Machine Learning Dashboard Workshop

Association of Data Science

Goals

- Load coffee sales data from GitHub
- Display dataset overview
- Visualize feature distribution
- Train a random forest model to predict features
- Show model performance
- Provide real-time prediction
- Create a Streamlit app for interactive predictions
- Deploy app using ngrok

Machine Learning

- Machine learning (ML) is a branch of Artificial Intelligence that enables computers to learn patterns from data and make predictions or decisions without it being explicitly programmed

Types of ML

- **Supervised Learning**: Learn from labeled data (e.g., predicting coffee type from purchase info)
- **Unsupervised Learning**: Find hidden structure in unlabeled data (e.g., clustering customer groups)
- **Reinforcement Learning**: Learn by trial and error with feedback (e.g., training a robot)

Classification Model

Definition

- Machine learning models that predict discrete categories rather than continuous values

Examples

- Will a customer buy coffee? → Yes / No
- What type of coffee? → Latte / Espresso / Cappuccino

Key Idea

- Learn from labeled data to map features → classes

Types of classification problems:

- Binary classification → 2 outcomes (e.g., yes/no)
- Multiclass classification → >2 outcomes (e.g., type of coffee)

Random Forest Classifier

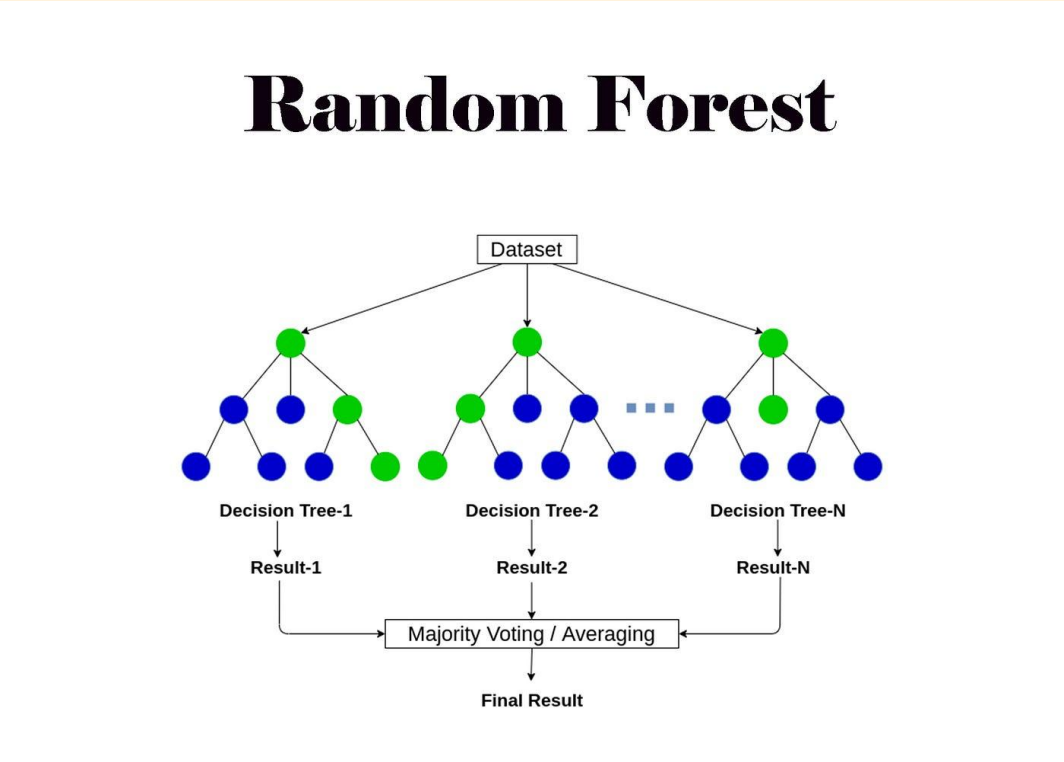
What it is

- An ensemble model that combines many decision trees
- Each tree votes → majority vote = prediction
- Supervised Learning

Benefits

- Handles both categorical and numerical data
- Reduces overfitting compared to a single decision trree
- Minimal tuning needed

Random Forest Classifier



Evaluating Classification Models

Accuracy

- % of correct predictions
- Easy to understand, but can be misleading with imbalanced data

Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Evaluating Classification Models

Key Metrics

- **Precision** = TP / (TP + FP) → “of predicted positives, how many were correct?”
- **Recall** = TP / (TP + FN) → “of actual positives, how many did we catch?”
- **F1 Score** → 2 * (Precision x Recall) / (Precision + Recall) → balance between precision and recall

Importance

- Accuracy is not enough, especially when data set is imbalance
- F1 Score gives a single balanced measure

What is Streamlit?

- Streamlit is an open-source Python framework for building interactive web apps/dashboards
- Designed for data scientists
- No front-end or web dev skills required

Key Features

- Simple, Pythonic API (`st.write()`, `st.plotly_chart()`)
- Turns scripts into shareable apps instantly
- Supports interactive widgets (slides, buttons, dropdowns)
- Integrates with popular libraries (Pandas, Matplotlib, Plotly, Scikit-learn, etc.)

Why use ngrok with Streamlit?

