Linear and Logistic Regression: Foundations of Predictive Modeling

Introduction

Welcome to Week 2 of our AI course! Today we're exploring two fundamental algorithms that serve as building blocks for predictive modeling: **Linear Regression** and **Logistic Regression**. These techniques help us make **predictions based on data** and form the foundation for many advanced machine learning methods.

1. Understanding Regression: The Big Picture

Regression helps us understand and **predict relationships between variables**:

- **Linear Regression**: **Predicts a number** (like price, temperature, or weight)
- **Logistic Regression**: **Predicts** the **probability of something happening** (usually a yes/no outcome)

Real-Life Examples:

- **Linear Regression**: Predicting house prices based on size, location, and number of rooms
- **Logistic Regression**: Predicting whether a customer will purchase a product or not

💡 **Interactive Question #1**: Which type of regression would be best for predicting a student's test score based on hours of study?

A) Logistic Regression

B) Linear Regression

C) Neither regression type is suitable

D) Both types would work equally well

2. Linear Regression: Finding the Line that Fits

2.1 The Basic Idea

Linear regression **finds the straight line** that **best represents the relationship between your input** (**what you know**) and **output** (**what you want to predict**).

Imagine plotting points on a graph and drawing a straight line that comes as close as possible to all points. This line becomes your prediction model.

A simple example: Predicting house prices based on square footage. As square footage increases, house prices tend to increase too - we can capture this relationship with a line.

2.2 The Simple Formula

The formula for a line in linear regression is:

$$y = b_o + b_1x$$

Where:

- y is what we're trying to predict (like house price)
- x is our input feature (like square footage)
- b_o is the y-intercept (where the line crosses the y-axis)
- b_1 is the slope (how steep the line is)

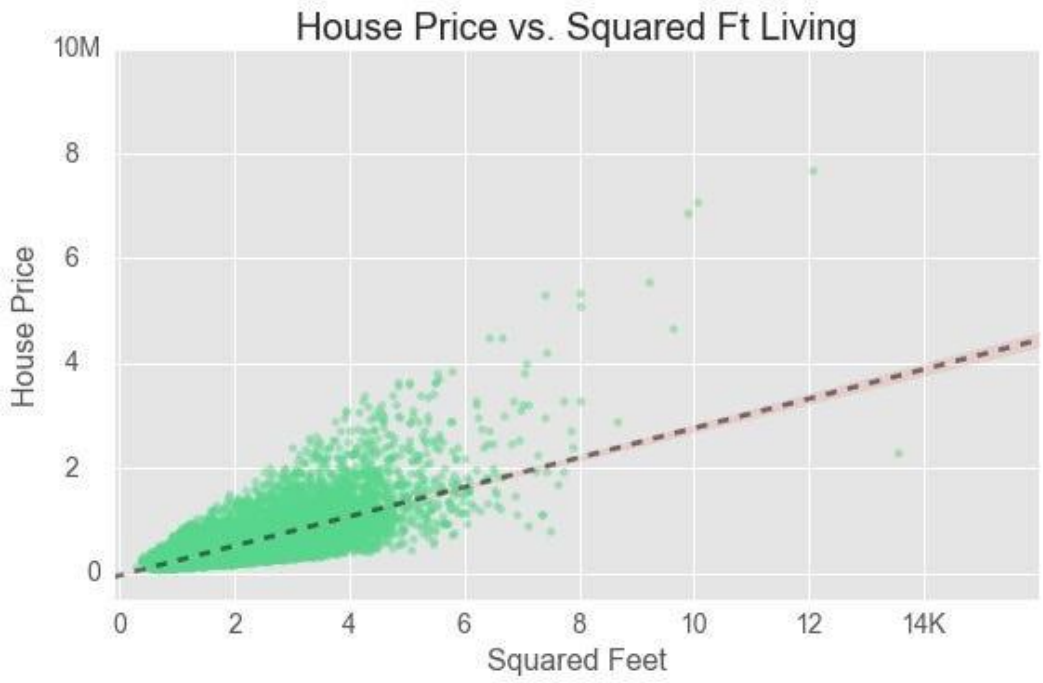

Image Source: https://datalesdatales.medium.com/predicting-house-prices-with-linear-regression-595422992c48

2.3 Multiple Features

When using **more than one input** (like house size AND number of bedrooms), we call it **Multiple Linear Regression**:

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n$$

Each feature has its own coefficient (b) that tells us **how important that feature is**.

💡 **Interactive Question #2**: In a house price prediction model, if the coefficient for "number of bathrooms" is 5000, what does this mean?

A) All houses have 5000 bathrooms

B) On average, each additional bathroom adds $5000 to the predicted price

C) Bathrooms are 5000 times more important than other features

D) The model is 5000% accurate

2.4 How Good is Our Line?

To measure **how well our line fits the data**, we use:

- **R-squared (R^2)**: A percentage (0-100%) that tells us how much of the output variation our model explains. **Higher is better!**
- **Mean Squared Error (MSE)**: The average of squared differences between predictions and actual values. **Lower is better!**

💡 **Interactive Question #3**: A student develops a model to predict test scores. The model has an R^2 of 0.64. What does this mean?

