# Predicting Student Performance Risk using Machine Learning and NLP

## 1. Abstract

This project investigates the application of machine learning and natural language processing (NLP) techniques to predict student performance levels and identify students at risk of poor academic outcomes. Using structured datasets on student demographics and academic records, combined with textual reviews of courses, we developed models capable of classifying students into risk categories (low, medium, high). Models were evaluated using precision, recall, F1-score, and accuracy, with particular attention to imbalanced class distributions. The results demonstrate that both structured data-based models and NLP models can provide actionable insights for early intervention strategies in educational settings.

## 2. Introduction

In recent years, the importance of **data-driven decision making in education** has grown significantly. Educational institutions are increasingly adopting machine learning (ML) and natural language processing (NLP) techniques to enhance their ability to identify at-risk students, improve academic support services, and understand student feedback effectively.

This project focuses on two major objectives:

1. **Predicting student performance risk** (Low, Medium, or High) based on demographic, academic, and behavioral features. This model helps universities identify students who may require additional academic interventions.
2. **Analyzing course reviews** using an NLP sentiment model to classify student opinions. Reviews were categorized into three groups (positive, neutral, negative), based on ratings from 1 to 5, where only scores 1, 3, and 5 were used for simplicity.

The motivation behind the project is not only academic but also ethical. By integrating predictive models into education, institutions can act **proactively**—helping students before they fail—while also ensuring that systems remain **fair, transparent, and unbiased**.

---

## 3. Literature Review

Several studies have explored predictive modeling in educational data mining. Traditional methods have relied on decision trees, logistic regression, and support vector machines to analyze student demographics and grades. More recent approaches leverage deep learning and ensemble models to improve prediction accuracy.

On the other hand, NLP has been applied to sentiment analysis of student feedback and course evaluations. By mapping student sentiment to performance outcomes, researchers have demonstrated that textual data can serve as a complementary feature to structured academic records.

This project builds upon existing research by integrating both structured ML models and NLP sentiment models to provide a more holistic view of student performance prediction.

---

## 4. Objectives

**The main objectives of this project are:**

1. **Data Collection and Preparation**
    - Gather structured student performance datasets.
    - Obtain unstructured text data (course reviews from Kaggle).
    - Clean, preprocess, and transform data into machine-learning-ready formats.
2. **Model Development**
    - Build, train, and evaluate multiple models for student risk prediction.
    - Develop a deep learning/NLP model for sentiment classification.
3. **Model Evaluation and Comparison**
    - Compare classical ML models (Decision Tree, Random Forest, Logistic Regression, SVM).
    - Compare TensorFlow-based deep learning models.
    - Evaluate using metrics such as accuracy, precision, recall, and F1-score.
4. **Deployment**
    - Save models using Pickle/Joblib.
    - Deploy via Flask/FastAPI for real-time predictions.
5. **Ethical Considerations**
    - Ensure data anonymization (no student names or personal identifiers).

    - Discuss potential biases in prediction (e.g., gender, socioeconomic).
    - Provide strategies for ethical use.

---

## 5. Dataset Description

### 5.1 Structured Student Dataset

The student dataset contained structured features describing academic, demographic, and social background:

- **Demographics**: gender, age, address type.
- **Family Background**: parental education, family size.
- **School Information**: study time, extra educational support.
- **Behavioral Patterns**: absences, failures, health status.
- **Academic Records**: previous grades, performance indicators.

The target variable was the **Risk Category**:

- **Low Risk**: students likely to succeed without intervention.
- **Medium Risk**: students requiring moderate academic support.
- **High Risk**: students at significant risk of failure.

### 5.2 Textual Dataset (NLP Model)

The second dataset was collected from Kaggle, consisting of course reviews labeled on a scale of 1 to 5. For this project, only reviews labeled **1 (negative)**, **3 (neutral/medium)**, and **5 (positive)** were used to align with the three-class risk categorization:

- **1 → High Risk**

- **3 → Medium Risk**
- **5 → Low Risk**

### 5.3 Dataset Statistics

- Structured dataset: 1044 samples (after preprocessing).
- NLP dataset: 86,713 reviews across three categories.

