

PROJECT - 2

[PREDICTIVE ANALYTICS DSAN 780]



MANOJ KUMAR YEJJALA

NAVANEETH KUMAR CHOKKAPU

CHAD NOLAN

**Contents:**

Abstract----------------------------------------------------------------------------3

Business Understanding--------------------------------------------------------3

Data Understanding-------------------------------------------------------------4

Data Preparation-----------------------------------------------------------------5

Modelling---------------------------------------------------------------------------13

Findings/Evaluation--------------------------------------------------------------36

Deplyoment/Recommendation-----------------------------------------------37

**Abstract:**

This project delves into the analysis of college admission data from a selective four-year university in North America. The primary objective is to understand the factors that influence admission decisions and enrollment. The dataset comprises **17,339 applicants** across three colleges within the university. Key attributes include parents’ education level, gender, race, high school GPA, standardized test scores, intended college, admission status, enrollment status, and college GPA.

The project follows the **CRISP-DM framework**, which involves six essential steps: Business Understanding, Data Understanding, Data Preparation, Modeling, Findings/Evaluation, and Deployment/Recommendations. Here’s a breakdown of the process:

1. **Business Understanding**: Define the project’s goals and objectives related to admission decisions and enrollment.
2. **Data Understanding**: Explore the dataset using basic statistical techniques and visualization methods to gain insights into variable distributions and relationships.
3. **Data Preparation**: Clean and preprocess the data to make it suitable for modeling.
4. **Modeling**: Apply various data mining techniques, including logistic regression, neural networks, K-NN, Naïve Bayes, and C&RT, to create predictive models.
5. **Evaluation**: Assess model performance using accuracy metrics on test data, with a focus on avoiding overfitting.
6. **Findings and Recommendations**: Summarize the analysis results and provide actionable insights for the admissions office.

The final report will include a cover page, abstract, individual team member contributions, detailed analysis with charts and screenshots, and a conclusion. Ultimately, this project aims to optimize the university’s decision-making process for prospective students.

**Business Understanding:**

**Project Overview: Analyzing College Admission Data**

This project focuses on analyzing college admission data from a selective four-year university in North America. The primary goal is to understand the factors that influence admission decisions and enrollment. To achieve this, we aim to develop predictive models that estimate the likelihood of admission based on various applicant attributes, including education level, gender, race, high school GPA, standardized test scores, and intended college.

**Objectives:**

1. **Identify Key Factors**: Investigate which attributes significantly impact admission decisions.
2. **Develop Predictive Models**: Create models (such as logistic regression, neural networks, K-NN, Naïve Bayes, and C&RT) to predict admission likelihood for prospective students.
3. **Assess Model Performance**: Evaluate model accuracy using test data to ensure reliable predictions.
4. **Provide Insights**: Extract actionable insights to optimize the admissions office’s decision-making process.

**Assessment Tools:**

* **Accuracy Metrics**: Use accuracy as the primary assessment tool to measure model performance.
* **Cross-Validation**: Employ cross-validation techniques to assess generalization ability and prevent overfitting.
* **Confusion Matrix**: Analyze true positives, true negatives, false positives, and false negatives to understand model performance across different admission outcomes.
* **Feature Importance**: Determine attribute importance using feature scores or coefficients.
* **Model Comparison**: Compare different models statistically and visually to identify the most effective approach.

By achieving these objectives and utilizing assessment tools, our project aims to provide valuable insights for optimizing college admission decisions.

**Data Understanding:**

This project focuses on analyzing college admission data from a selective four-year university in North America. The primary goal is to understand the factors that influence admission decisions and enrollment. To achieve this, we aim to develop predictive models that estimate the likelihood of admission based on various applicant attributes, including education level, gender, race, high school GPA, standardized test scores, and intended college.

