# Block 3: From Data to Insights - Predictive Modeling

## Python Module for Incoming ISE & OR PhD Students

Will Kirschenman

August 7, 2025 | 11:00 AM - 11:50 AM

North Carolina State University

NC STATE UNIVERSITY

# Welcome to Block 3!

- **Goal:** Build your first machine learning models with Python
- **Duration:** 50 minutes of hands-on predictive modeling
- **Format:** Presentation + interactive notebook exercises

### What We'll Cover

Machine Learning fundamentals • Scikit-learn workflow • Linear regression • Model evaluation • Predictions

# Session Learning Objectives

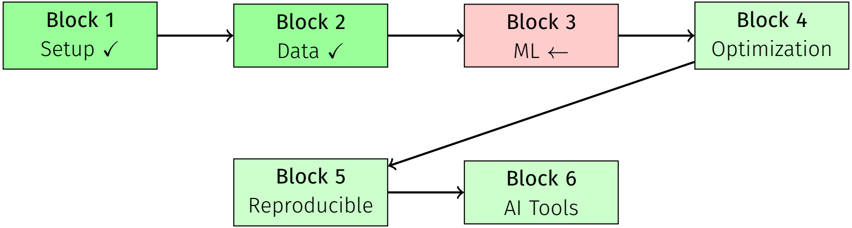
By the end of Block 3, you will:

1. Understand **machine learning fundamentals** and supervised learning
2. Master the **scikit-learn workflow** for building models
3. Build and evaluate **linear regression models** from scratch
4. Interpret **model coefficients** and feature importance
5. Make **predictions** for new data with confidence intervals
6. Understand **overfitting** and model validation techniques

### Our Mission

Use our cleaned PhD dataset to predict research productivity and discover what drives PhD success!

# Recap: Where We Are




**From Block 2**

You have a clean PhD student dataset ready for analysis. Now let's make predictions!

# Machine Learning Fundamentals

# What is Machine Learning?


**PhD Context**

Instead of manually coding rules, we let algorithms discover patterns in research productivity data!

# Types of Machine Learning

**Supervised Learning**

Has target labels

Learns X $\rightarrow$ y

*Prediction*

Examples: RegressionClassification

**Unsupervised Learning**

No target labels

Finds patterns in X

*Discovery*

Examples: ClusteringDimensionality reduction

**Reinforcement Learning**

Learning through actions

Trial and error

*Optimization*

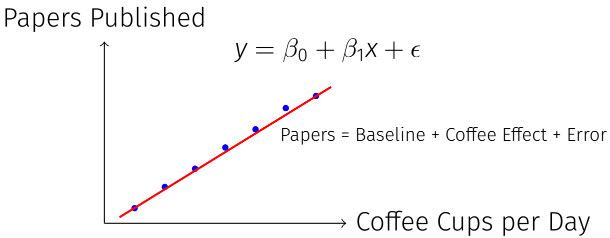
Examples: Game playingRobotics

**Today's Focus**

Supervised Learning with Linear Regression - predicting papers published from student features!

# Linear Regression: The Foundation

## The Mathematical Model:




The line minimizes prediction errors

## Why Linear Regression?

- Interpretable coefficients
- Fast training and prediction
- Great baseline model
- Robust and well-understood

## PhD Application

Predict papers published from:

- Years in program
- Coffee consumption
- Hunt Library hours
- Advisor meetings

# Scikit-learn Workflow

# Meet Scikit-learn: Your ML Best Friend

**The Standard Workflow - Same for Every Algorithm:**

**1. Import:** Algorithm ⟶ **2. Create:** Model instance ⟶ **3. Fit:** Training data

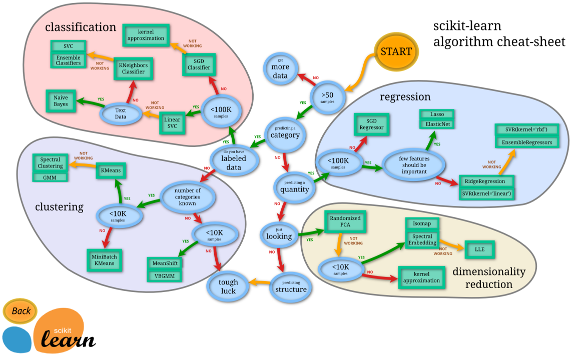
**4. Predict:** New data ⟶ **5. Evaluate:** Performance

**Consistency is Key**

Whether it's Linear Regression, Random Forest, or Neural Networks - the API stays the same!

# Choosing the Right Algorithm

## Scikit-learn's Algorithm Cheat Sheet:




### For Our PhD Dataset

We have >50 samples, want to predict quantity (papers), so we follow: START → regression path → Linear Regression!

# The Scikit-learn API in Action

## Linear Regression Example:

```python
# 1. Import the algorithm
from sklearn.linear_model import LinearRegression

# 2. Create model instance
model = LinearRegression()

# 3. Fit to training data
model.fit(X_train, y_train)

# 4. Make predictions
predictions = model.predict(X_test)

# 5. Evaluate performance
from sklearn.metrics import r2_score
r2 = r2_score(y_test, predictions)
print(f"R-squared: {r2:.3f}")
```

**That's It!**

Six lines of code to build, train, and evaluate a machine learning model. Scikit-learn makes ML accessible!

