

Week 5: Predictive Analytics for Education

Data Science for Mathematics Teachers

November 11, 2025

Course Information

Course: Data Science for Mathematics Teachers

Series: Professional Development Series

Duration: 8 weeks

Level: Beginner To Intermediate

Target: Mathematics Teachers

1 Week 5: Predictive Analytics for Education

1.1 Learning Objectives

By the end of this week, you will be able to:

- Build models to predict student exam performance
- Identify students who may need additional support
- Create early warning systems for academic intervention
- Understand the ethics of predictive analytics in education

1.2 Topics Covered This Week

- Predicting student performance using regression
- Identifying at-risk students with classification
- Early intervention systems

1.3 Key Concepts You Will Work With

- Linear regression with `sklearn.linear_model.LinearRegression`
- Feature selection and correlation analysis
- Train/test splitting with `{train_test_split()}`
- Model evaluation with `mean_squared_error` and `r2_score`
- Classification with `LogisticRegression` and `DecisionTreeClassifier`
- Confusion matrices and classification reports
- Cross-validation for model reliability
- Handling categorical variables with `pd.{get_dummies()}`

1.4 Practical Exercises

Difficulty Level: Intermediate

Total Exercises: 3

Exercise 1: Concept-Based Challenge: Predicting student performance using regression

Difficulty: Intermediate

Task: Implement a solution to build models to predict student exam performance.

Technical Requirements:

- Use Linear regression with `sklearn.linear_model.LinearRegression` as the primary method
- Integrate Feature selection and correlation analysis for enhanced functionality
- Test your solution with real classroom data
- Document your code with clear comments

Deliverable: A complete Python script that mathematics teachers can use immediately.

Teaching Context: Proactive student support and intervention

Exercise 2: Concept-Based Challenge: Identifying at-risk students with classification

Difficulty: Intermediate

Task: Establish a complete Python workspace for educational use.

Setup Process:

- Install Python and verify the installation
- Use Train/test splitting with `{train_test_split()}` to manage educational packages
- Configure Model evaluation with `mean_squared_error` and `r2_score` for interactive lessons
- Test the environment with sample math problems
- Create a teacher's guide for setup

Goal: A ready-to-use Python environment for mathematics education.

Teaching Context: Proactive student support and intervention

Exercise 3: AI-Enhanced Challenge: Early intervention systems**Difficulty:** Intermediate**AI-Enhanced Programming Exercise****Task:** Using Python, develop a Python solution focusing on classification with `logisticregression` and `decisiontreeclassifier`.**Step-by-Step Instructions:**

1. Focus on implementing classification with `logisticregression` and `decisiontreeclassifier` effectively
2. Create clear, educational examples for student understanding
3. Test your implementation with classroom scenarios
4. Add comprehensive comments for teaching purposes
5. Validate results and create sample outputs

Technical Requirements:

- Use `Classification` with `LogisticRegression` and `DecisionTreeClassifier` in your implementation
- Use Confusion matrices and classification reports in your implementation
- Use Cross-validation for model reliability in your implementation

Expected Output: A working Python script that mathematics teachers can run in their classroom to solve real educational problems.**Assessment:** Your solution should be practical, well-commented, and directly applicable to teaching mathematics.**Teaching Context:** Proactive student support and intervention