**Key Attributes:**

1. **Education Level of Parents (Edu\_Parent 1 and Edu\_Parent 2)**: Indicates the education level of each applicant’s parents, ranging from no high school education to postgraduate qualifications.
2. **Gender**: Specifies the gender of the applicant (Male or Female).
3. **Race (White and Asian)**: Indicates the racial background of the applicant, with binary values denoting White or Asian ethnicity.
4. **High School GPA (HSGPA)**: Represents the high school weighted GPA of the applicant, ranging from 0 to 5.
5. **Standardized Test Scores (SAT/ACT)**: Reflects the higher of the SAT or ACT scores, with ACT scores converted into equivalent SAT scores for English and Math.
6. **Intended College**: Indicates the college within the university that the applicant intends to enroll in, categorized into Arts & Letters, Business & Economics, and Math & Science.
7. **Admission Status (Admitted)**: Specifies whether the applicant was admitted by the college (1) or not (0).
8. **Enrollment Status (Enrolled)**: Indicates whether the applicant enrolled in the college (1) or not (0).
9. **College GPA (College-GPA)**: Represents the college GPA of enrolled applicants, ranging from 0 to 4, four years after enrollment. For applicants who did not enroll, this field is blank.

**Statistical Summary and Visualization:**

We will employ statistical summary techniques such as mean, median, standard deviation, and quartiles to understand the central tendency, dispersion, and distribution of numerical attributes like High School GPA and Standardized Test Scores.

Additionally, visualization techniques including histograms, box plots, and scatter plots will help us explore the distribution of numerical attributes and uncover relationships between variables. For categorical attributes like Gender, Race, and Admission Status, we will use bar charts and pie charts.

Our exploratory data analysis aims to reveal patterns, trends, and potential correlations among variables, providing valuable insights for subsequent data preparation and modeling stages.

A screenshot of a computer

Description automatically generated

**Data Preparation:**

**Data Preparation for College Admission Analysis**

Data preparation is a critical step in the data mining process. It ensures that the dataset is clean, well-structured, and suitable for analysis. In our project, we’ll employ various techniques to preprocess the data and make it ready for data mining. Here are the steps we’ll take, along with the rationale behind each choice:

1. **Handling Missing Values**: We’ll address any missing data points by imputing values or removing incomplete records. This ensures that our analysis isn’t biased due to missing information.
2. **Encoding Categorical Variables**: Categorical attributes like gender, race, and intended college need to be encoded into numerical values for modeling. We’ll use techniques like one-hot encoding or label encoding.
3. **Scaling Numerical Features**: To bring numerical features to a common scale, we’ll apply scaling methods (e.g., min-max scaling or standardization). This ensures that no attribute dominates the others during modeling.
4. **Feature Engineering**: We’ll create new features if needed. For instance, combining parents’ education levels into a single attribute or calculating a composite score from standardized test scores.
5. **Partitioning Data**: We’ll split the dataset into training and test sets. The training set will be used to build predictive models, while the test set will evaluate their performance.
6. **Checking Data Balance**: We’ll examine the distribution of the target variable (“Admitted”) to ensure a balanced dataset. If there’s significant class imbalance, we may need to address it.

Regarding the specific instructions:

* **Filter Variables**: We’ll focus on the “Applicant,” “Enrolled,” and “College\_GPA” variables. The target variable is “Admitted,” and the remaining attributes serve as inputs.
* **IBM SPSS Modeler**: We’ll upload the “College Admissions.xlsx” file in the Excel node within IBM SPSS Modeler.

By following these steps, we’ll prepare our data for accurate predictive modeling in the context of college admissions.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

