Statistics, Mathematical Models, and Algorithms

Module 3

Mohammed Al-Muqsit, Kimiwa Sadat, Ibrahim Rahat, Anour Ibrahim, Avirup Ray, and Shafin Rehman

Why Math & Statistics Matter in Data Science

Understanding Relationships

Discover how variables influence each other through correlation and causation analysis.

Predicting Outcomes

Build models that forecast future events using historical patterns and statistical inference.

Informed Decisions

Quantify uncertainty to support data-driven decision-making in business and science.

Real-world applications: Stock price forecasting, customer segmentation, disease diagnosis, and risk assessment all depend on mathematical modeling and statistical analysis.

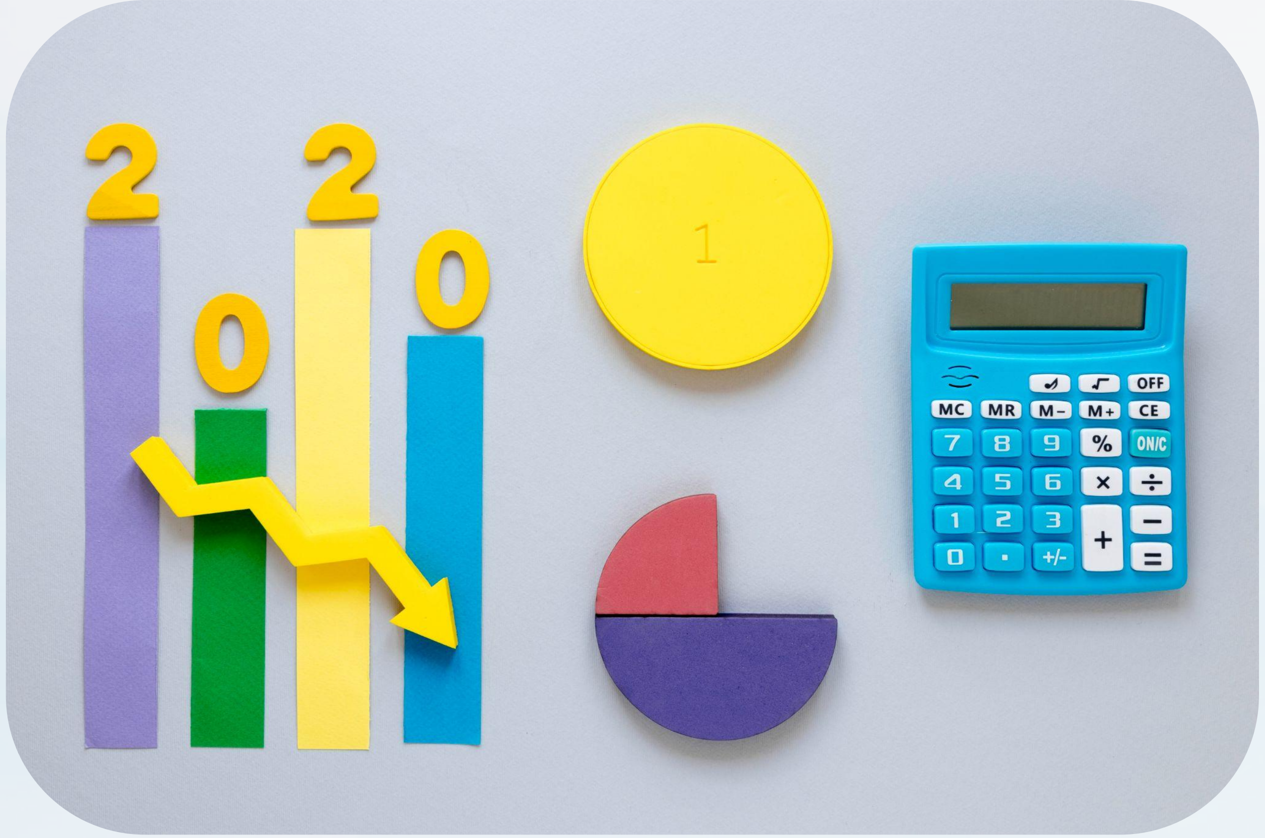
Mathematical and Statistical Models

Mathematical Models

- Equations that show systems, patterns, or relationships in data.
- Uses deterministic functions to show input changes to outputs

