Overview

1. Course Plan ✓

2. AWS SageMaker
- ○ Introduction

3. AWS services:
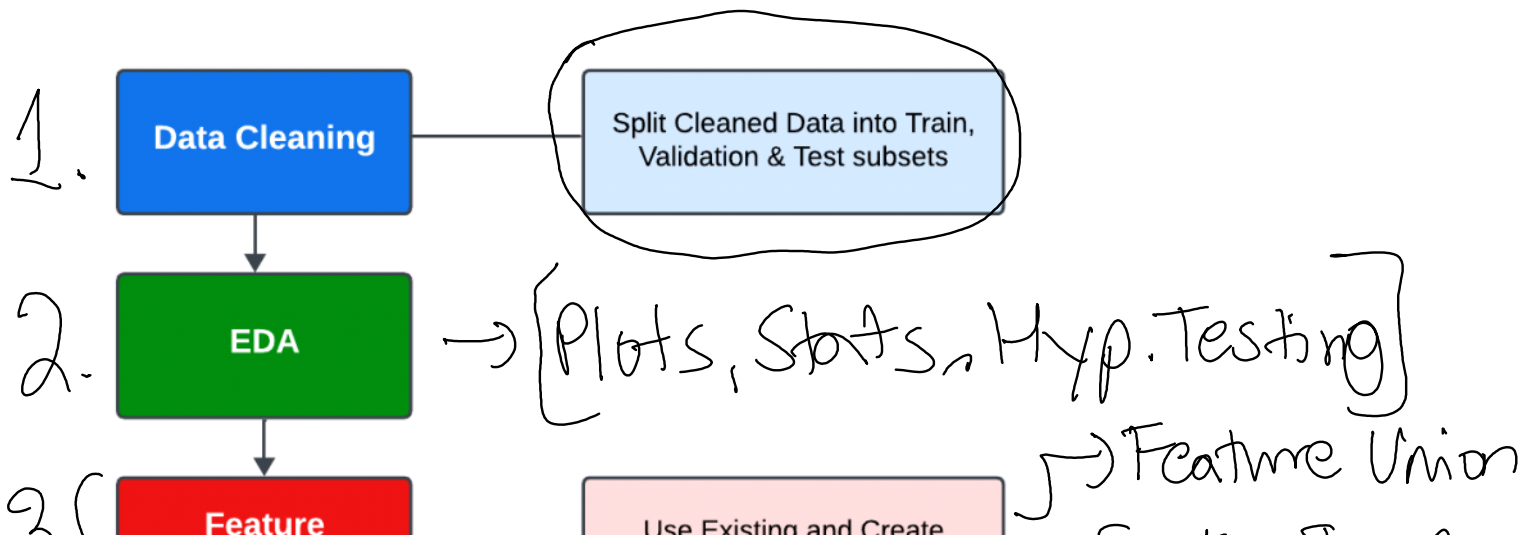- ○ IAM
- ○ EC2
- ○ S3

4. SDK (Software Development Kit)
- ○ Definition
- ○ Advantages
- ○ Boto3

5. AWS Free Tier Plans
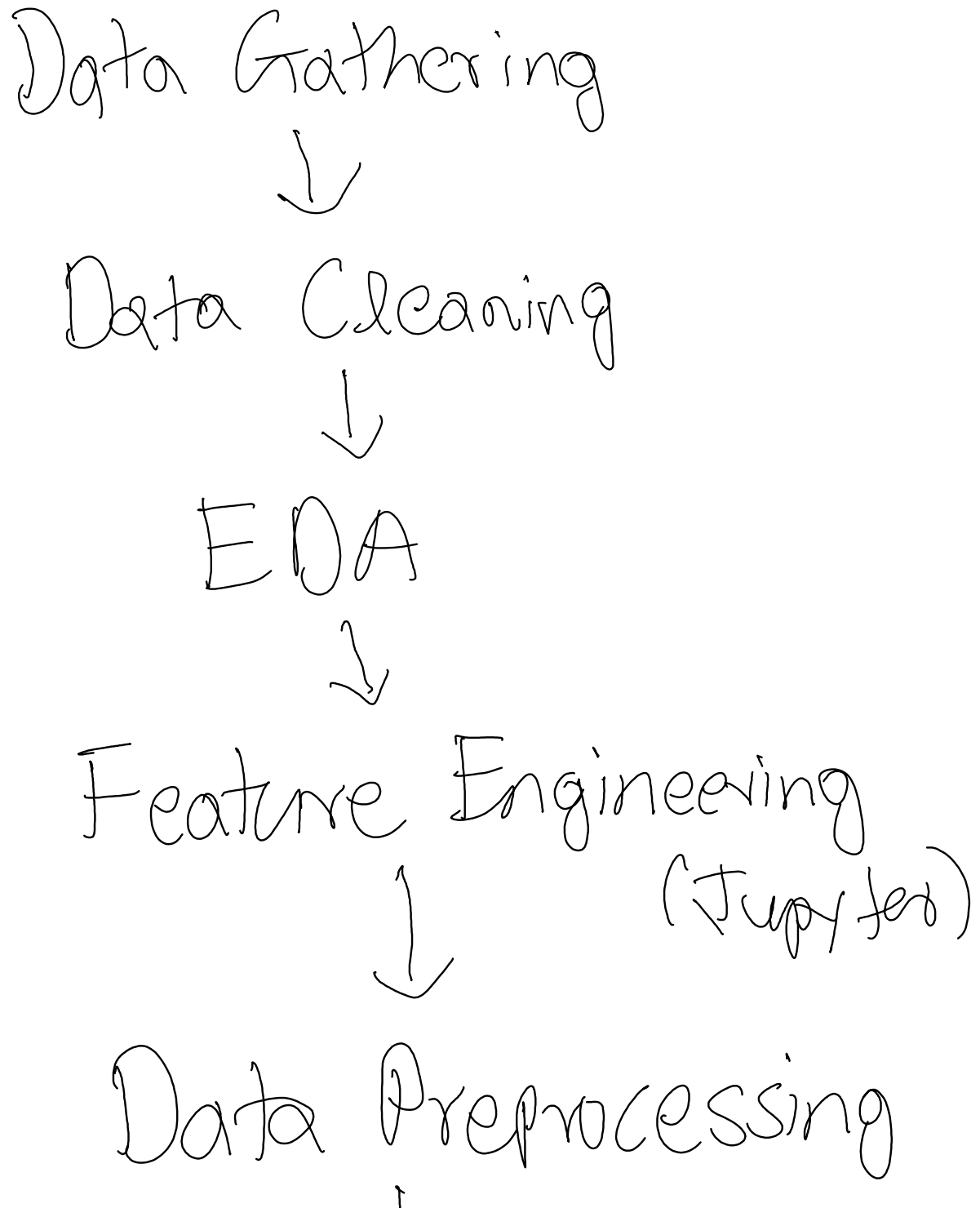
6. Create AWS account

7. Project Workflow

1.

Data Cleaning — Split Cleaned Data into Train, Validation & Test subsets

2. **EDA** → [Plots, Stats, Hyp. Testing]

→ Feature Union

3. **Feature** — Use Existing and Create

3. {

Feature Engineering → Use Existing and Create Custom Classes

↱ Feature Union
→ Fraction Transform
↳ Custom Class

Data Preprocessing → Pipelines, Feature Union and Column Transformer

4. **SageMaker NB Instance** → **5** → **Web Application using Streamlit** → **6** → **Deploy Web Application using Streamlit Cloud**

7. **Clean-up Operations**

Preprocess all 3 subsets

Upload Data to S3 Bucket

Set-up ML model

Set-up Hyperparameter Tuner

Train and Tune model using EC2

Save Best Model in S3 Bucket

Save Best Model in
S3 Bucket

8. Create SageMaker Notebook Instance

Data Gathering

↓

Data Cleaning

↓

EDA

↓

Feature Engineering
(Jupyter)

