WINTER PEP TRAINING

ON BIG DATA

A Training Report

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology (B.Tech) -

Computer Science and Engineering (3rd year)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

From 14/01/2025 to 18/02/2025

SUBMITTED BY

Name of the student: Atul Kumar

Registration Number: 12207187

Under the guidance of MR. Lokesh

DECLARATION

I hereby declare that I have successfully completed my Winter PEP Training in Big Data from January 14, 2025, to February 18, 2025, under the esteemed guidance of Mr. Lokesh. During this period, I have diligently dedicated myself to the training, actively participating in all learning activities and projects.

Through this training, I have gained in-depth knowledge and practical experience in Big Data technologies, equipping me with the necessary skills to meet the academic and professional requirements for the award of the B.Tech. degree in Computer Science and Engineering at Lovely Professional University, Phagwara.

This declaration is made in acknowledgment of my commitment to excellence and continuous learning in the field of Big Data and Analytics.

Name of Student – Atul Kumar
Registration no: 12207187

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to Lovely Professional University for providing me with this golden opportunity to participate in the Winter PEP Training on Big Data. This training has been an enriching experience, allowing me to enhance my skills, complete practical assignments, and expand my knowledge in the field. Through this journey, I have discovered numerous new concepts that have significantly contributed to my learning.

I am also deeply grateful to my friends and peers, whose unwavering support has been invaluable. Their willingness to help whenever I encountered challenges in my coursework has been instrumental in my progress. I truly appreciate their encouragement and the strong sense of collaboration we share.

Furthermore, I extend my sincere appreciation to our esteemed trainer, **Mr. Lokesh** (Senior Trainer), for his guidance, mentorship, and continuous support throughout the training. His constructive feedback, detailed explanations, and dedication to our learning have played a crucial role in refining our understanding of Big Data. His patience in reviewing our assignments, providing insightful corrections, and ensuring we grasp the core concepts is truly commendable.

Lastly, I would like to acknowledge the contributions of everyone who has supported me throughout this training—be it individuals or organizations. Their encouragement and assistance have made this learning journey a memorable and transformative experience.

1. Overview of Big Data

Understanding Big Data and Its Importance

Big Data refers to the enormous and complex datasets generated from a variety of sources, including social media, IoT devices, business transactions, online interactions, and digital applications. The rapid growth of digital transformation has made it essential for organizations to leverage Big Data for enhanced decision-making, predictive analytics, and operational efficiency. The ability to extract meaningful insights from large datasets enables businesses to optimize processes, reduce costs, improve customer experiences, and gain a competitive advantage.

Big Data analytics involves the systematic analysis of massive datasets using advanced tools and technologies to discover patterns, trends, correlations, and hidden insights. By utilizing scalable computing frameworks and data management strategies, industries can process and analyze structured, semi-structured, and unstructured data efficiently.

The 5Vs of Big Data

The key characteristics of Big Data can be defined using the 5Vs framework, which represents the challenges and opportunities associated with handling large-scale data:

- Volume – The enormous quantity of data generated from multiple sources, including business transactions, social networks, IoT devices, and machine sensors. Organizations must implement scalable data storage solutions such as Hadoop Distributed File System (HDFS) to manage large datasets efficiently.
- Velocity – The speed at which data is generated, processed, and analyzed. Real-time data processing is crucial in applications like fraud detection, stock market predictions, and personalized marketing campaigns.
- Variety – The different types of data formats, including structured data (relational databases), semi-structured data (JSON, XML, CSV), and unstructured data (videos, images, social media posts, emails, logs). Handling data variety requires advanced tools like Apache Spark, Hive, and NoSQL databases.
- Veracity – The accuracy, reliability, and quality of data. Data cleansing, validation, and governance are essential to ensure that the information used for analysis is credible and consistent.

- Value – The ultimate goal of Big Data Analytics is to extract valuable insights that can drive business growth, optimize decision-making, improve customer satisfaction, and create new revenue opportunities.