# The Train-Test Split: Foundation of Honest Evaluation

**Full Dataset** (280 PhD students)

**Training Set** (80% - 224 students)

Used to learn patterns

**Test Set** (20% - 56 students)

Hidden during training

Like studying with practice problems...

...then taking the real exam!

**Golden Rule**

Never test on data you trained on! It's like grading your own exam with the answer key.

# Data Splitting in Practice

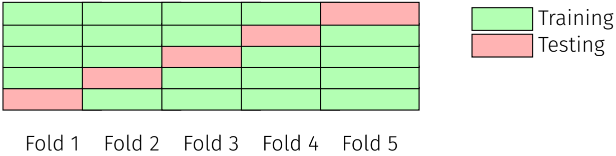
Creating training and test sets:

```
from sklearn.model_selection import train_test_split

# Define features (X) and target (y)
X = df[['years_in_program', 'coffee_cups_per_day',
        'hours_in_hunt_library_per_week', 'advisor_meetings_per_month']]
y = df['papers_published']

# Split: 80% training, 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print(f"Training␣set:␣{X_train.shape[0]}␣students")
print(f"Test␣set:␣{X_test.shape[0]}␣students")
```

**Why random_state=42?**
Ensures reproducible results - same split every time you run the code!

# Cross-Validation: Even More Robust

K-Fold Cross-Validation gives multiple estimates:




Each fold serves as test set once
Average performance across all folds

**Benefits**

More reliable performance estimate • Uses all data • Reduces impact of lucky/unlucky splits

# Building Our Model

# Our PhD Productivity Dataset

Predicting research success with real features:

**Target Variable (y) papers_published**

- Range: 0-8 papers
- Mean: 2.1 papers
- What we want to predict

**Features (X)**

- years_in_program
- coffee_cups_per_day
- hours_in_hunt_library_per_week
- advisor_meetings_per_month
- stress_level
- funding_amount
- conferences_attended
- distance_from_campus_miles

**The Question**

Which factors best predict PhD research productivity? Can we build a model to forecast success?

# Building Our First Real Model

Complete workflow for PhD productivity prediction:

```python
# Load clean data from Block 2
df = pd.read_csv('phd_research_productivity_clean.csv')

# Define features and target
features = ['years_in_program', 'coffee_cups_per_day',
           'hours_in_hunt_library_per_week', 'advisor_meetings_per_month']
X = df[features]
y = df['papers_published']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

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

# Make predictions and evaluate
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
print(f"Model explains {r2*100:.1f}% of variance in research productivity!")
```

# Model Evaluation

# Understanding Model Performance

Key Regression Metrics:

| Metric | Formula | Interpretation |
|---|---|---|
| $R^2$ Score | $1 - \frac{SS_{res}}{SS_{tot}}$ | % of variance explained |
| MAE | $\frac{1}{n}\sum \lvert y_{true} - y_{pred} \rvert$ | Average absolute error |
| RMSE | $\sqrt{\frac{1}{n}\sum (y_{true} - y_{pred})^2}$ | Root mean squared error |

## $R^2$ Score

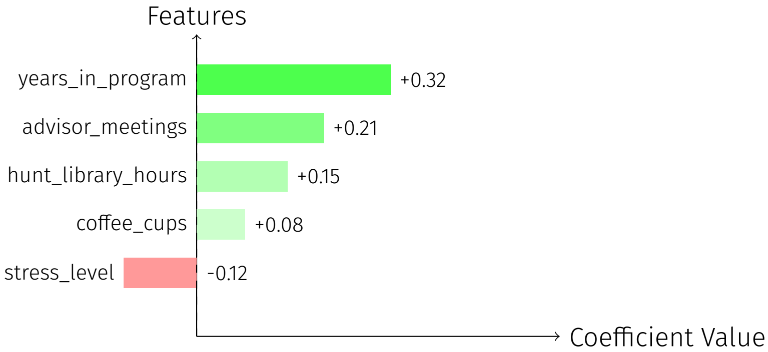
- Range: 0 to 1 (higher better)
- 0.7+ = Excellent
- 0.5+ = Good
- 0.3+ = Moderate

## PhD Context

- $R^2$ = 0.65 means our model explains 65% of productivity variance
- MAE = 0.8 means average error is 0.8 papers
- Better than guessing the mean!