A) The model gets 64% of predictions exactly right

B) The model explains 64% of the variation in test scores

C) The model is accurate 64% of the time

D) 64% of students' scores were correctly predicted

3. Logistic Regression: Predicting Yes or No

3.1 When to Use Logistic Regression

Use logistic regression when you want to predict:

- Whether a customer will buy a product (yes/no)
- If an email is spam (yes/no)
- Whether a student will pass a course (yes/no)

3.2 How It Works

Instead of predicting a continuous number, logistic regression **predicts the probability of something happening** (**a value between 0 and 1**).

It uses a special S-shaped curve called the **sigmoid function** to transform a linear equation into probability values:

The sigmoid function formula is:

$$p = 1 / (1 + e^{(-z)})$$

Where:

- p is the probability (between 0 and 1)
- e is a mathematical constant (approximately 2.718)
- z is our linear model ($b_o + b_1x_1 + b_2x_2 + ...$)

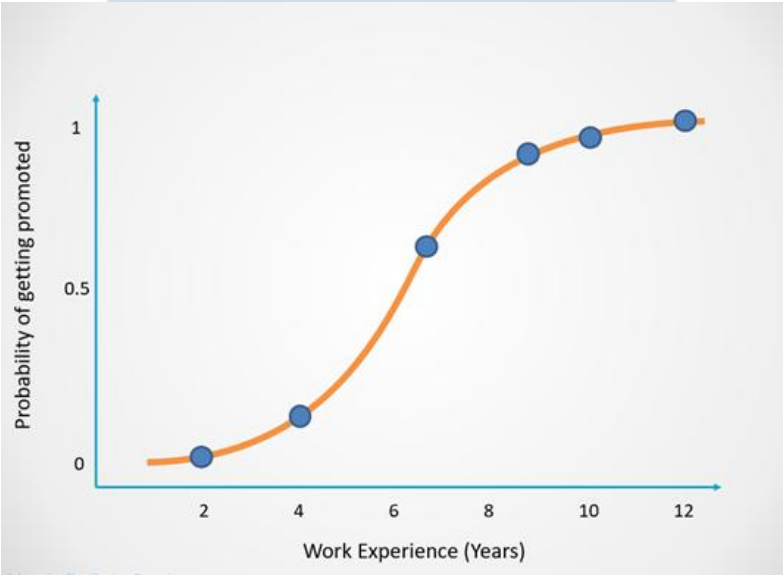

Image Source: https://medium.com/data-science/logistic-regression-explained-in-7-minutes-f648bf44d53e

This S-shaped curve has important properties:

- It always outputs values **between 0 and 1** (perfect for probabilities)
- Probability near 0 = likely "no"
- Probability near 1 = likely "yes"
- Probability of 0.5 = the dividing line

The sigmoid function is what allows logistic regression to take any range of input values and convert them to meaningful probabilities.

3.3 Making Decisions

To convert probabilities into yes/no predictions, we set a **threshold** (usually 0.5):

- If probability > 0.5 → Predict "Yes"
- If probability ≤ 0.5 → Predict "No"

💡 **Interactive Question #4**: A logistic regression model predicts a 0.75 probability that a customer will subscribe to a service. What does this mean?

A) 75% of similar customers will subscribe

B) The model predicts the customer will subscribe (above the 0.5 threshold)

C) The model is 75% certain of its prediction

D) The customer will subscribe to 75% of available services

4. Preparing Data for Regression

Before building regression models, we need to prepare our data:

4.1 Handling Non-Numeric Data

Regression models work **with numbers**. For categories (like color or city), we need to **convert them to numbers**:

- **Create separate yes/no columns** for each category
- **Assign numeric codes** to ordered categories

4.2 Scaling Features

It's often helpful to bring all features to a similar scale:

- Scaling values between 0 and 1
- Standardizing data to have an average of 0

4.3 Missing Values

Options for handling missing data include:

- Removing rows with missing values (if there are few)
- Filling in missing values with averages
- Using special techniques to estimate missing values

💡 **Interactive Question #5**: When preparing data for a regression model, what should you do with categorical data like "shirt size" (S, M, L, XL)?

A) Remove it from the dataset

B) Use it as is since regression can handle text data

C) Convert it to numeric values or create separate columns for each size

D) Average the sizes numerically

5. Building Regression Models in Python

5.1 Scikit-learn Library for Regression

Before we look at the code examples, let's understand the scikit-learn library we'll be using:

Scikit-learn (from sklearn...):

- A powerful machine learning library that makes implementing algorithms simple
- Provides consistent interfaces for different models, making it easy to try different algorithms
- For regression tasks, we'll use these specific modules:
    - **sklearn.linear_model**: Contains **LinearRegression** and **LogisticRegression** classes
    - **sklearn.model_selection**: Contains **train_test_split** for dividing data into training and testing sets
    - **sklearn.metrics**: Contains functions for evaluating model performance (accuracy, etc.)