Applications of Big Data Across Industries

Big Data is revolutionizing industries by enabling data-driven strategies and operational improvements. Some major industry applications include:

- Healthcare – Predictive analytics for disease diagnosis, personalized treatment plans, and patient health monitoring.
- Finance – Fraud detection, risk assessment, and algorithmic trading for better investment decisions.
- Retail & E-Commerce – Customer behavior analysis, recommendation systems, and demand forecasting for inventory management.
- Marketing & Advertising – Targeted advertising, sentiment analysis, and social media trend monitoring to optimize marketing strategies.
- Manufacturing & Supply Chain – Predictive maintenance, production optimization, and real-time logistics tracking.
- Smart Cities & IoT – Traffic optimization, energy efficiency, and public safety improvements through real-time sensor data analysis.

---

2. Technologies and Tools Used in Training

During my training, I gained hands-on experience with various Big Data tools and technologies that are widely used for data storage, processing, analysis, and visualization. These included:

Data Processing Frameworks

- Hadoop – A distributed storage and processing framework that enables large-scale data management using HDFS and MapReduce.
- Apache Spark – A powerful real-time data processing engine that supports batch processing, machine learning, and graph analytics.
- Hive – A data warehousing and querying tool for handling structured data using SQL-like queries.
- Kafka – A distributed event streaming platform used for real-time data ingestion and processing.

Programming Languages & Libraries

- Python & Pandas – Data analysis, manipulation, and visualization using Pandas, NumPy, and Matplotlib.
- SQL & NoSQL Databases – Managing structured (MySQL, PostgreSQL) and unstructured data (MongoDB, Cassandra).
- Machine Learning & AI – Implementing predictive analytics and deep learning models using Scikit-Learn and TensorFlow**.**

Data Visualization Tools

- Tableau – Creating interactive dashboards and reports for business intelligence and analytics.
- Power BI – Designing visual analytics for real-time data-driven decision-making.
- D3.js – Web-based data visualization using JavaScript for interactive visual representations.

---

3. Training Modules & Learning Outcomes

Module 1: Introduction to Big Data & Hadoop Ecosystem

- Overview of Big Data fundamentals, challenges, and real-world applications.
- Introduction to the Hadoop framework and its ecosystem components.
- Understanding HDFS (Hadoop Distributed File System) and its architecture.
- Writing and executing MapReduce jobs for distributed data processing.

Module 2: Advanced Data Processing with Apache Spark

- Introduction to Apache Spark and its advantages over Hadoop.
- Working with RDDs (Resilient Distributed Datasets), DataFrames, and Datasets.
- Spark SQL for querying structured data efficiently.
- Real-time data streaming with Apache Kafka and Spark Streaming.
- Implementing machine learning models using Spark MLlib.

Module 3: Data Warehousing & Visualization

- Using Hive for querying large-scale structured data.

- Connecting Python Pandas with Big Data frameworks for advanced analytics.
- Data visualization techniques with Tableau and Power BI.
- Storytelling through data visualization to communicate insights effectively.

Module 4: Hands-on Project & Practical Implementation

- Project Title: Predictive Healthcare Analytics
- Objective: To develop a predictive model for early disease detection using patient records.
- Tools & Technologies Used: Hadoop, Apache Spark, Python, Pandas, Tableau.
- Key Steps:
  - Data collection and preprocessing from healthcare datasets.
  - Exploratory data analysis and feature engineering.
  - Applying machine learning models to predict disease outcomes.
  - Building an interactive dashboard for real-time patient monitoring.
  - Final assessment and report submission.