Statistical Models

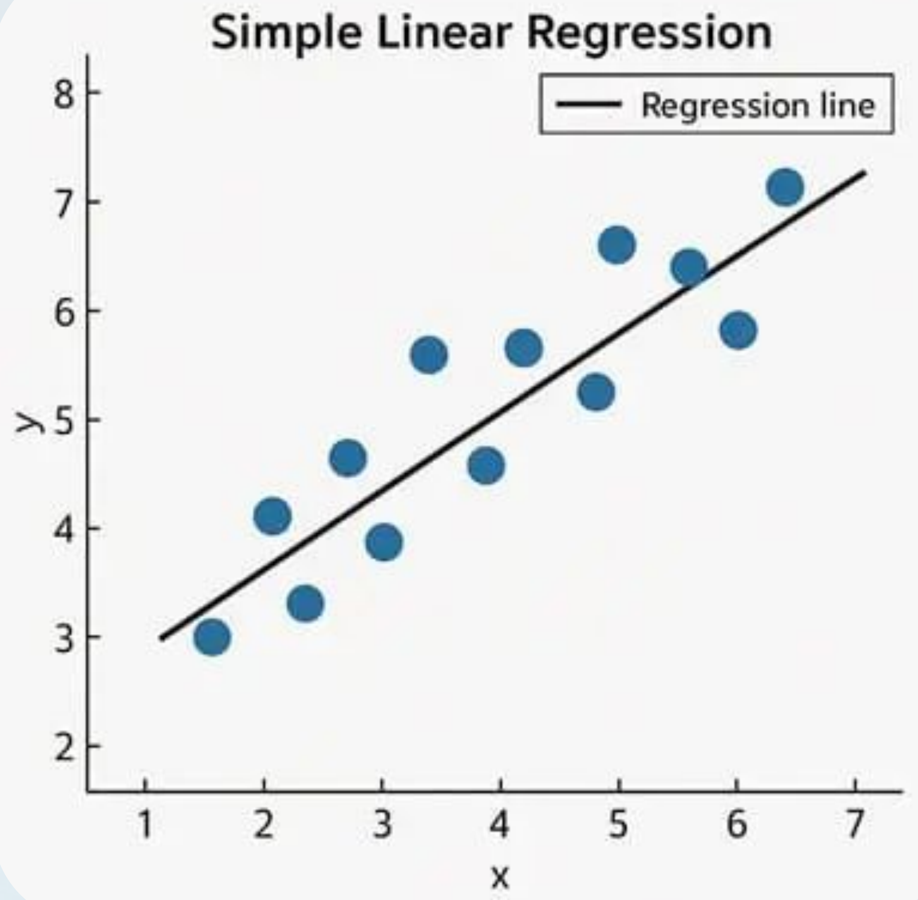
- Probabilistic frameworks that infer relationships and quantify uncertainty.
- Account for randomness and variability in data.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.9718	0.9971	0.9718	0.9780	0.9712	0.9573	0.9609	0.9190
knn	K Neighbors Classifier	0.9718	0.9830	0.9718	0.9780	0.9712	0.9573	0.9609	0.0370
qda	Quadratic Discriminant Analysis	0.9718	0.9974	0.9718	0.9780	0.9712	0.9573	0.9609	0.0300
lda	Linear Discriminant Analysis	0.9718	1.0000	0.9718	0.9780	0.9712	0.9573	0.9609	0.0330
lightgbm	Light Gradient Boosting Machine	0.9536	0.9935	0.9536	0.9634	0.9528	0.9298	0.9356	0.3150
nb	Naive Bayes	0.9445	0.9868	0.9445	0.9525	0.9438	0.9161	0.9207	0.0300
et	Extra Trees Classifier	0.9445	0.9935	0.9445	0.9586	0.9426	0.9161	0.9246	0.0880
catboost	CatBoost Classifier	0.9445	0.9922	0.9445	0.9586	0.9426	0.9161	0.9246	0.1220
gbc	Gradient Boosting Classifier	0.9355	0.9792	0.9355	0.9416	0.9325	0.9023	0.9083	0.1360
xgboost	Extreme Gradient Boosting	0.9355	0.9868	0.9355	0.9440	0.9343	0.9023	0.9077	0.0710
dt	Decision Tree Classifier	0.9264	0.9429	0.9264	0.9502	0.9201	0.8886	0.9040	0.0270
rf	Random Forest Classifier	0.9264	0.9909	0.9264	0.9343	0.9232	0.8886	0.8956	0.0900
ada	Ada Boost Classifier	0.9155	0.9947	0.9155	0.9401	0.9097	0.8720	0.8873	0.0580
ridge	Ridge Classifier	0.8227	0.0000	0.8227	0.8437	0.8186	0.7320	0.7454	0.0220
svm	SVM - Linear Kernel	0.7618	0.0000	0.7618	0.6655	0.6888	0.6333	0.7048	0.0300
dummy	Dummy Classifier	0.2864	0.5000	0.2864	0.0822	0.1277	0.0000	0.0000	0.0490

Linear Regression

Linear regression models the relationship between input variables and continuous outputs using the equation: **$Y = a + bX + \varepsilon$**




```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([1, 2, 3, 4, 5, 6, 7]).reshape(-1, 1)
y = np.array([3, 4, 4.5, 5.5, 6, 7, 7.5])

# Fit model
model = LinearRegression().fit(X, y)
y_pred = model.predict(X)
```

Use Cases

Price prediction, temperature forecasting, salary estimation, and trend analysis across continuous domains.