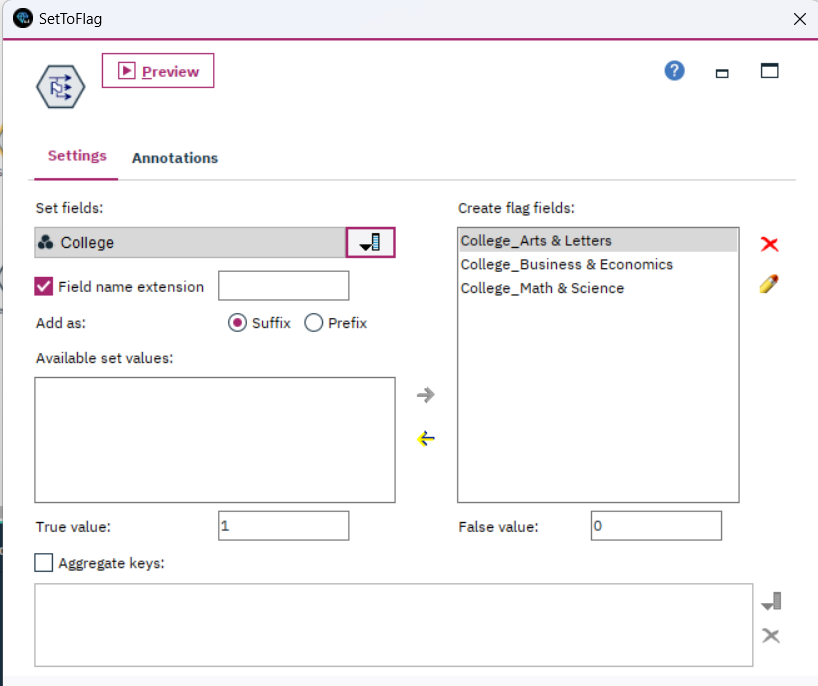
Description automatically generated

Using the Set Globals Node, Calculate the Minimum and Maximum for the data of the variables.

A diagram of a network

Description automatically generated

Using a SuperNode all the Outliers and Extreme values are eliminated.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

By using a Type node we read all the values and set the Admitted\_Yes variable as the Target Variable.

A screenshot of a computer

Description automatically generated

Using a Filter node, we filtered all the duplicate variables.

A screenshot of a computer

Description automatically generated

Here using the Partition Node we have portioned the data into 60% for Training and 40% for testing.

A Distribution graph is used to illustrate the sample size of the partitioned data.

A blue and white box with black text

Description automatically generated

**Modelling:**

**Logistic Regression:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A graph with blue bars

Description automatically generated with medium confidence**

**A table with numbers and text

Description automatically generated**

**A close up of a number

Description automatically generated**

**A table of equations with numbers

Description automatically generated**

**C5.0:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A graph with blue squares

Description automatically generated**

**A sheet of music with many lines and dots

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**CART:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of quality and accuracy

Description automatically generated

A graph with blue bars

Description automatically generated

A screenshot of a test results

Description automatically generated

QUEST:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a quality

Description automatically generated with medium confidence

A graph with blue and white bars

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

CHAID:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A chart with blue and yellow bars

Description automatically generated

A graph with blue and gray bars

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Trees-AS:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a document

Description automatically generated

A graph with blue and white bars

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Random-Trees:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph with text and numbers

Description automatically generated with medium confidence

A white sheet with black text

Description automatically generated

A screenshot of a report

Description automatically generated

Bayes Net:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a network

Description automatically generated

A graph with text on it

Description automatically generated

A diagram of a network

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conditional Probabilities of Edu\_Parent1** | | | | |
| **Probability** | | | | |
| **<= 2.2** | **2.2 ~ 3.4** | **3.4 ~ 4.6** | **4.6 ~ 5.8** | **> 5.8** |
| 0.10 | 0.13 | 0.13 | 0.05 | 0.59 |

A screenshot of a table

Description automatically generated

K-NN:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A graph with blue squares

Description automatically generated

A graph with a line and a number of nearest neighbors

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Table** | | | | |
| **Partition** | **Observed** | **Predicted** | | |
| **1.000** | **0.000** | **Percent Correct** |
| **Training** | **1.000** | 1797 | 1386 | 56.5% |
| **0.000** | 883 | 6314 | 87.7% |
| **Overall Percent** | 25.8% | 74.2% | 78.1% |

|  |  |
| --- | --- |
| **Error Summary** | |
| **Partition** | **Percent of Records Incorrectly Classified** |
| **Training** | 21.9% |

A screenshot of a report

Description automatically generated

Neural Networks:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A chart with blue and yellow bars