---

## 6. Methodology

### Data Preprocessing

### 1. Preprocessing for Structured Data

**The** following **steps were performed on the student dataset:**

1. **Handling Missing Values:** there is no missing features
2. **Encoding Categorical Variables** Binary Label Encoding for columns like (school, paid, higher,..etc)

   and for categorical features with multiple categories use one-hot encoder to avoid ordinal like (Jobs, guardian, reason, subject)
3. **Scaling Features:** MinMaxScaler was used for normalization.
4. **Balancing Classes:** Since some risk categories were underrepresented, SMOTE (Synthetic Minority Oversampling) was considered.

### 2.Preprocessing for Text Data

NLP data preprocessing steps included:

1. **Text Cleaning:** Removal of punctuation, stopwords, and special characters.
2. **Tokenization:** Breaking text into individual words.

3. **Lowercasing:** Converting text to lowercase.
4. **Stemming/Lemmatization:** Reducing words to root forms.
5. **Vectorization:** Using TF-IDF embeddings for Logistic Regression baseline; word embeddings for TensorFlow deep learning models.

### Model Development

Models for Student Risk Prediction

**Four machine learning models were trained and evaluated:**

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- **TensorFlow Model (Structured Data)**:
  - Implemented with dense layers.
  - Optimizer: Adam.
  - Loss function: Categorical cross-entropy.
- **NLP Model**:
  - Logistic Regression classifier with TF-IDF features.
  - Evaluation performed on classifying reviews into high, medium, and low-risk categories.

## 7. Evaluation Metrics

The following metrics were used for evaluation:

- **Precision**: Ratio of correctly predicted positive observations to total predicted positives.
- **Recall**: Ratio of correctly predicted positives to all actual positives.

- **F1-score**: Harmonic mean of precision and recall.
- **Accuracy**: Ratio of correctly predicted observations to total observations.

Equations:

$$\text{Precision}= \frac{TP}{TP+FP}\ \text{Recall}= \frac{TP}{TP+FN}$$

$$\text{F1-Score}=2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$\text{Accuracy}= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