Strengths

Simple to implement and interpret. Fast training. Provides clear coefficient values showing variable impact.

Limitations

Assumes linear relationships. Sensitive to outliers. Poor performance on non-linear patterns and complex data.

Logistic Regression

Logistic regression predicts whether something belongs to one of two groups (like yes/no or 0/1). It is used to predict yes/no outcomes by giving probabilities between 0 and 1.

📧 Spam Email Detection

Classify emails as spam or legitimate

🩺 Medical Diagnosis

Predict disease presence with associated confidence scores for clinical decision-making.

Customer Churn

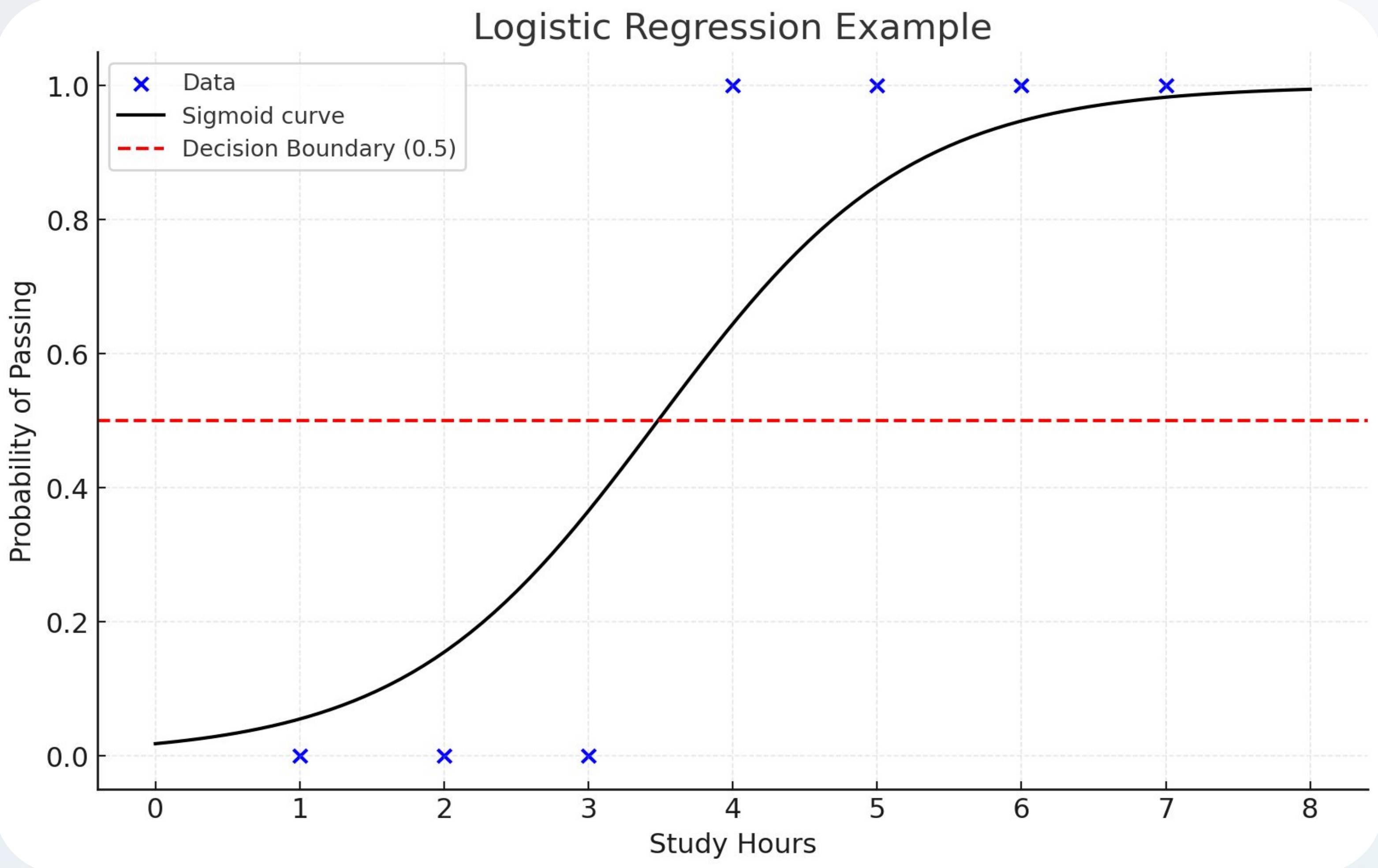
Identify customers likely to leave using historical behavior and engagement metrics.

