Lesson 15: Build and Evaluate Your Predictive Model

Lesson Description:

In this lesson, students will implement the modeling plan they created in Lesson 14 by building and evaluating two machine learning models using Azure ML Designer: one linear regression model and one decision tree model. Students will analyze how well each model fits their data, interpret outputs such as accuracy and error, identify outliers, and explain what the models reveal about player performance in their dataset. This technical execution phase is critical to their final project and prepares them to refine their models in Lesson 16.

Main Learning Goal:

Students will run both a linear regression model and a decision tree model on their dataset, interpret the results, identify outliers, and explain how the models make predictions.

Essential Question:

How can we use regression and decision trees to make accurate predictions and uncover meaningful patterns in sports data?

Standards:

- AP6.2 – Design, build and test level-appropriate algorithms that use linear regression and describe existing outliers in the context of the programming solution.
- AP6.2 – Use decision trees and linear regressions to make predictions and classifications.

Objectives:

By the end of this lesson, students will:

- Build both a linear regression and a decision tree model using Azure ML Designer
- Visualize and interpret key model outputs (R^2, RMSE, AUC, confusion matrix)
- Identify and describe outliers or misclassifications

- Compare model results and reflect on effectiveness

What Makes a Model Work?

Why Did the Model Get It Wrong?

Imagine this: You just built a machine learning model to identify **sluggers**—players who hit for power. One player has:

- A **.310 batting average**
- A **.500 slugging percentage**
- **6 home runs**

But the model predicts: **“Not a slugger.”**
That doesn’t feel right, does it?

- **What to do:**
    - Think silently for 30 seconds: Why might the model make this mistake?
    - Then, turn to a partner or your group and share your ideas.
- **Need help thinking? Consider these possibilities:**
    - Maybe 6 home runs isn’t enough compared to other players in the training set.
    - Maybe the player only played a few games, so the model didn’t trust the stats.
    - Maybe the model didn’t pick the right features to focus on.

You don’t have to be “right” here—just be curious. Models often make mistakes, and your job is to understand why.

Let’s Dig Deeper: How Models Think

Now let’s figure out what’s going on under the hood. You’ll talk through these questions as a class:

- **What might cause the model to make this kind of mistake?**
  Think about issues like biased training data, weak patterns, or noisy stats.
- **What if the player was just below the slugger cutoff?**
  Some models make hard cutoff decisions. If it learned “slugger = 7+ home runs” and your player has 6, they get marked “not a slugger”—even if they’re close.

- **What happens when the dataset is unbalanced?**
  If most players in the dataset are “not sluggers,” the model might start guessing “not slugger” most of the time—even if that’s not helpful.
- **How would a regression model handle this differently than a decision tree?**
  - A decision tree might say “If HR ≥ 7 → slugger.”
  - A regression model might say “This player has a 48% chance of being a slugger.”
    Both are useful—but in different ways.

As you talk, keep an ear out for these terms:
Threshold, Decision Boundary, Class Imbalance, Feature Weight, Outlier, Edge Case, False Positive, Model Confidence

Before you jump into Azure ML Designer, it helps to sketch out how your models will actually work. You’ll build **two different machine learning pipelines**—one for each model type.

Your Task:

- Draw one pipeline for a **regression model** (like predicting ERA_2024 or OPS_2024)
- Draw one pipeline for a **classification model** (like predicting slugger or ace_pitcher)

Each pipeline should include these steps:

1. Import Data
2. Clean Missing Data
3. Select Columns in Dataset
4. Split Data
5. Train Model
   - (Mark which model: Linear Regression or Decision Forest)
6. Score Model
7. Evaluate Model

Tips:

- Use arrows to show how the steps connect.
- Label each diagram at the top:
    - “Regression Model – Predicting [your label]”
    - “Classification Model – Predicting [your label]”

Build Your Two Azure ML Pipelines

You’re putting your ideas into action. You’re designing and building two machine learning pipelines in **Azure ML Designer**. These pipelines will help you answer real questions like:

- “How many home runs might this player hit next season?”
- “Does this player qualify as a slugger or not?”

You’ll create **two supervised learning models**:

- A **linear regression model** to predict a number (like OPS or ERA).
- A **decision tree model** to classify players (like slugger = yes/no).

Your job isn’t just to make the model run — your job is to:

- Choose meaningful input features
- Interpret model results and metrics
- Identify errors or surprises
- Compare which model is more useful for your goal

This is a solo challenge. You’ve already made decisions in Lesson 14 — now you’ll test them in action.

Your Two Pipelines

You will build **two separate pipelines** using the dataset that you chose in the previous lesson and label. Each model tells you something different — and your goal is to compare them.

Pipeline 1 – Linear Regression Model

Goal: Predict a number (a continuous value with decimals)
Example Labels:

- OPS_2024 – On-base Plus Slugging
- ERA_2024 – Pitching ERA
- Total_Points – Football scoring estimate

Example Features:

- Batting: OBP_2023, HR_2023, SLG%_2023
- Pitching: WHIP_2023, K_2023, BB_2023
- Football: Yards_Gained, Completion%, Rushing_TDs

Model to use in Train Model:

- Linear Regression

What you'll learn:

- Does your model explain the number well? (Check R^2)
- Are the prediction errors small or large? (Check RMSE)
- Are the errors meaningful in the real world?