↓

Data Preprocessing

↑

Data Preprocessing

↓

Model Training

↓

Model Tuning

↓

Deployment

↓

Monitor

Great Model + Bad Data → Poor Results

Basic Model + Great Data → Amazing Results

# AWS SageMaker

1. Definition

Amazon SageMaker is a fully managed machine learning (ML) service. With SageMaker, data scientists and developers can quickly and confidently build, train, and deploy ML models into a production-ready hosted environment. It provides a UI experience for running ML workflows that makes SageMaker ML tools available across multiple integrated development environments (IDEs).
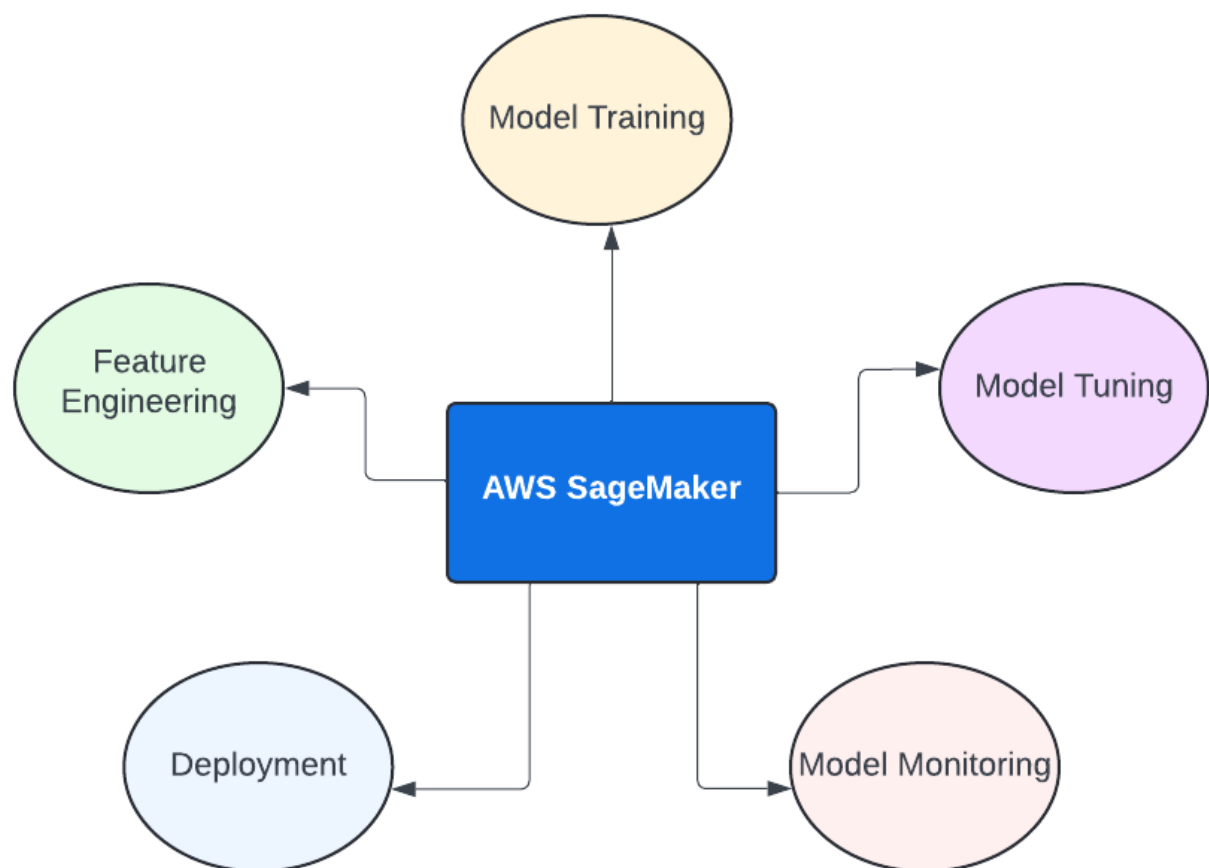
2. Advantages

- It's a fully-managed service
- Provides wide range of libraries for ML workflows
- Provides wide range of algorithms for various problems
  - *Regression*
  - *Classification*
  - *Object Detection*
  - *Forecasting*
  - *Clustering*
  - *Dimensionality Reduction*
  - *Recommendation Systems*
- One-click deployment
- Integration with other AWS services
- Provides notebooks and IDE for development
  - *Classic Jupyter*
  - *Jupyter Lab*
  - *SageMaker Studio (IDE)*
- Detailed guides and tutorials
- Huge community and online support

Apart from SageMaker, we'll be using the following services from AWS to implement our End-to-End ML Project:
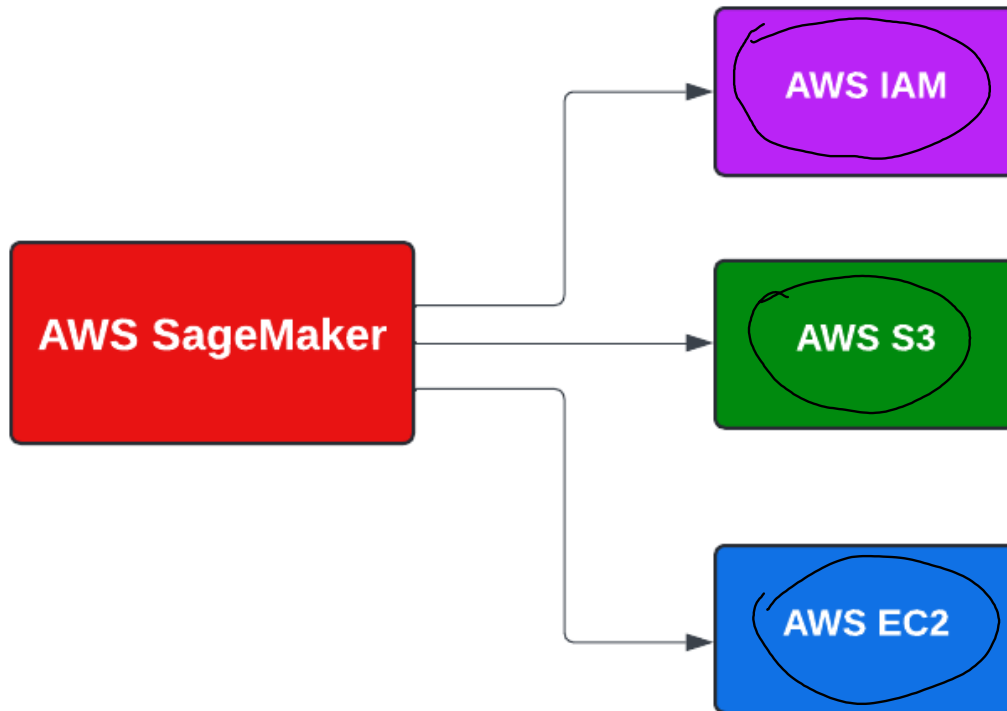
1. **IAM** ~ Identity & Access Management