Strengths: Interpretable probabilities, computationally efficient.

Limitations: Struggles with complex non-linear boundaries, sensitive to multicollinearity between features.

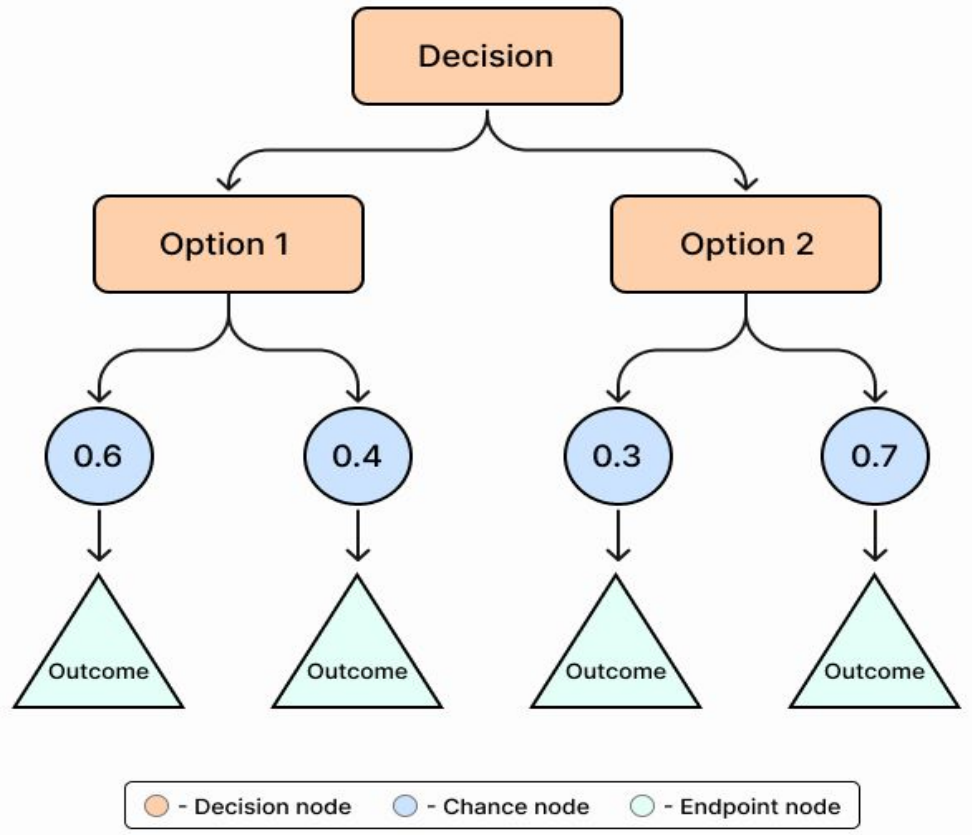
Logistic Regression: Code and Graph

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

# Sample data (X = study hours, y = pass/fail)
X = np.array([1, 2, 3, 4, 5, 6, 7]).reshape(-1, 1)
y = np.array([0, 0, 0, 1, 1, 1, 1])