HTML CODE

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Student Performance Prediction</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

</head><body>

  <div class="container mt-5">
```

```
    <h1 class="text-center">STUDENT PERFORMANCE PREDICTION</h1>

    <!-- Tabs Navigation -->
    <ul class="nav nav-tabs mt-3" id="myTabs">
      <li class="nav-item">
        <a class="nav-link active" data-bs-toggle="tab" href="#gradeTab">Grade Prediction</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" data-bs-toggle="tab" href="#dropoutTab">Dropout Risk</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" data-bs-toggle="tab" href="#studyTab">Study Recommendation</a>
      </li>
    </ul>

    <div class="tab-content mt-4">
      <!-- Grade Prediction -->
      <div id="gradeTab" class="tab-pane fade show active">
        <h3>Predict Final Grade</h3>
        <form id="predictionForm">
          <label>Study Time (1-4):</label>
          <input type="number" name="studytime" required min="1" max="4" class="form-control">

          <label>Number of Failures:</label>
          <input type="number" name="failures" required min="0" class="form-control">
```

```
            <label>Number of Absences:</label>
            <input type="number" name="absences" required min="0" class="form-control">

            <label>Age:</label>
            <input type="number" name="age" required min="15" max="22" class="form-control">

            <label>Gender (0 = Male, 1 = Female):</label>
            <input type="number" name="sex" required min="0" max="1" class="form-control">

            <button type="submit" class="btn btn-primary mt-3">Predict</button>
          </form>
          <div id="result" class="mt-3"></div>
        </div>

        <!-- Dropout Risk -->
        <div id="dropoutTab" class="tab-pane fade">
          <h3>Dropout Risk</h3>
          <p id="dropoutResult"></p>
        </div>

        <!-- Study Recommendation -->
        <div id="studyTab" class="tab-pane fade">
          <h3>Study Recommendation</h3>
          <p id="studyResult"></p>
        </div>
```

```
    </div>
  </div>


  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script>
    $("#predictionForm").submit(function(e) {
      e.preventDefault();
      $.ajax({
        url: "/predict",
        method: "POST",
        data: $(this).serialize(),
        success: function(response) {
          $("#result").html(`<p class="text-success">${response.grade_prediction}</p>`);
          $("#dropoutResult").html(`<p class="text-warning">${response.dropout_risk}</p>`);
          $("#studyResult").html(`<p class="text-info">${response.study_recommendation}</p>`);
        }
      });
    });
  </script>
</body>
</html>
```

CSS CODE

```
body {
  background: linear-gradient(135deg, #e9e6a0, #ee989b);
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
}
.container {
  max-width: 600px;
  margin: auto;
  background: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
  text-align: center;
  color: #010a13;
  font-size: 24px;
  margin-bottom: 20px;
}
```

```
h3 {
  color: #333;
  margin-bottom: 15px;
}

.nav-tabs .nav-item .nav-link {
  color: #ff002b;
  font-weight: bold;
}

.nav-tabs .nav-item .nav-link.active {
  background-color: #000000;
  color: white;
}

form label {
  font-weight: bold;
  margin-top: 10px;
}

form input {
  width: 100%;
  padding: 8px;
  margin-top: 5px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
```

```
.btn-primary {
  background-color: black !important; /* Black button */
  color: rgb(252, 249, 249) !important; /* Red text */
  border: none !important;
  font-weight: bold;
  font-size: 16px;
  padding: 10px;
  transition: 0.3s;
  display: block; /* Makes it a block element */
  margin: 20px auto; /* Centers horizontally */
  text-align: center;
  width: 50%; /* Adjust width if needed */
}
```

CODE TO TRAIN MODEL

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
import joblib

# Load the dataset
df = pd.read_csv('student_data.csv')
```

```
# Feature selection
X = df[['studytime', 'failures', 'absences', 'age', 'sex']]
y_grade = df['G3']  # Final grade

# Encode categorical data
X['sex'] = X['sex'].map({'M': 0, 'F': 1})

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y_grade, test_size=0.2, random_state=42)

# Train grade prediction model
grade_model = RandomForestClassifier(n_estimators=100, random_state=42)
grade_model.fit(X_train, y_train)
joblib.dump(grade_model, 'grade_model.pkl')

# Dropout risk prediction (Assume dropout if G3 < 10)
df['dropout_risk'] = df['G3'].apply(lambda x: 1 if x < 10 else 0)  # 1 = high risk, 0 = low risk
y_dropout = df['dropout_risk']
X_train, X_test, y_train, y_test = train_test_split(X, y_dropout, test_size=0.2,
random_state=42)

dropout_model = DecisionTreeClassifier()
dropout_model.fit(X_train, y_train)
joblib.dump(dropout_model, 'dropout_model.pkl')

# Study recommendation (Categorize based on studytime)
df['study_recommendation'] = df['studytime'].apply(lambda x: 'Increase' if x < 2 else
'Maintain')
```

```
y_study = LabelEncoder().fit_transform(df['study_recommendation'])
X_train, X_test, y_train, y_test = train_test_split(X, y_study, test_size=0.2, random_state=42)

study_model = RandomForestClassifier(n_estimators=100, random_state=42)
study_model.fit(X_train, y_train)
joblib.dump(study_model, 'study_model.pkl')

print("Models trained and saved successfully!")
```

