

Lesson Plan: Predicting Free Throws with Machine Learning

Duration: 3 classes × 50 minutes

Standards Alignment

NGSS

- HS-ETS1-4: Use a computer simulation to model the impact of proposed solutions to a complex real-world problem.
- Crosscutting Concepts:
 - Cause and Effect: Analyze how input features (e.g., fatigue, distance) affect free throw accuracy.
 - Patterns: Recognize trends in performance data.
 - Systems and System Models: Build and evaluate a predictive system using machine learning.

Common Core State Standards (CCSS)

- CCSS.Math.Content.HSA-CED.A.2: Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes.

AP Computer Science Principles

- Topic 2.3: Extract information from data to support claims and make predictions. Emphasizes data interpretation, modeling, and drawing conclusions from large datasets.

Prerequisites:

- Basic Python programming
- Familiarity with linear equations and slopes
- Algebra 2 recommended (for advanced modeling)

Learning Objectives

By the end of this 3-lesson series, students will be able to:

1. Describe how machine learning can be used to model real-world phenomena.
2. Interpret patterns and relationships in datasets using mathematical and computational thinking.
3. Build, train, and test a basic predictive model using Python and real/simulated sports data.
4. Reflect on the role of data, systems, and modeling in decision-making and prediction.

Lesson 1: Introduction to Data Modeling & ML in Sports

Objectives

- Introduce the concept of machine learning (ML) and its applications in sports.
- Discuss how variables and data features can affect performance.
- Connect math variables to real-world systems.

Standards Emphasized:

- NGSS Crosscutting Concepts: Cause and Effect, Patterns
- AP CSP 2.3: Using data to support predictions
- CCSS.HSA-CED.A.2: Identify variables and relate them through equations

Activities

- Hook (10 min): Ask students to brainstorm: What factors influence whether a basketball player makes a free throw?
- Mini-Lecture (15 min): Introduction to machine learning and real-world examples. Highlight the concept of using data as evidence to make predictions.
- Explore Dataset (20 min): Show example of free throw dataset. Define variables such as:
shot_distance, fatigue_level, practice_hours_last_week, game_pressure
- Wrap-up (5 min): Students write a response to: How could data help improve performance or coaching?

Lesson 2: Build & Train a Free Throw Prediction Model in Python

Objectives

- Learn to preprocess data using Python (with pandas, numpy).
- Train a simple ML model using scikit-learn.
- Evaluate model accuracy using real or simulated test data.

Standards Emphasized:

- NGSS HS-ETS1-4: Using computer simulation to model solutions
- AP CSP 2.3: Build and evaluate models with data

- CCSS.HSA-CED.A.2: Work with equations representing relationships

Activities

- Review (5 min): Recap variables and dataset structure.
- Code Walkthrough (35 min):
 - Load dataset using pandas
 - Preprocess and split into training and testing data
 - Train a RandomForestClassifier or SVC
 - Evaluate using accuracy_score

Sample Code

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X = df.drop('free_throw_made', axis=1)
y = df['free_throw_made']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

- Class Discussion (10 min):
 - What does model accuracy mean?
 - How could this help coaches make decisions?

Lesson 3: Build & Train a Free Throw Prediction Model in Python

Objectives

- Use the trained ML model to make new predictions.
- Connect ML model to math concepts: slope, intercept, error.
- Evaluate the strengths and limitations of a computational model.

Standards Emphasized:

- NGSS Crosscutting Concepts: Systems and Models, Patterns
- AP CSP 2.3: Data interpretation to support predictions
- CCSS.HSA-CED.A.2: Linear models, slopes, and relationships

Activities

- Warm-Up (5 min): Pose a new scenario: A tired player who hasn't practiced is shooting. Will they make it?
- Prediction Task (25 min):
 - Students use Python to input new shot data and use `.predict()` method
 - Compare model's prediction with their own hypothesis

Sample Code

```
new_shot = pd.DataFrame({  
    'shot_distance': 16,  
    'fatigue_level': 8,  
    'practice_hours_last_week': 2,  
    'game_pressure': 7  
})  
prediction = model.predict(new_shot)  
print("Prediction:", "Made" if prediction[0] == 1 else "Missed")
```

- Math Deep Dive (15 min):
 - Introduce linear regression briefly using `LinearFit` or graphing calculator.
 - Discuss slope, intercept, and mean squared error (MSE).
- Wrap-Up Reflection (5 min):
 - How did using a model change your understanding of the math behind prediction?
 - Where else could we use models like this?

Assessment

- Formative: Student reflections, exit slips, accuracy of prediction tasks.
- Summative: Mini-report or reflection:
 - Describe how ML was used to model free throws.
 - Explain which variable most influenced accuracy.
 - Connect model outcomes to slope, intercept, and prediction error.