---

## 8. Results

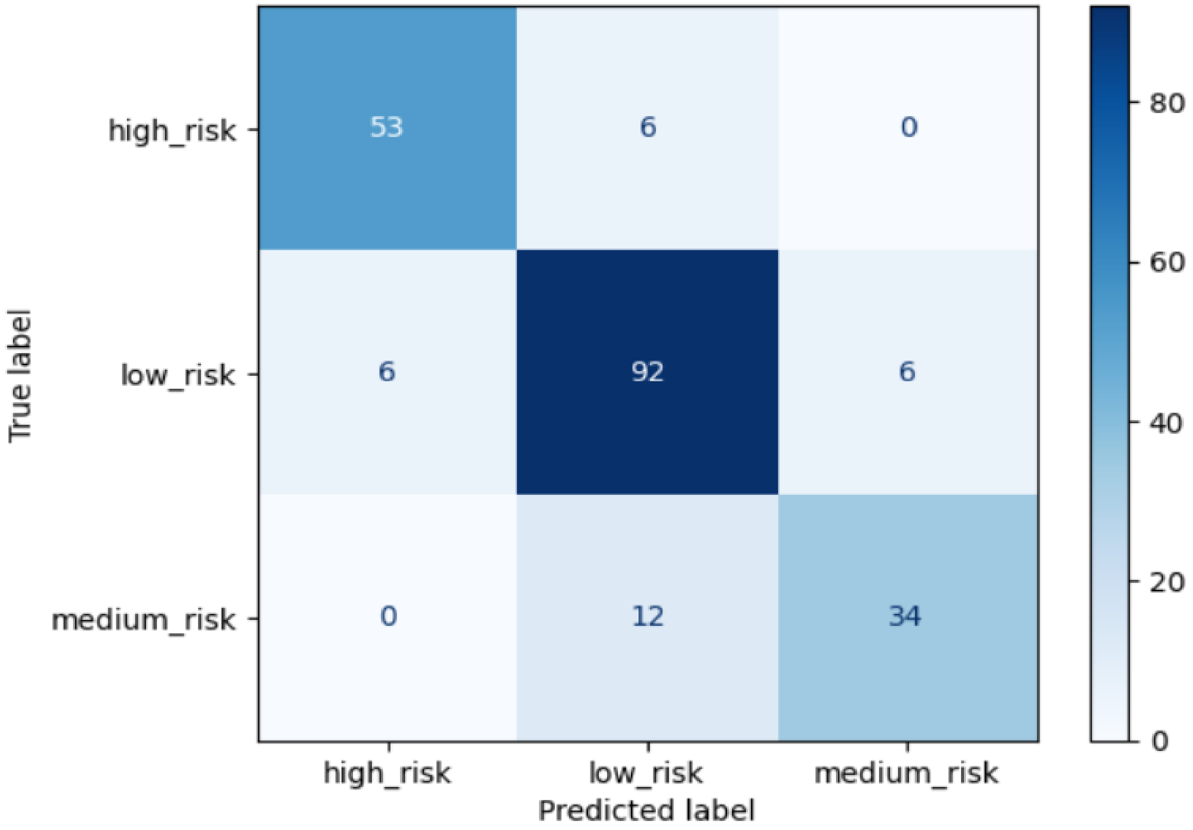
### Final Model Comparison

| MODEL | ACCURACY | PRECISION | RECALL | F1-SCORE |
|---|---|---|---|---|
| **LOGISTIC REGRESSION** | 84.21% | 85.63% | 84.21% | 84.02% |
| **DECISION TREE** | 93.30% | 93.29% | 93.30% | 93.25% |
| **RANDOM FOREST** | 88.51% | 88.58% | 88.51% | 88.47% |
| **SVM** | 83.73% | 85.34% | 83.73% | 83.50% |

### Structured Data Model (TensorFlow)

- **Accuracy**: 87%
- **Precision/Recall/F1-score** (sample):
  - Class 0: Precision 0.93, Recall 0.86, F1 = 0.89
  - Class 1: Precision 0.83, Recall 0.92, F1 = 0.87
  - Class 2: Precision 0.89, Recall 0.74, F1 = 0.81