The Challenge

- Streamlit apps run locally (default http://localhost:8501)
- In Google Colab or cloud notebooks, localhost isn’t publicly accessible

The Solution

- A tunneling service that creates a secure public URL
- Redirects traffic to your local Streamlit app
- Lets others (or you) access the app via browser

The Workflow

- Run Streamlit app
- Start ngrok tunnel
- Use provided public link to access the app

Using ngrok

In order to use ngrok, you must create an account and get a token

1. Go to https://ngrok.com/ and create an account
2. Once you have your account, go to the menu on the left side of the screen
3. Find the "Getting Started" section, and under that click "Your Authtoken"
4. Copy the token and paste it below where it says "PASTE_YOUR_AUTHTOKEN"

Token & API Key Safety

Why It Matters

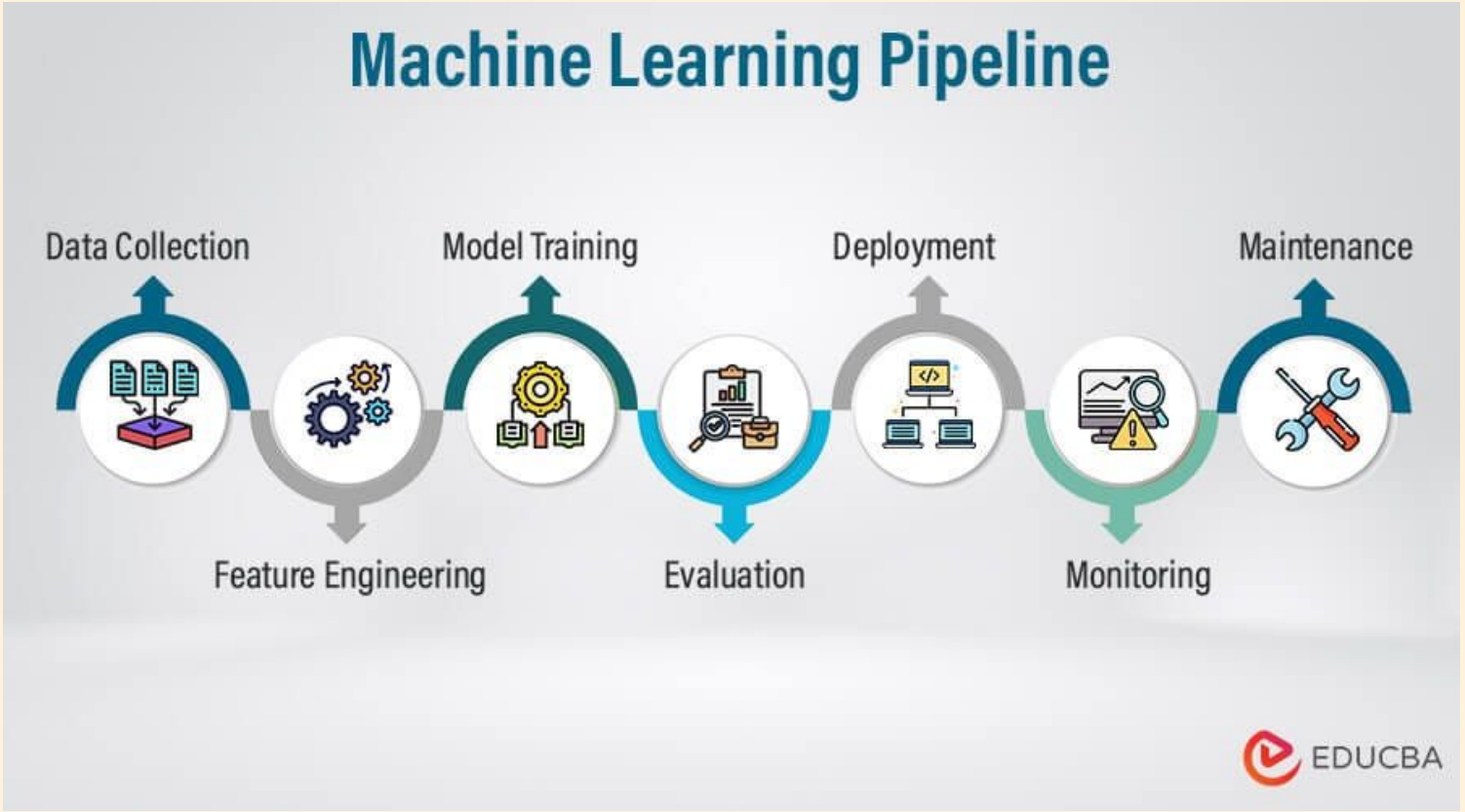
- API Keys/Tokens = passwords for your services
- If leaked, others can use your account → costs and data breaches

Best Practices

- Never hardcode keys in code
- Store your keys in environment variables or secret managers
- Rotate keys regularly
- Restrict permissions

Note we will hardcode our keys in this workshop since we are using colab notebooks, but it is a bad practice*

Machine Learning Pipeline



Building off of the workshop!

- Find your own data set (kaggle.com is a great resource)
- Determine what you are trying to answer/solve
- Create visualizations to explore the data and answer the question
- Add interactivity
- If using an IDE, you can run Streamlit locally and deploy it publicly using Modal

https://github.com/hannahbanjo/AssociationOfDataScience

Dues & T-Shirts