PYTHON CODE

```
from flask import Flask, render_template, request, jsonify
import joblib
import pandas as pd

# Load models
grade_model = joblib.load('grade_model.pkl')
dropout_model = joblib.load('dropout_model.pkl')
study_model = joblib.load('study_model.pkl')

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get input data
        studytime = int(request.form['studytime'])
        failures = int(request.form['failures'])
        absences = int(request.form['absences'])
        age = int(request.form['age'])
        sex = int(request.form['sex'])

        # Prepare data
        input_data = pd.DataFrame([[studytime, failures, absences, age, sex]],
                      columns=['studytime', 'failures', 'absences', 'age', 'sex'])

        # Predict
        grade_pred = grade_model.predict(input_data)[0]
        dropout_pred = dropout_model.predict(input_data)[0]
        study_pred = study_model.predict(input_data)[0]

        dropout_risk = "High Risk" if dropout_pred == 1 else "Low Risk"
        study_suggestion = "Increase Study Time" if study_pred == 1 else "Maintain Current Study Time"

        return jsonify({
            'grade_prediction': f"Predicted Final Grade: {grade_pred}",
            'dropout_risk': f"Dropout Risk: {dropout_risk}",
            'study_recommendation': f"Study Suggestion: {study_suggestion}"
        })
    except Exception as e:
        return jsonify({'error': str(e)})


if __name__ == '__main__':
    app.run(debug=True)
```