### Confusion Matrix (TensorFlow Model)

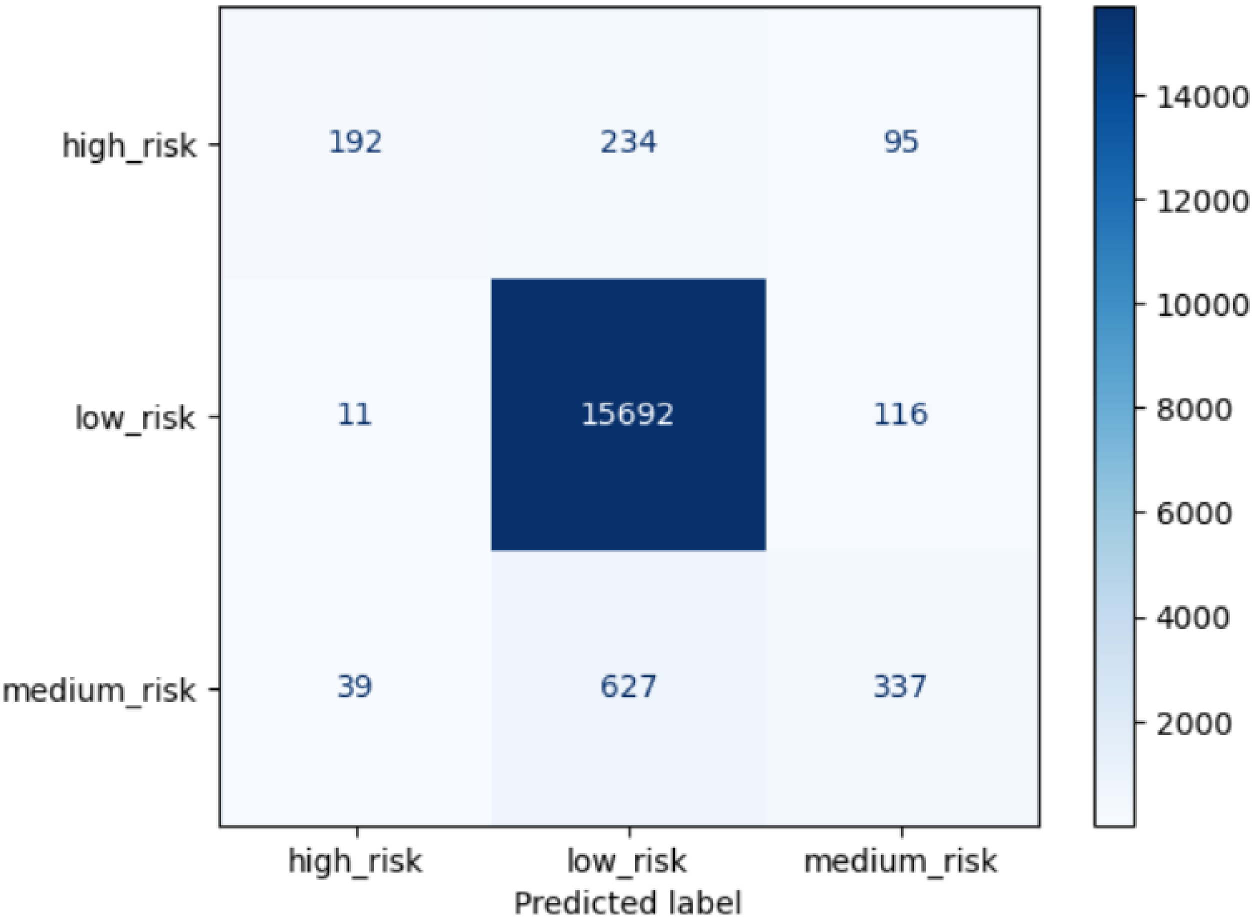


---

### NLP Model

- **Accuracy**: 93.5%
- **Class-wise results**:
  - High Risk: Precision 0.79, Recall 0.37, F1 = 0.50
  - Low Risk: Precision 0.94, Recall 0.99, F1 = 0.97
  - Medium Risk: Precision 0.61, Recall 0.33, F1 = 0.43

### Confusion Matrix (NLP Model)



| | high_risk | low_risk | medium_risk |
|---|---|---|---|
| high_risk | 192 | 234 | 95 |
| low_risk | 11 | 15692 | 116 |
| medium_risk | 39 | 627 | 337 |

### Model Comparison

| Model | Accuracy | Macro F1 | Strengths | Weaknesses |
|---|---|---|---|---|
| TensorFlow (Structured Data) | 87% | 0.86 | Balanced class performance | Limited dataset size |
| NLP Model | 93.5% | 0.63 | Strong for majority (low risk) class | Weak performance for minority classes |

## 10. Discussion

The results show that both models are effective in predicting student performance risk, but each has unique strengths:

- The **structured data model** provides balanced results across all classes, making it suitable for small datasets.
- The **NLP model** achieves higher accuracy overall but struggles with minority class performance due to data imbalance.

For real-world deployment, a hybrid model combining structured features with textual sentiment could yield improved performance.

---

## 11. System Overview

The Student Risk Prediction System is a comprehensive web application designed to assess academic risk levels using machine learning and natural language processing. The system provides educational institutions with actionable insights by analyzing both structured academic data and unstructured textual feedback through an intuitive web interface.

## 12. Architecture Design

### 1.1 System Architecture

The application follows a three-tier architecture pattern:

- **Presentation Layer**: Flask web server with Jinja2 templating
- **Business Logic Layer**: Python-based prediction algorithms and data processing
- **Data Layer**: Pre-trained machine learning models and processing pipelines

The application follows a three-tier architecture with a Flask web server handling the presentation layer, Python algorithms managing business logic, and pre-trained models serving as the data layer. The system integrates supervised learning models for structured data analysis with natural language processing for textual sentiment analysis.

## 2. Core Functionality

### 2.1 Prediction Models

The system employs five distinct machine learning models for structured data prediction:

- Logistic Regression classifier
- Decision Tree algorithm
- Random Forest ensemble method
- Support Vector Machine classifier
- Neural Network deep learning model

### 2.2 Natural Language Processing

The NLP pipeline incorporates:

- Custom text cleaning and normalization
- TF-IDF vectorization for feature extraction
- Logistic Regression for text classification
- VADER sentiment analysis for emotional tone detection
- Educational keyword analysis for context-specific assessment

## 3. Data Processing Pipeline

### 3.1 Input Features

The system processes thirteen structured input features including:

- Academic performance metrics (G1, G2 grades)

- Attendance records and ratios
- Demographic information (age, school)
- Support service indicators (school support, paid classes)
- Behavioral factors (social activity frequency)

### 3.2 Feature Engineering

Automated feature processing includes:

- Binary feature encoding (yes/no → 1/0)
- Categorical variable transformation
- Numerical value normalization
- Derived feature calculation (average grades, attendance ratios)

---

### 3.3 Text Processing

Textual input undergoes comprehensive processing:

- Special character removal and tokenization
- Lowercase conversion and stop-word elimination
- Lemmatization for word normalization
- Sentiment scoring and keyword analysis

## 4. Risk Classification System

Three risk levels are defined: High Risk indicates probability of academic failure requiring immediate intervention; Medium Risk suggests marginal passing probability benefiting from additional support; Low Risk shows high success probability requiring maintenance of current strategies. Predictions include confidence metrics representing model certainty, with neural networks providing probability distributions and scikit-learn models outputting prediction probabilities.