# Fit logistic regression model
model = LogisticRegression()
model.fit(X, y)

# Predict probabilities
X_test = np.linspace(0, 8, 100).reshape(-1, 1)
y_prob = model.predict_proba(X_test)[:, 1]
```



Decision Tree

Decision trees split data into smaller groups based on feature values, forming a tree-like structure. Each branch represents a decision rule, and each leaf gives a prediction.



Use Cases

Classification, Regression, Feature selection and quick model interpretation

Strengths

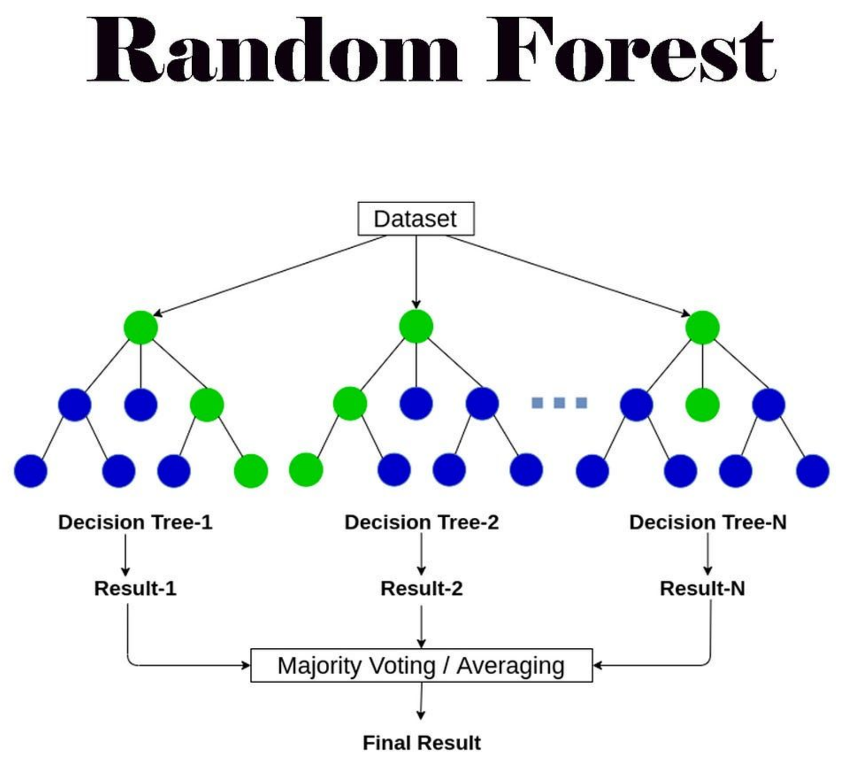
Easy to understand and visualize, works with both numerical and categorical data, captures non-linear relationships

Limitations

Prone to overfitting on small datasets, small changes in data can change the tree structure, less accurate than ensemble models like Random Forests

Random Forest

Random forests combine multiple decision trees through bagging and feature randomness, creating robust models that handle complex, non-linear relationships in data.



Use Cases

Builds many trees on random data samples and feature subsets. Final prediction: majority vote (classification) or average (regression).

Strengths

Handles non-linear patterns, high accuracy, reduces overfitting, naturally ranks feature importance.

Limitations

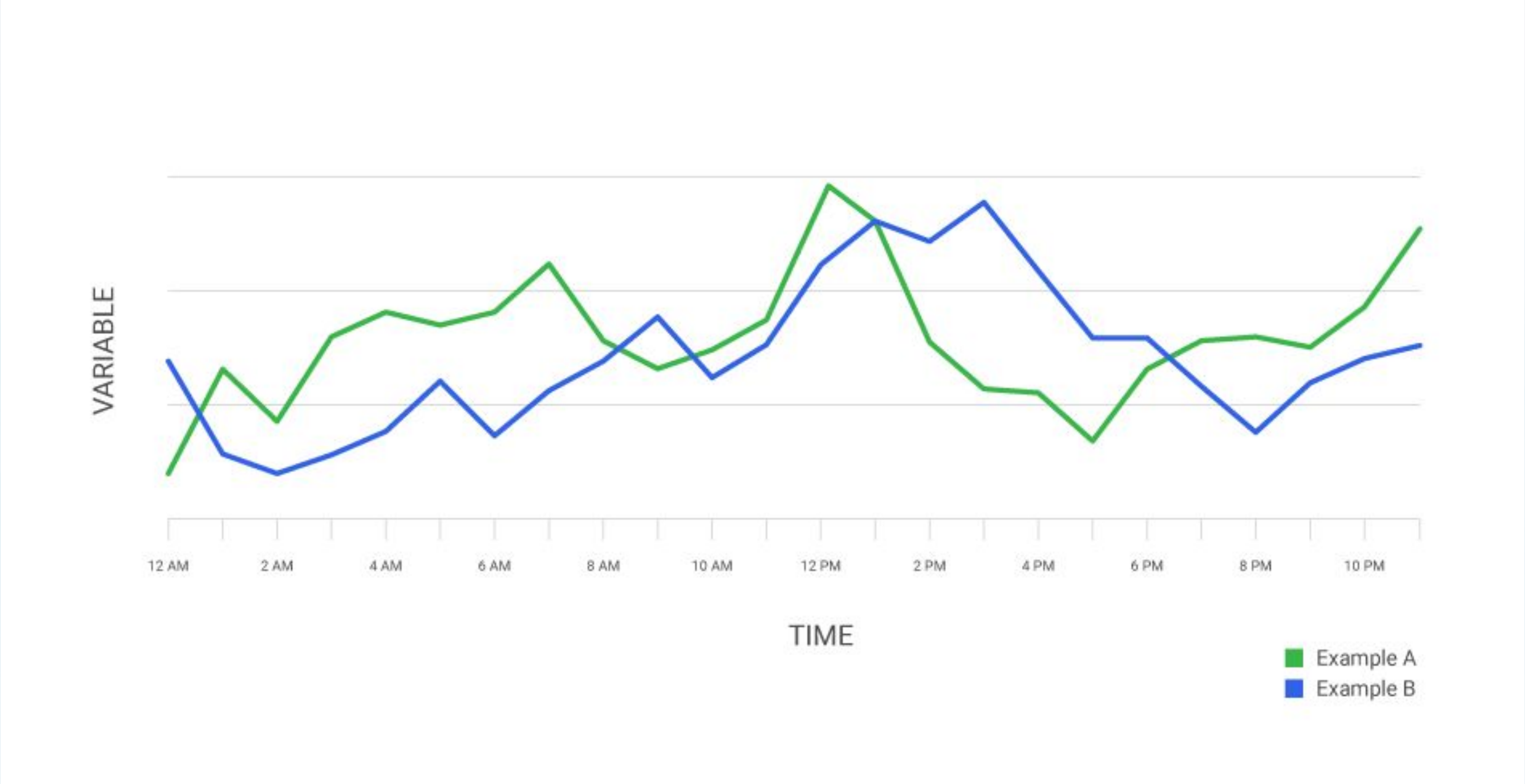
Slower training and prediction. Less interpretable than simpler models. Requires more computational resources.