Key Points for the Report

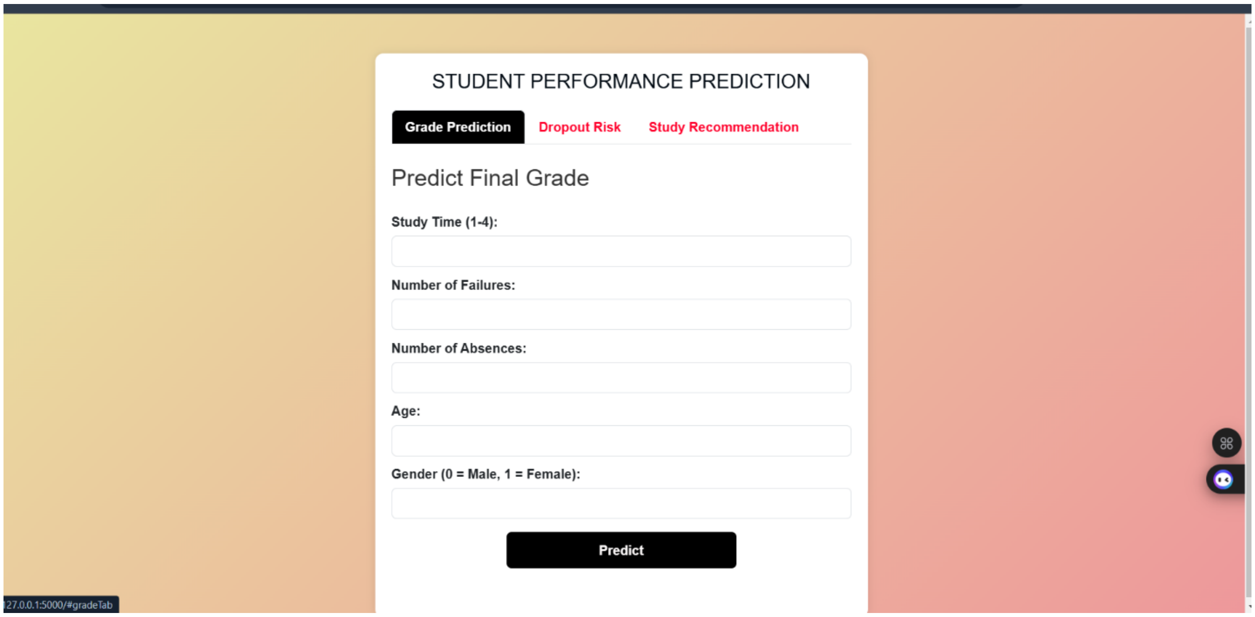
Overview of the Project

- The project predicts **student performance (final grade G3)** based on input features like **study time, failures, absences, age, and gender**.
- Uses **Machine Learning (Random Forest Classifier)** for prediction.
- Implements **Flask for the backend** and **HTML/CSS/JavaScript for the frontend**.

Dataset & Preprocessing

