Let’s Talk Science Workshop: Can We Predict Diabetes Using Data?"

Activity Overview

- Duration: 90 minutes
- Audience: High school students (grades 10–12)
- Tools Needed: Google Colab or Jupyter Notebook, internet access, Python environment (optional pre-setup link)

PIMA dataset

In the 1970s and 80s, researchers from the U.S. National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) began collecting health data from Pima Indian women to study Type 2 diabetes risk. The Pima people are a group of Native Americans who live in the southwestern United States. The dataset includes health-related features like the number of pregnancies, glucose levels, blood pressure, BMI, age, and a diabetes pedigree function (estimates family history of diabetes). Each record includes whether the person was diagnosed with diabetes (1) or not (0). This dataset illustrates how health data can be used to develop evidence-based tools to inform healthcare decisions.

PART 1: Pre-Activity Questions

- What do you think could help doctors know if someone has diabetes?
- What could go wrong if a computer tells someone they don’t have diabetes — but they actually do?

Machine learning fundamentals:

What is machine learning and when should it be used?

- Machine learning (ML) is a branch of artificial intelligence where computers learn patterns from data to make predictions or decisions — without being explicitly programmed for every possible scenario.

What are the types of machine learning?

- Supervised learning: The model is trained on labeled data (e.g., “has diabetes” vs “does not”).
- Unsupervised learning: Finds hidden patterns in data without labels (e.g., customer clustering).
- Reinforcement learning: The model learns by trial and error (e.g., game-playing agents).

What is logistic regression?

- Logistic regression is a machine learning algorithm used for classification problems — where we want to predict a category or label (like has diabetes/doesn’t have diabetes).
- It’s used to classify data into discrete classes, not predict continuous numbers
- The diabetes model uses supervised machine learning — specifically logistic regression — to learn patterns (like high glucose and BMI) that are associated with having diabetes.

- Logistic regression uses a sigmoid function to turn numbers into probabilities between 0 and 1.
    - If the probability is above a threshold (usually 0.5), it predicts class 1; otherwise, it predicts class 0.

$$P(Y=1) = \frac{1}{1+e^{-z}} \quad \text{where } z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Each x_number is a “feature” and every beta_number is its corresponding weight. Beta_0 is the “bias”

For our dataset, we are using logistic regression to classify whether a patient has diabetes or not based on their health features (like glucose, BMI, etc.). This is a binary classification task (two possible outcomes). The model outputs a probability — and we decide using a threshold (e.g., 0.5 or 0.3)

PART 2: Explaining the “logic” behind our code

2.1: Let’s evaluate the quality of our dataset.
Before we build any model, we need to make sure our dataset is clean, reliable, and representative.

What are some things we should verify in assessing the quality of the PIMA dataset?

- Do we have missing data? This can be in the form of blank values, zeros (where zeros don’t make sense), special codes (like 999 or -1), NaN (Not a Number).
- In the PIMA Diabetes exercise, columns like Glucose, BloodPressure, BMI, Insulin have suspicious zeros. For example, a BMI of 0 or glucose of 0 is medically unlikely and probably means missing data.

Does one class appear way more than the other?
HINT: What does “ print(y.value_counts(normalize=True))” print?

How can we address problems with our dataset? What are the tradeoffs?

- Drop missing rowsSimple, but can lose valuable data
- Impute with mean/median: Fill missing values with the average or median
- Use domain knowledge: Replace with medically appropriate defaults or assumption
- Flag missing data: Add a new column like glucose_missing = 1 if it was missingOnly ~35% of the patients have diabetes, meaning it's an imbalanced dataset. That’s why you focused on recall instead of accuracy.

What are parameters vs hyperparameters?
Parameters:
These are the values learned by the model during training.

In logistic regression, parameters are:

- The coefficients (weights) for each feature
- The intercept (bias term)

Hyperparameters:
In your Pima diabetes model, these were the coefficients that told us how important features like Glucose, BMI, and Age were in predicting diabetes. They control how the model learns. Examples for logistic regression:

- C: Controls the strength of L2 regularization
- max_iter: How many iterations the model runs to converge
- solver: The optimization algorithm used (e.g., 'lbfgs', 'saga')

Feature engineering: how to deal with categorical and numerical data

Feature Type	**Examples**	**How to Handle**
Numerical	Age, BMI, Glucose, Blood pressure	Scale it: StandardScaler()
Categorical	Gender, Region, Smoking status	Convert to numbers using encoding methods

All our features are numerical, so we must scale them so they're on the same scale.

- scaler = StandardScaler()
- X_scaled = scaler.fit_transform(X)

This is important because features like Glucose (0–200) and DiabetesPedigreeFunction (0–2.5) are on different ranges.

How can we evaluate the quality of our logistic regression model?
Answer: It depends! Consult the table below.

Metric	Meaning	Why it Matters
Accuracy	% of total predictions that are correct	Misleading if the classes are imbalanced (like in diabetes detection)
Precision	% of predicted positives that are actually positive	Helps when false positives are costly

Recall	% of actual positives that were correctly identified	Helps when false negatives are dangerous (like missing a diagnosis)
F1 Score	Harmonic mean of precision and recall	Useful when you want a balance of precision & recall
Confusion Matrix	A table showing TP, FP, FN, TN	Helps visualize the kinds of errors the model is making
Average Precision (AP)	Area under the precision-recall curve	Summarizes model performance across all thresholds

How should we address class imbalances in our PIMA dataset?
A class imbalance happens when one category (or class) appears much more often than the other in your dataset.
0 (no diabetes): ~65%
1 (has diabetes): ~35%
We should avoid using “accuracy” as our evaluation metric. Indeed, if our model just guesses “no diabetes” every time, it’s still 65% accurate — but totally useless.

We can also adjust the threshold to lower the decision boundary to catch more positives.

What are false positives? What are false negatives? What tolerance do we have for each one?

- False Positive (FP): The model predicts a person has diabetes, but they actually don’t
- False Negative (FN): The model predicts a person does NOT have diabetes, but they actually do.

Confusion Matrix

	Predicted: No	Predicted: Yes
Actual: No	True Negative	False Positive
Actual: Yes	False Negative	True Positive

We emphasize recall because we are more afraid of false negatives.. That’s why we experimented with lowering the threshold to catch more diabetics.

How does a change in threshold alter the FPR and the FNR?
Decreasing the classification threshold tends to increase the number of false positives and decrease the number of false negatives.

Post-activity recap

- What happens if the threshold is set too low?