Model Comparison: Which Algorithm to Choose?

Model	Input Type	Output Type	Optimization	Key Strength
Linear Regression	Continuous	Continuous	Mean Squared Error	Simplicity & speed
Logistic Regression	Mixed	Binary	Maximum Likelihood	Interpretability
Random Forest	Mixed	Both types	Bagging	Robustness & accuracy

Time Series: Data Across Time

Time series data captures observations recorded at regular intervals—hourly stock prices, daily weather, monthly sales.



Strengths

Excellent for forecasting sequential patterns. Captures temporal dependencies missed by cross-sectional models.

Limitations

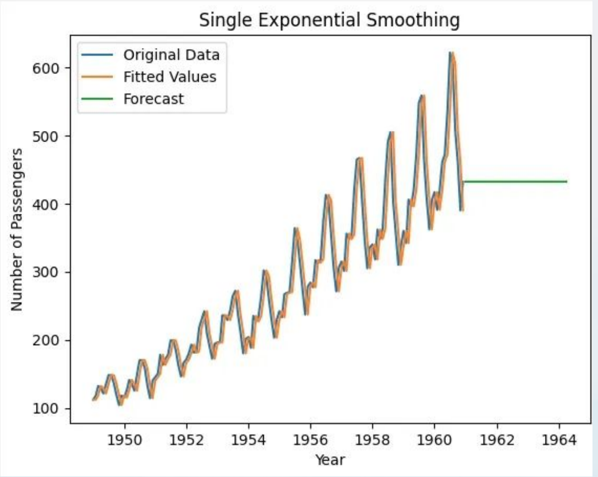
Requires stationarity and complete data. Sensitive to noise, anomalies, and structural breaks in underlying patterns.

Time Series Forecasting Techniques

Different methods help smooth data changes and capture time patterns, each working best for certain types of data and forecast lengths.

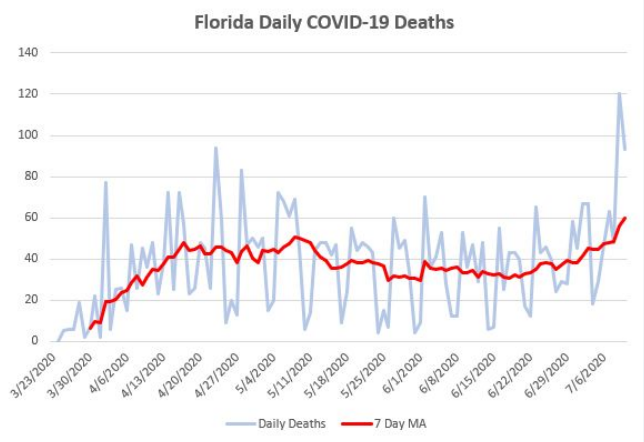
Exponential Smoothing

Weights recent data more heavily than distant observations, adapting quickly to level changes.



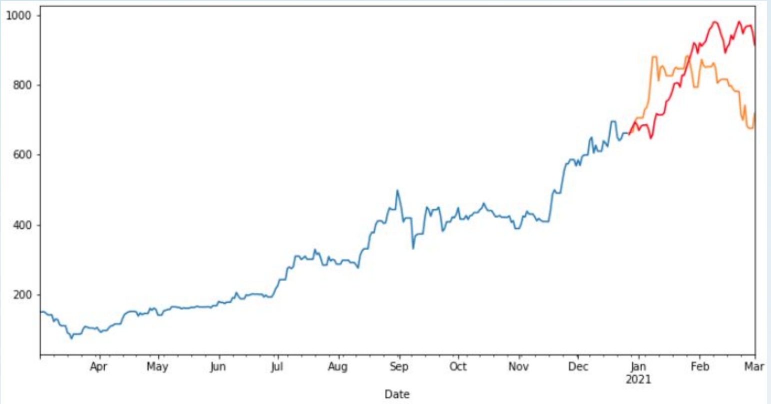
Moving Average

Smooths short-term fluctuations by averaging recent observations, reducing noise in volatile series.



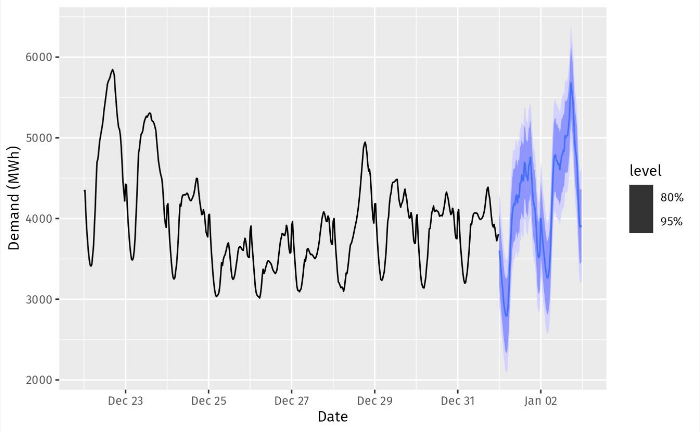
ARIMA

Models autoregressive relationships, integration, and moving average components for complex temporal patterns.