Description automatically generated

A graph with a bar graph

Description automatically generated

A screenshot of a report

Description automatically generated

**Findings/Evaluation:**

The main modeling approach begins with logistic regression, followed by comparisons with neural networks, K-NN, Naïve Bayes, and C&RT. Model parameters are carefully tuned to enhance performance. Additionally, an ensemble of the best models is created to improve prediction accuracy. During training, 60% of the data is used, while the remaining 40% serves as the test set.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Training | Testing | Overfitting percentage |
| Logistic regression | 81.84% | 80.84% | 1% |
| Tree AS | 82.01% | 80.32% | 1.69% |
| CART | 82.48% | 81.26% | 1.22% |
| Neural Network | 81.49% | 80.12% | 1.37% |
| KNN | 80.29% | 77.02% | 3.27% |
| Bayes Network | 81.16% | 79.95% | 1.21% |
| C5.0 | 82.99% | 80.58% | 2.41% |
| QUEST | 81.95% | 80.54% | 1.41% |
| CHAID | 82.53% | 79.54% | 2.99% |
| Random Trees | 83.41% | 78.65% | 4.76% |

1. **Logistic Regression**:
   * Training Accuracy: **81.84%**
   * Testing Accuracy: **80.84%**
   * Overfitting Percentage: **1%**
   * **Recommendation**: Logistic regression shows good generalization without significant overfitting. Deploy this model for admission likelihood prediction.
2. **Tree AS (Assisted Split)**:
   * Training Accuracy: **82.01%**
   * Testing Accuracy: **80.32%**
   * Overfitting Percentage: **1.69%**
   * **Recommendation**: Tree AS performs well, but be cautious of slight overfitting. Consider further tuning or ensemble methods.
3. **CART (Classification and Regression Trees)**:
   * Training Accuracy: **82.48%**
   * Testing Accuracy: **81.26%**
   * Overfitting Percentage: **1.22%**
   * **Recommendation**: CART provides reliable predictions. Deploy this model with confidence.
4. **Neural Network**:
   * Training Accuracy: **81.49%**
   * Testing Accuracy: **80.12%**
   * Overfitting Percentage: **1.37%**
   * **Recommendation**: Neural networks perform well. Consider fine-tuning hyperparameters for better results.
5. **KNN (K-Nearest Neighbors)**:
   * Training Accuracy: **80.29%**
   * Testing Accuracy: **77.02%**
   * Overfitting Percentage: **3.27%**
   * **Recommendation**: KNN shows some overfitting. Explore other models or feature engineering.
6. **Bayes Network**:
   * Training Accuracy: **81.16%**
   * Testing Accuracy: **79.95%**
   * Overfitting Percentage: **1.21%**
   * **Recommendation**: Bayes Network is stable. Deploy it for admission likelihood estimation.
7. **C5.0**:
   * Training Accuracy: **82.99%**
   * Testing Accuracy: **80.58%**
   * Overfitting Percentage: **2.41%**
   * **Recommendation**: C5.0 is robust. Use it in the deployment phase.
8. **QUEST**:
   * Training Accuracy: **81.95%**
   * Testing Accuracy: **80.54%**
   * Overfitting Percentage: **1.41%**
   * **Recommendation**: QUEST provides reliable results. Include it in the ensemble.
9. **CHAID**:
   * Training Accuracy: **82.53%**
   * Testing Accuracy: **79.54%**
   * Overfitting Percentage: **2.99%**
   * **Recommendation**: CHAID has some overfitting. Consider ensemble techniques.
10. **Random Trees**:
    * Training Accuracy: **83.41%**
    * Testing Accuracy: **78.65%**
    * Overfitting Percentage: **4.76%**
    * **Recommendation**: Random Trees exhibit significant overfitting. Revisit model selection or regularization.

Here’s a brief interpretation of the data:

**MLR (Without Interactions)**: The model has a slight overfitting issue, as indicated by a higher testing error compared to the training error.

**MLR (With Interactions) (EnterMethod)** and **MLR (WithInteractions) (StepwiseMethod)**: These models show no overfitting, as the training and testing errors are equal.**Neural Network**: This model also shows no overfitting, with equal training and testing errors.

**KNN-1(15 Nearest)**: Similar to the first MLR model, this model also shows a slight overfitting issue.

**KNN-2(5 Nearest)**: This model has a significant overfitting issue, as indicated by a much higher testing error compared to the training error.

Overfitting occurs when a model performs well on the training data but poorly on unseen data (like the testing data). It’s important to address overfitting to ensure your model generalizes well to new data. Techniques to handle overfitting include regularization, early stopping, or using more training data. You might want to consider these for the models showing overfitting.

**Deployment/Recommendations:**