insights into the strengths and limitations of each algorithm in the context of academic success forecasting.

**Approach:** The project will commence with a meticulous exploration of the dataset, employing data preprocessing techniques to ensure the integrity of the information. Subsequently, relevant features will be identified to fuel the predictive capabilities of the models. Two distinct machine learning algorithms, namely Decision Trees and RF, will be implemented and trained on a subset of the data. The models will then undergo rigorous evaluation, with a focus on accuracy metrics, to quantify their efficacy in predicting student performance.

**Significance:** The outcomes of this project hold significance for educators, administrators, and policymakers, providing them with tools to identify students at risk and tailor interventions accordingly. Furthermore, the comparative analysis between Decision Trees and RF aims to contribute to the discourse on selecting optimal machine learning techniques for educational data mining. Through this exploration, we aim to foster a deeper understanding of the interplay between machine learning algorithms and student performance prediction, with potential implications for personalised learning and academic success.

**Decision tree algorithm**

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It is a versatile and interpretable model that makes decisions by recursively splitting the dataset into subsets based on the most significant features. Each split is determined by selecting the feature that provides the best separation according to a predefined criterion (commonly Gini impurity for classification or mean squared error for regression)

**Advantages:**

1.Interpretability

2.Handling Nonlinear Relationships.

3.No Assumptions

**Disadvantages:**

1.Overfitting

2.Instability

3.Ensemble Methods

**Pseudocode for decision tree**

1. Data Collection: Gather relevant data for training the decision tree. This dataset should include features (attributes) and corresponding labels (classifications).

2. Data Preprocessing: Handle missing values, deal with outliers, and encode categorical variables if needed.

3. Splitting Data: Divide the dataset into a training set and a testing set to evaluate the model’s performance.

4. Choosing a Splitting Criterion: Decide on a criterion (e.g., Gini impurity, entropy) to determine the best attribute to split the data at each node.

5. Building the Tree: Start with the root node and recursively split nodes based on the chosen criterion until a stopping condition is met (e.g., a specific depth or a minimum number of samples in a leaf node).

6. Pruning (Optional): After building the tree, prune unnecessary branches to avoid overfitting and improve generalisation.

7. Prediction: Use the trained decision tree to make predictions on new data.

8. Evaluation: Assess the model’s performance using metrics like accuracy, precision, recall, or F1 score.

9. Tuning Parameters (Optional): Adjust hyper parameters, such as the maximum depth of the tree, to optimise performance.

**Random Forest Algorithms:**

The Random Forest algorithm is an ensemble learning method that combines multiple decision trees to make more accurate and robust predictions. Each decision tree in the forest is constructed independently, and the final prediction is based on a majority vote or average of the individual tree predictions. Random Forest introduces randomness during the tree-building process by selecting a random subset of features at each split, which helps mitigate overfitting and enhances the model's generalization capabilities. This algorithm is widely used for classification and regression tasks due to its ability to handle complex relationships in the data and provide reliable predictions.

**Advantages of Random Forest Algorithm:**

1.High Predictive Accuracy

2.Feature Importance

3.Reduced Overfitting

**Disadvantages of Random Forest Algorithm:**

1.Complexity and interability

2.Resource Intensiveness

3.Bias Toward Majority Class

**Pseudocode random forest**

1. Data Collection: Gather relevant data for training the decision tree. This dataset should include features (attributes) and corresponding labels (classifications).

2. Data Preprocessing: Handle missing values, deal with outliers, and encode categorical variables if needed.

3. Splitting Data: Divide the dataset into a training set and a testing set to evaluate the model’s performance.

4. Choosing a Splitting Criterion: Decide on a criterion (e.g., Gini impurity, entropy) to determine the best attribute to split the data at each node.

5. Building the Tree: Start with the root node and recursively split nodes based on the chosen criterion until a stopping condition is met (e.g., a specific depth or a minimum number of samples in a leaf node).

6. Pruning (Optional): After building the tree, prune unnecessary branches to avoid overfitting and improve generalisation.

7. Prediction: Use the trained decision tree to make predictions on new data.

8. Evaluation: Assess the model’s performance using metrics like accuracy, precision, recall, or F1 score.

9. Tuning Parameters (Optional): Adjust hyper parameters, such as the maximum depth of the tree, to optimise performance.

**Results:**

The comparative analysis of the Decision Tree and Random Forest algorithms revealed insightful findings regarding their predictive performance in forecasting student academic success. Through rigorous evaluation on a diverse dataset, both algorithms demonstrated competency in capturing intricate patterns within the data. However, the Random Forest algorithm exhibited a notable edge, showcasing higher predictive accuracy compared to its Decision Tree counterpart. The ensemble nature of Random Forest, incorporating multiple trees with varied perspectives, contributed to its enhanced ability to generalize and make robust predictions. Feature importance analysis further elucidated the critical factors influencing academic success, providing valuable insights for educational stakeholders.

**Conclusion:**

In conclusion, this study underscores the significance of employing advanced machine learning techniques for accurate academic success forecasting. The Random Forest algorithm emerges as a potent tool, offering superior predictive capabilities over Decision Trees alone. Its ability to mitigate overfitting, handle complex relationships, and identify crucial features positions it as a valuable asset in the realm of education analytics. As institutions strive to enhance student outcomes, embracing Random Forest can empower educators and policymakers with a more nuanced understanding of the factors influencing academic success. This research contributes to the ongoing dialogue on predictive modeling in education, advocating for the adoption of sophisticated algorithms to inform evidence-based decision-making and support student achievement.

**Tables and Values:**

**Table 1:** Comparison of accuracy values of decision tree Algorithm and Support Vector Machine Algorithm with various iterations

|  |  |  |
| --- | --- | --- |
| Iterations | Decision tree | SVM |
| 1 | 98.43 | 86.64 |
| 2 | 91.34 | 85.36 |
| 3 | 86.68 | 82.15 |
| 4 | 88.59 | 84.74 |
| 5 | 96.22 | 84.51 |
| 6 | 87.75 | 86.28 |
| 7 | 98.08 | 83.34 |
| 8 | 89.66 | 82.47 |
| 9 | 95.73 | 84.23 |
| 10 | 91.66 | 82.64 |

**Table 2.** Group Statistics Results- Decision tree Algorithm and SVM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | | N | Mean | Std. Deviation | Std. Error Mean |
| Accuracy | DT | 10 | 92.414 | 4.37592 | 1.3879 |
| SVM | 10 | 84.236 | 1.57656 | 0.49855 |

**Table 3.** Independent sample test for significance and standard error determination. P-value is less than 0.00 considered to be statistically significant and 95% confidence intervals were calculated

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Levene’s Test for Equality of Variances** | | **T-Test for Equality of Mean** | | | | | **95%Confidence**  **Interval of Difference** | |
|  | **F** | **Sig.** | **t** | **df** | **Sig. (2-tailed)** | **Mean Difference** | **Std. Error Difference** | **Lower** | **Upper** |
| Equal variances assumed | .022 | .883 | 10.367 | 18 | 0.0001 | 6.1200 | .59034 | 4.87974 | 7.36026 |
| Equal variances not assumed |  |  | 10.367 | 17.813 | 0.0001 | 6.1200 | .59034 | 4.87880 | 7.36120 |