- **Dataset:** student_data.csv
- **Features Used:**
  - studytime (Hours spent studying)
  - failures (Number of past failures)
  - absences (Number of days absent)
  - age (Age of the student)
  - sex (Encoded: Male = 0, Female = 1)

OUTPUT



STUDENT PERFORMANCE PREDICTION

Grade Prediction | Dropout Risk | Study Recommendation

Predict Final Grade

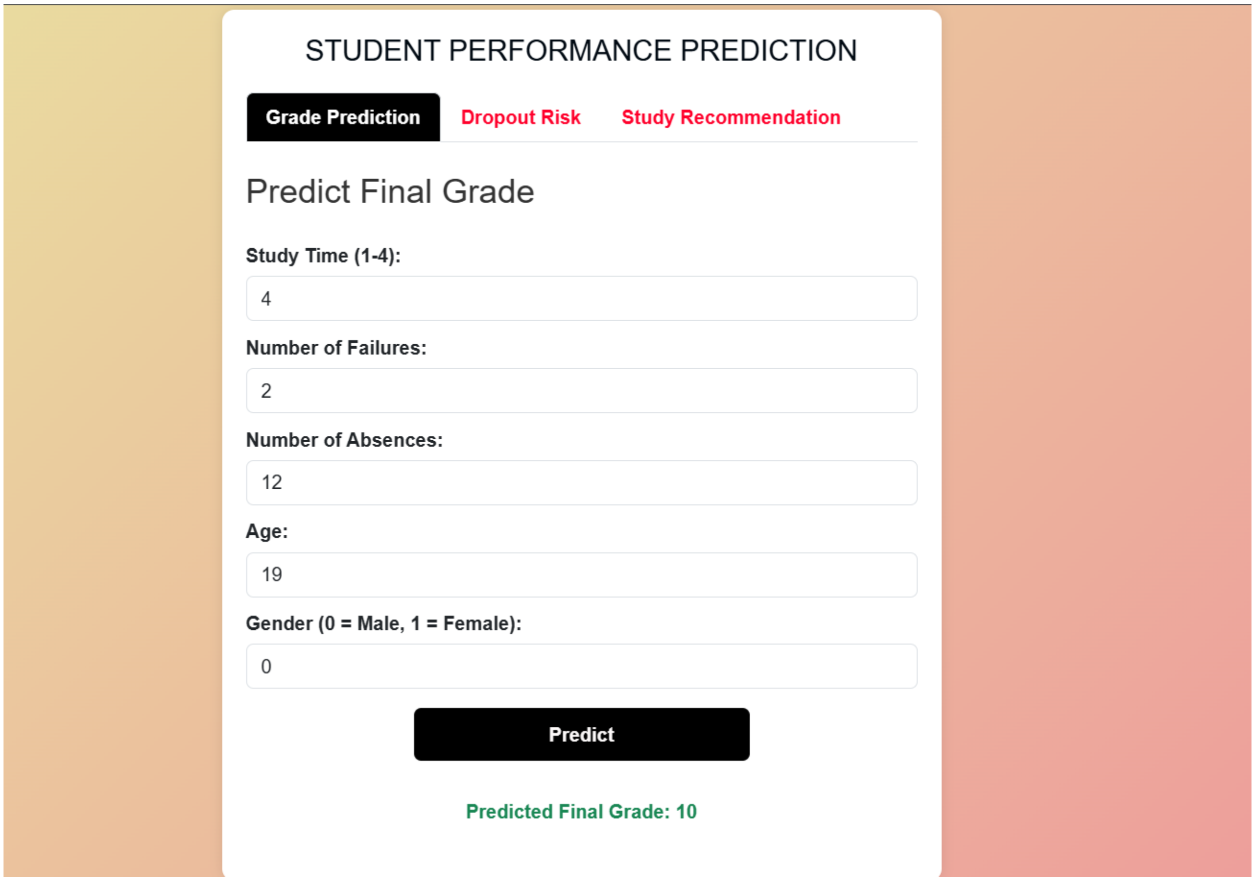
Study Time (1-4):