Prophet

Facebook's framework decomposes trend an seasonality explicitly, handling missing data and outliers gracefully.



Understanding Algorithms

Supervised Learning

Algorithms learn from labeled data to make predictions

- Regression for continuous outcomes
- Classification for categorical outcomes

Unsupervised Learning

Discover hidden patterns in unlabeled data

- Clustering to group similar items
- Dimensionality reduction for simplification

Reinforcement Learning

Optimize actions through trial and feedback

- Learn from rewards and penalties
- Ideal for dynamic environments

Statistical Testing Overview

Statistical tests validate hypotheses and ensure findings aren't due to chance. Choosing the right test depends on your data type and research question.

T-Test

Data Type: Continuous

Purpose: Compare means between groups

Pros: Simple and widely accepted standard

Cons: Assumes normal distribution

Chi-Square

Data Type: Categorical

Purpose: Test independence between variables

Pros: Distribution-free approach

Cons: Unreliable with small samples

Mann-Whitney U

Data Type: Non-normal distributions

Purpose: Compare medians between groups

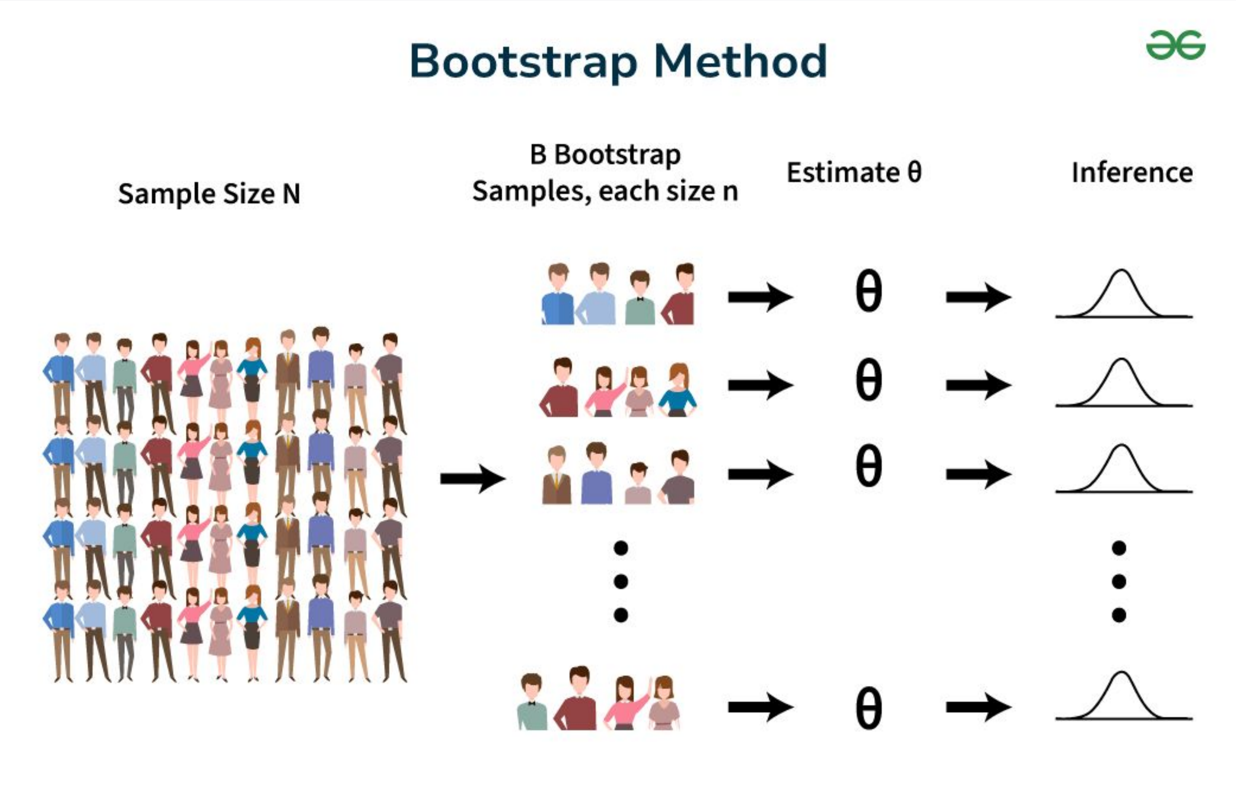
Pros: Robust to outliers and skewness

Cons: Lower power when data is normal

Advanced Techniques

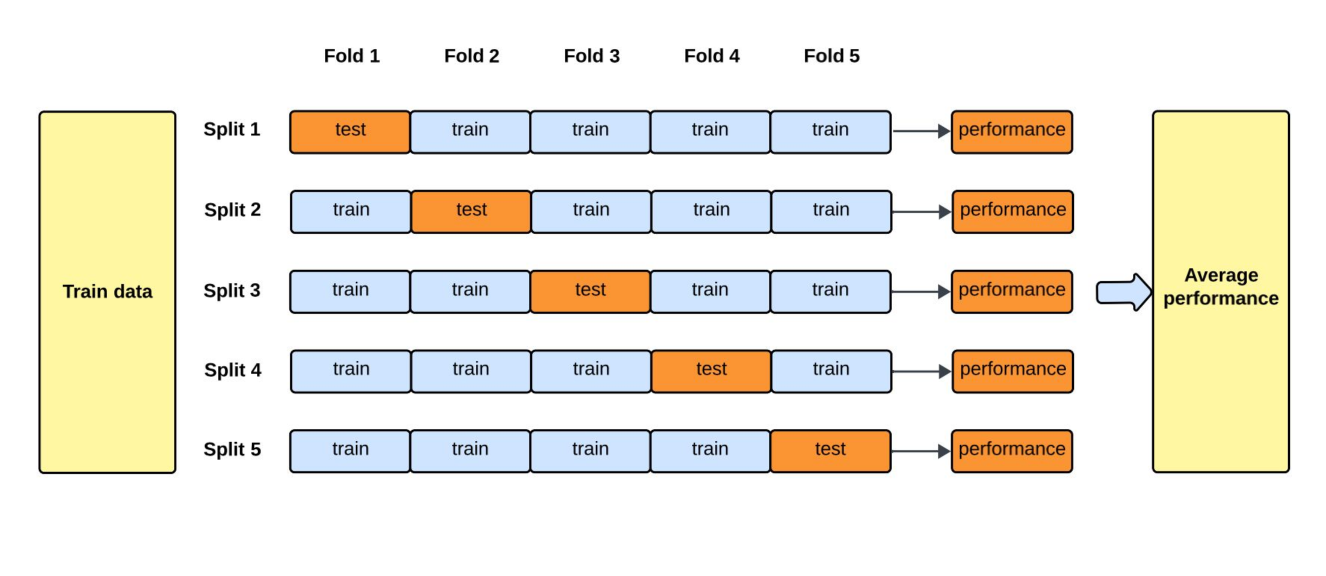
Bootstrapping

Resampling technique that estimates confidence intervals and standard errors without assuming distribution shape. Implemented using SciPy for robust statistical inference.



Cross-Validation

Evaluate model generalization by testing on multiple data subsets. K-fold cross-validation divides data into K parts, training on K-1 and testing on the remaining fold iteratively.



Visualization & Tools

Effective visualization transforms complex data patterns into intuitive insights. Python visualization ecosystem offers tools for every level of complexity.

Matplotlib

The foundation of Python visualization. Offers maximum flexibility and fine-grained control over every plot element. Perfect for custom, publication-ready graphics.

Seaborn

High-level interface built on Matplotlib. Specializes in statistical visualization with beautiful default styles. Ideal for correlation heatmaps and regression plots.

The Analytical Backbone of Data Science

Statistics Describe and infer from data	**Math Models** Formalize and predict outcomes
Algorithms Scale and automate learning	**Time Series** Capture change over time

Together, these pillars form a comprehensive toolkit for transforming data into knowledge and driving informed decision-making.

Key Libraries & Tools

Pandas • NumPy • SciPy • Statsmodels • Scikit-learn • Seaborn • Matplotlib • Prophet

Any Questions?

Thank You For Listening