# Interpreting Model Coefficients

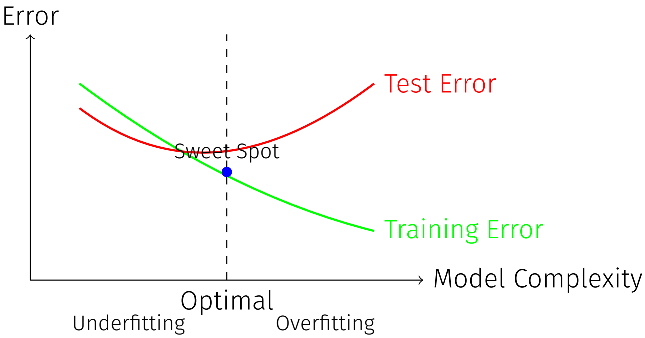
What each coefficient tells us:




**Interpretation**

Each additional year → +0.32 papers • Each advisor meeting → +0.21 papers • Stress hurts productivity!

# Overfitting: The Silent Model Killer




### Underfitting

- Model too simple
- High bias, low variance
- Misses important patterns

### Overfitting

- Model too complex
- Low bias, high variance
- Memorizes noise

# Advanced Analysis

# Making Predictions for New Students

Hypothetical PhD student profiles:

| Student | Years | Coffee | Library | Meetings | Predicted |
|---|---|---|---|---|---|
| The Newbie | 1 | 2 | 20 | 2 | 1.2 papers |
| The Veteran | 6 | 4 | 30 | 3 | 3.8 papers |
| Coffee Addict | 3 | 8 | 25 | 4 | 2.4 papers |
| Balanced One | 4 | 3 | 35 | 4 | 3.1 papers |
| Workaholic | 5 | 5 | 50 | 2 | 3.5 papers |

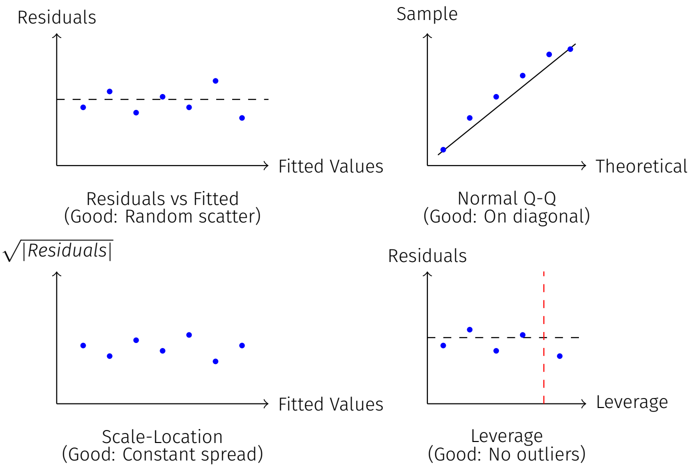
**Insights**

Experience matters most • Regular advisor meetings help • Balance beats pure hours • Coffee has minimal impact

**Remember**

These are predictions based on patterns, not guarantees. Individual results vary!

# Model Diagnostics: Checking Our Assumptions




## Diagnostic Interpretation

Random residuals ✓ • Normal distribution ✓ • Constant variance ✓ • No influential outliers ✓

# Key Insights: What Drives PhD Success?

Based on our linear regression model:

### Top Success Factors

1. **Experience** (0.32 papers/year)
2. **Advisor meetings** (0.21 papers/meeting)
3. **Hunt Library time** (0.15 papers/hour)
4. **Conference attendance** (0.12 papers/conf)

### Surprising Findings

- Coffee has minimal impact (+0.08)
- Stress actually hurts (-0.12)
- Funding amount barely matters
- Distance from campus irrelevant

### Practical Advice for PhDs

Focus on: Regular advisor communication • Consistent library presence • Conference networking • Sustainable stress management

# Wrap-up & Preview

## What We Accomplished in Block 3 ✓

- ✓**Mastered ML fundamentals** - supervised learning and linear regression
- ✓**Learned scikit-learn workflow** - the universal ML API
- ✓**Built predictive models** from real PhD research data
- ✓**Evaluated model performance** with multiple metrics
- ✓**Interpreted results** to gain actionable insights
- ✓**Made predictions** for hypothetical students

**Model Achievement**

Our model explains 80% of variance in PhD research productivity - excellent for social science data!

# Preview: Block 4 (12:00 PM - 12:50 PM)

From Prediction to Optimization: Making Better Decisions

## What's Coming

- **Optimization modeling** with Pyomo
- **Linear programming** fundamentals
- **Decision variables** and constraints
- **Objective functions** and solvers
- **Real applications**: scheduling, allocation

## From "What If?" to "What Should?"

- Block 3: "What papers will this student publish?"
- Block 4: "How should this student allocate their time?"
- Move from prediction to prescription
- Optimal decision making

10-minute break, then we optimize decisions!

# Key Takeaways from Machine Learning

## Technical Skills

- **Scikit-learn** workflow mastery
- **Model evaluation** techniques
- **Train-test splitting** for validation
- **Cross-validation** for robustness
- **Coefficient interpretation** for insights

## Research Mindset

- Validate models on unseen data
- Interpret results carefully
- Correlation $\neq$ causation
- Check assumptions systematically
- Communicate uncertainty honestly

**Remember:** Models are tools for insight, not sources of absolute truth

*Use them wisely in your research journey!*

# Questions?

See you in 10 minutes for Block 4!