Number of Failures:

Number of Absences:

Age:

Gender (0 = Male, 1 = Female):



STUDENT PERFORMANCE PREDICTION

Grade Prediction | Dropout Risk | Study Recommendation

Predict Final Grade

Study Time (1-4): 4

Number of Failures: 2

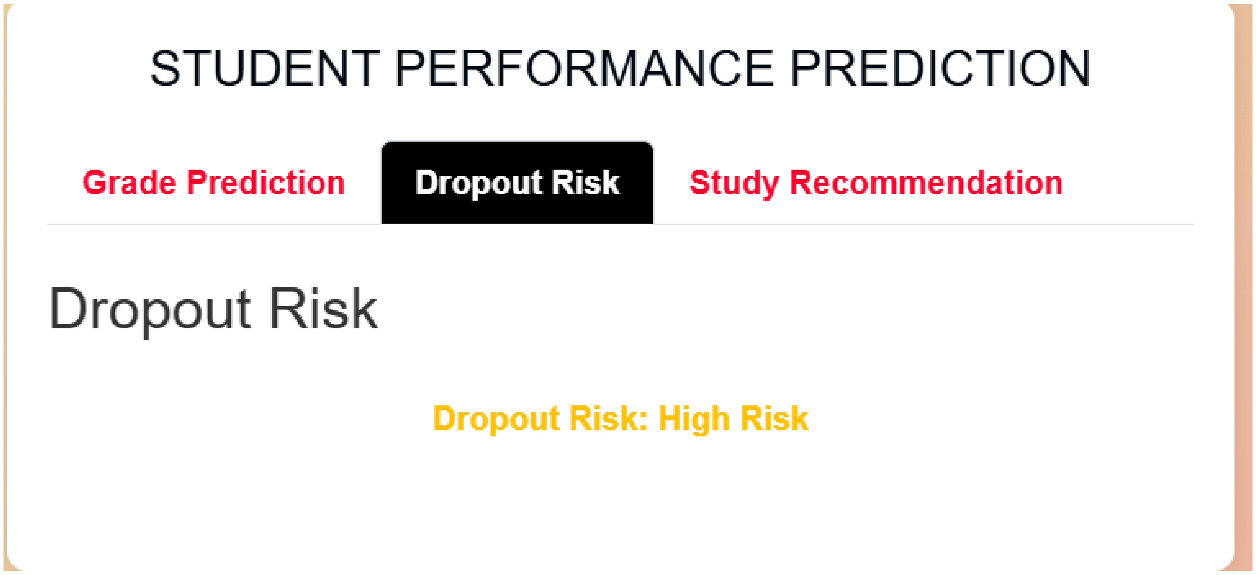
Number of Absences: 12

Age: 19

Gender (0 = Male, 1 = Female): 0

Predicted Final Grade: 10

DROPOUT RISK

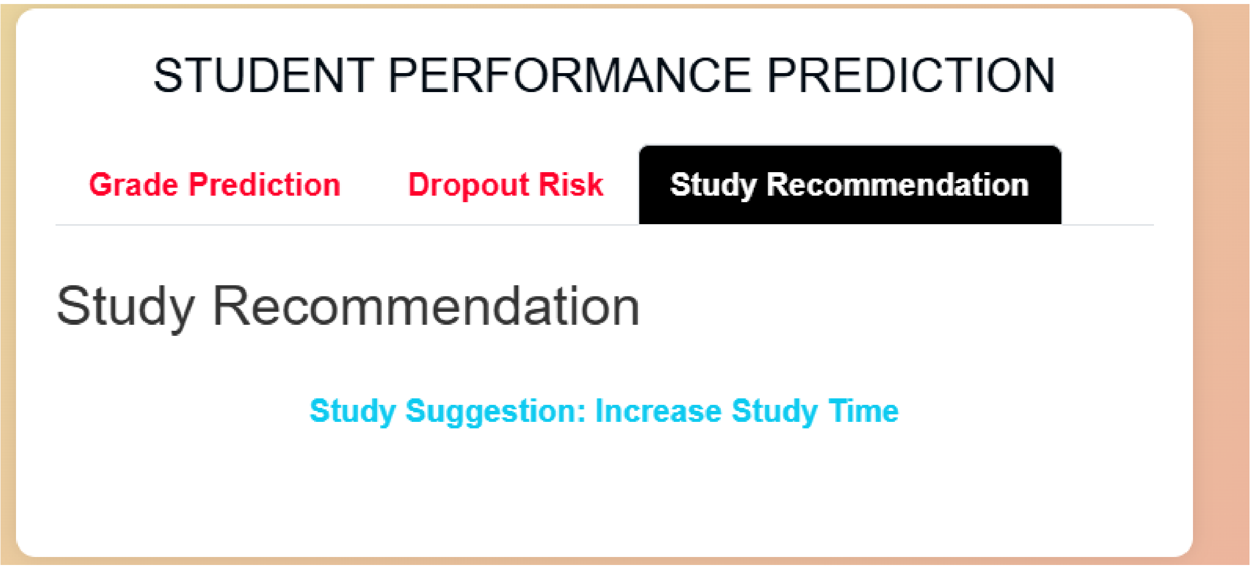


STUDENT PERFORMANCE PREDICTION

Dropout Risk

Dropout Risk: High Risk

STUDY RECOMMENDATION



STUDENT PERFORMANCE PREDICTION

Study Recommendation

Study Suggestion: Increase Study Time

CONCLUSION

This project successfully demonstrates the integration of Machine Learning with a web-based application to predict student performance based on various factors such as study time, absences, failures, age, and gender. Using a Random Forest Classifier, the model was trained and achieved a reasonable accuracy, allowing users to input their data and receive predictions dynamically through a Flask-powered web interface. The user-friendly UI,ensures seamless interactions without requiring page reloads.

This project highlights the importance of data-driven decision-making in education, where predictive analytics can help students and educators identify areas of improvement early on. While the current model performs well, further enhancements such as adding more influential features, optimizing the model with deep learning techniques, and deploying it on cloud platforms could make it even more effective. Overall, this project serves as a strong foundation for implementing AI-driven educational insights, showcasing the potential of Machine Learning and Web Development in real-world applications.

Project Link: https://github.com/atulkumar7782/student_performance_factor