Pipeline 2 – Decision Tree Model

Goal: Classify into two categories (yes/no or 1/0)
Example Labels:

- slugger – 1 = yes, 0 = no
- ace_pitcher – meets pitching threshold or not
- is_qb1 – primary quarterback or not

Example Features:

- Batting: HR, SLG%, AVG
- Pitching: K, ERA, WHIP
- Football: TDs, Pass_Yards, Rush_Yards

Model to use in Train Model:

- Decision Forest Classifier
  (If you want to compare trees to regression, use Decision Forest Regression)

What you'll learn:

- How does the model split features into rules?
- Which features matter most?
- How accurate are its yes/no decisions?

Key Differences to Notice

Concept	Linear Regression	Decision Tree Classification
Prediction Type	Numeric (e.g., 0.532 OPS)	Binary class (e.g., "slugger = yes")
Evaluation Metrics	RMSE, R^2	Accuracy, AUC, Confusion Matrix
How It Learns	Fits a line to the data	Builds a tree of rules and splits
What Can Go Wrong	Outliers cause large errors	Misclassifies players near the cutoff
Explanation Style	Feature weights	Feature importance + split logic

Get Ready — Check Your Plan

Before jumping into Azure ML Designer, review what you prepared in Lesson 14. You should already know:

- Your **label** (the thing you want to predict)
- Your **model type** (regression or classification)
- Your **input features** (which stats matter)

Double-check:

- Are you using only **past-year stats** to predict future ones?
  (Example: To predict OPS_2024, use HR_2023, not HR_2024)
- Did you remove **irrelevant or text columns** like player names or teams?

- Are your features **numeric and meaningful**?

Your Pipeline Structure

All models should follow this 7-step structure in Azure ML Designer:

Step	Azure Component	Purpose
1	Import Data	Load your cleaned dataset
2	Clean Missing Data	Remove or fill any blanks
3	Select Columns	Choose your input features and label column
4	Split Data	Divide into training (80%) and testing (20%) sets
5	Train Model	Use Linear Regression or Decision Forest Classifier
6	Score Model	Predict the label using your test data
7	Evaluate Model	Review metrics like RMSE, R^2, AUC, or confusion matrix

Label your pipelines clearly:

- "Pipeline 1 – Linear Regression (OPS_2024)"
- "Pipeline 2 – Decision Tree Classification (Slugger)"

Troubleshooting Tips

If your pipeline doesn't work or gives unexpected results:

- Did you select the correct **label column** in Train Model?
- Are your **features numeric and relevant**?
- Did you connect the Split Data component correctly?
  (One path to Train Model, the other to Score Model)

Check your Engage sketch if needed. Try solving it yourself first before asking for help.

What to Look for in Azure Results

For Linear Regression:

- **RMSE (Root Mean Squared Error):** Lower = better. Compare it to your label's range.
- **R^2 (R-squared):** Shows how much of the label's variation your model explains.
- **Plot:** Check whether your predicted values line up with actual values.

For Decision Tree Classification:

- **Confusion Matrix:** Where is your model right or wrong?
- **AUC:** Measures how well your model separates categories.
- **Accuracy, Precision, Recall:** Each metric gives different insights.

Example – Regression Model:

You build a model to predict OPS_2024 from HR_2023, SLG%_2023, and OBP_2023.
Results:

- RMSE = 0.032
- R^2 = 0.81

Your model is highly accurate and explains most of the variation in OPS.

Example – Classification Model:

You build a decision tree to classify slugger.
Results:

- True Positives = 12
- False Negatives = 3
- AUC = 0.87

Your model is strong but missed 3 actual sluggers. Look closely at why.

Interpret Results and Identify Outliers

You've now built and tested two different machine learning models. This step helps you go beyond just checking accuracy — it's about understanding *why* your models worked the way they did, and what they struggled with. Outliers and misclassifications are not just "errors" — they're clues that reveal where your model logic, features, or data might need revision. These insights will help you improve your pipeline in Lesson 16.

Your Task: Analyze Your Results

Use the prompts below to guide your individual written analysis. Be specific, reference your actual Azure outputs, and include both models.

1. **What did your models predict?**
    - Identify your label for each model (e.g., OPS_2024, slugger, ace_pitcher)
    - Was it a regression or classification task?
    - Briefly explain how your label and model choice connect to the plan from Lesson 14.
2. **How well did each model perform?**
    - **For regression:** Report RMSE and R^2.
    - **For classification:** Report Accuracy, AUC, and briefly interpret your Confusion Matrix.
    - Explain what these numbers tell you. Was the model performance strong? Were you surprised by any results?
3. **What features were most important?**
    - Use the Feature Importance chart in Azure ML Designer (especially for the decision tree).
    - List the top contributing features.
    - Did any surprise you? Why might the model rely on those features?
4. **Did any players stand out as unusual cases?**
    - **Regression:** Who had the largest prediction errors?
    - **Classification:** Who was misclassified? (e.g., predicted "not a slugger" but actually was.)
    - Reflect on why the model might have struggled with those examples.
5. **What might explain those results?**

- Consider if there were missing values, weird stat combinations, or unexpected patterns.
- Was your model too simple? Were key features missing?
- What would you improve in your next version?

Optional Peer Discussion

Compare your findings with a classmate. Talk through:

- Which model gave better results — and why?
- Did either of you discover a surprising outlier or error?
- What's one modeling choice (label, features, algorithm) you would change in Lesson 16?

Final Reflection

After writing your analysis and (optionally) discussing with a peer, complete a short written reflection to summarize what you learned. Keep your answers focused but thoughtful.

Reflection Prompts:

- **What label did your models try to predict, and why did you choose it?**
- **Which features seemed most important? Were you surprised by their impact?**
- **What did your best model do well — and where did it struggle?**
- **Describe one specific outlier or misclassification and what you think caused it.**
- **If you were to rebuild this pipeline, what would you change to make your results better?**