Scikit-learn has a consistent workflow for most models:

1. Import the model class you need
2. Create an instance of the model
3. Fit the model to your training data using .fit()
4. Make predictions using .predict()

This consistency makes it easy to learn and apply different algorithms as you advance your skills.

5.2 Linear Regression Example

Let's use a simple dataset to predict student test scores based on hours studied and previous GPA:

No Copy-Paste, Only Typing

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Sample dataset: hours studied, previous GPA, and final test score
data = {
    'hours_studied': [2, 3, 5, 4, 3, 1, 7, 8, 5, 6, 7, 2],
    'previous_gpa': [3.2, 3.5, 3.8, 3.3, 3.7, 2.9, 3.9, 4.0, 3.6, 3.8, 3.5, 3.0],
    'test_score': [65, 72, 88, 75, 81, 60, 95, 98, 85, 90, 92, 68]
}

# Create DataFrame
df = pd.DataFrame(data)

# Features (X) and target (y)
X = df[['hours_studied', 'previous_gpa']]
y = df['test_score']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Examine coefficients
print("Intercept:", model.intercept_)
print("Coefficients:", model.coef_)


# Make predictions
predictions = model.predict(X_test)

# Sample prediction
new_student = np.array([[4, 3.5]])  # 4 hours studied, 3.5 GPA
predicted_score = model.predict(new_student)
print(f"Predicted score: {predicted_score[0]:.1f}")
```

No Copy-Paste, Only Typing

5.2 Logistic Regression Example

Now let's use a dataset to predict whether a student will pass (1) or fail (0) based on hours studied and previous GPA:

No Copy-Paste, Only Typing

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Sample dataset: hours studied, previous GPA, and pass/fail (1/0)
data = {
    'hours_studied': [2, 3, 5, 4, 3, 1, 7, 8, 5, 6, 7, 2],
    'previous_gpa': [3.2, 3.5, 3.8, 3.3, 3.7, 2.9, 3.9, 4.0, 3.6, 3.8, 3.5, 3.0],
    'pass': [0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0]
}

# Create DataFrame
df = pd.DataFrame(data)

# Features (X) and target (y)
X = df[['hours_studied', 'previous_gpa']]
y = df['pass']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the model
log_model = LogisticRegression()
log_model.fit(X_train, y_train)

# Make predictions
predictions = log_model.predict(X_test)
probabilities = log_model.predict_proba(X_test)

# Sample prediction
new_student = np.array([[4, 3.5]])  # 4 hours studied, 3.5 GPA
pass_probability = log_model.predict_proba(new_student)[0][1]
print(f"Pass probability: {pass_probability:.1%}")
```

No Copy-Paste, Only Typing

💡 **Interactive Question #6**: In the Python code above, what does test_size=0.2 mean?

A) The model will be 20% accurate

B) 20% of the data will be used for testing the model

C) The test will run 20% faster

D) The model will be trained 20% of the time

6. When to Use Each Regression Type

Use Linear Regression when:

- You're predicting a number (price, temperature, age, score)
- There's a roughly linear relationship between inputs and outputs
- You want to understand how each feature affects the prediction

Use Logistic Regression when:

- You're predicting a yes/no outcome (buy/don't buy, pass/fail, click/don't click)
- You need the probability of something happening
- The relationship between features and outcome can be separated by a boundary

💡 **Interactive Question #7**: Which regression type would be best for predicting whether a bank loan will be repaid or defaulted?

A) Linear Regression

B) Logistic Regression

C) Both would work equally well

D) Neither would be appropriate

Summary

Linear and logistic regression are powerful tools for making predictions based on data. Linear regression helps us predict numbers by finding the best-fit line through our data points. Logistic regression helps us predict yes/no outcomes by calculating probabilities.

Both methods require proper data preparation, including handling non-numeric data and addressing missing values. Though they're among the simplest machine learning algorithms, they remain essential tools that provide clear, interpretable results and serve as building blocks for more complex methods.

Understanding these fundamental techniques prepares you for exploring more advanced machine learning algorithms in future lessons.

Glossary

- **Coefficient**: A number that shows how much the prediction changes when a feature increases by one unit
- **Feature**: An input variable used to make predictions (also called a predictor or independent variable)
- **Intercept**: Where the regression line crosses the y-axis when all predictors are zero
- **Probability**: A number between 0 and 1 representing the likelihood of an event occurring
- **R-squared**: A measure of how well the model explains the variation in the target variable
- **Regression**: A statistical method for estimating relationships between variables
- **Residual**: The difference between a predicted value and the actual value
- **Threshold**: In logistic regression, the cutoff point (usually 0.5) for converting probabilities to yes/no predictions

Multiple Choice Answers

1. B) Linear Regression
2. B) On average, each additional bathroom adds $5000 to the predicted price
3. B) The model explains 64% of the variation in test scores
4. B) The model predicts the customer will subscribe (above the 0.5 threshold)
5. C) Convert it to numeric values or create separate columns for each size
6. B) 20% of the data will be used for testing the model
7. B) Logistic Regression