### 4.1 Confidence Scoring

Predictions include confidence metrics representing model certainty:

- Neural networks provide probability distributions
- Scikit-learn models output prediction probabilities
- Confidence scores converted to percentage values

## 5. API Architecture

### 5.1 Endpoint Structure

The system provides multiple API endpoints:

**Primary Prediction Endpoint**

- Accepts form data submissions
- Processes both structured and unstructured data
- Returns rendered HTML response

**REST API Endpoint**

- JSON-based request/response format
- Programmatic access for integration
- Standardized output structure

**Specialized Analysis Endpoints**

- Text-only analysis interface
- Model comparison functionality
- Sentiment analysis service

### 5.2 Response Format

API responses follow consistent structure:

- Prediction results with confidence scores
- Risk category classification
- Model metadata and processing information
- Error handling and validation messages

## 6. Implementation Details

### 6.1 Technology Stack

#### Backend Framework

- Flask web application framework
- Jinja2 templating engine
- RESTful API implementation

#### Machine Learning Libraries

- Scikit-learn for traditional models
- TensorFlow/Keras for neural networks
- NLTK for natural language processing
- TextBlob for sentiment analysis

#### Data Processing

- Pandas for data manipulation
- NumPy for numerical computations
- Joblib for model serialization

### 6.2 Model Management

The system employs persistent model storage:

- Pre-trained model loading during initialization
- Model version management
- Fallback handling for missing components

## 7. User Interface Design

### 7.1 Interface Components

The web interface features tab-based navigation, dynamic form generation, real-time input validation, and automated calculation features. Results presentation includes color-coded risk indicators, confidence level displays, comparative analysis views, and interactive tab navigation.

---

## 9. Operational Characteristics

### 9.1 Performance Metrics

- Real-time prediction processing
- Efficient model inference
- Scalable request handlin

### 9.2 Error Handling

Comprehensive exception management includes:

- Input validation and sanitization
- Model loading fallback procedures
- Graceful degradation for missing components
- Detailed error logging and reporting

### 10. Maintenance Procedures

### 10.1 Model Updates

- Periodic model retraining procedures
- Version control implementation
- Performance monitoring and validation

### 10.2 System Monitoring

- Usage analytics collection
- Performance metric tracking
- Error rate monitoring
- User feedback incorporation

---

## 11. Ethics in AI

The development and deployment of the Student Risk Prediction System requires careful consideration of ethical principles to ensure fairness, transparency, and responsible use.

1. **Data Privacy and Security**
    - All student records are anonymized to protect personal information.
    - No sensitive identifiers such as names, addresses, or IDs are stored in the system.
    - Data handling complies with standard privacy guidelines to prevent misuse.

2. **Bias and Fairness**

    - Machine learning models may unintentionally reflect biases present in training data.
    - Factors such as gender, socioeconomic status, and school type may influence predictions if not properly addressed.
    - Class balancing and monitoring of feature importance are employed to reduce bias.

3. **Responsible Use**

    - The system is intended as a **decision-support tool**, not a replacement for teacher or counselor judgment.
    - Predictions highlight students who may need support, but final interventions remain the responsibility of educators.

4. **Transparency and Accountability**

    - Model predictions include confidence scores to indicate uncertainty.
    - Schools and educators remain accountable for how predictions are applied in practice.
    - Clear documentation of system design and limitations ensures transparency.

By addressing these ethical considerations, the system can support early intervention strategies while safeguarding fairness, privacy, and trust.

---

## 12. Conclusion

This project demonstrates the potential of machine learning and NLP for predicting student performance. While structured datasets capture demographic and academic attributes, NLP analysis of course reviews provides additional insights into student sentiment and risk levels. Both approaches highlight the importance of early detection and data-driven decision-making in education.

## 13. Future Work

- Incorporating ensemble models to improve minority class predictions.
- Using advanced NLP methods (BERT, transformers) for better sentiment understanding.
- Collecting larger, more diverse datasets to generalize the models.
- Integrating real-time prediction systems for institutions.