2. **EC2**

3. **S3**

IAM → Permissions, Access

↳ "Root User"
  ↳ [JAM Group]
  [JAM User]
  [JAM Role] → Project

EC2 - Elastic Cloud Compute
  ↳ Train   ["ml.m4.xlarge"]
  ↳ Tune

S3 - Simple Storage Service
  ↳ Bucket
  ↳ Datasets

↳ Best Model (after training)

SDK

1. Definition

What is an SDK?

A software development kit (SDK) is a set of platform-specific building tools for developers. You require components like debuggers, compilers, and libraries to create code that runs on a specific platform, operating system, or programming language. SDKs put everything you need to develop and run software in one place. Additionally, they contain resources like documentation, tutorials, and guides as well as APIs and frameworks for faster application development.

Code

Documentation

API

APIs

Software
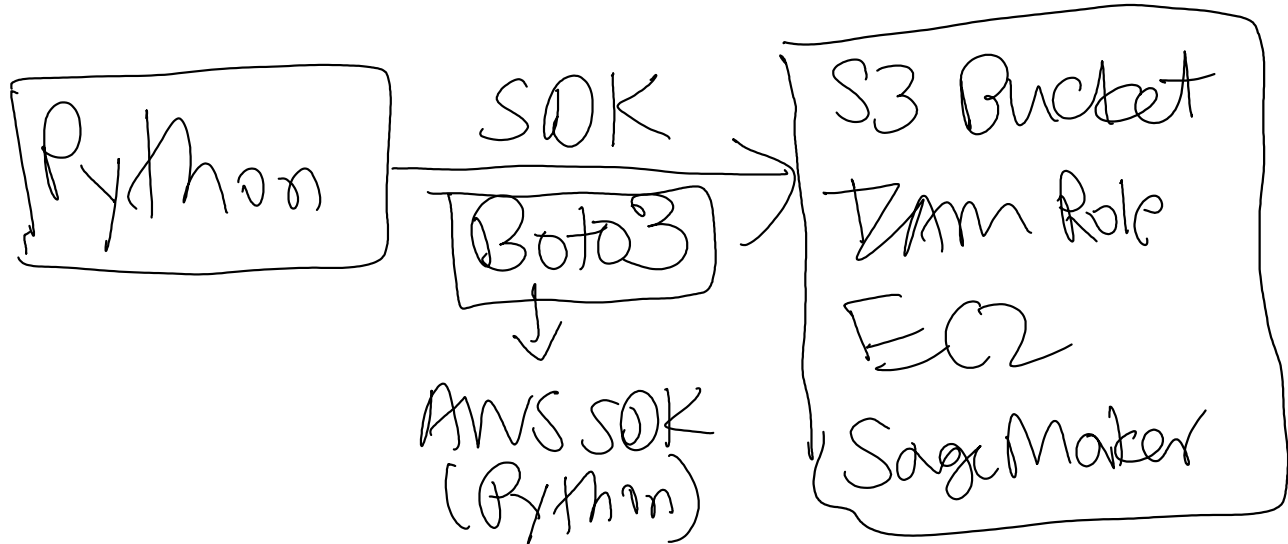# Development Kit

Libraries

Guides

2. Advantages
- ○ Efficient Development
- ○ Reduced time complexity
- ○ Seamless Integration

○ Reduced Costs

3. Why did we study about SDK?
  ○ We'll be using Python to train models
  ○ Need a way for Python to interact with AWS services

AWS

Python — SDK → [S3 Bucket
                   IAM Role
                   EC2
                   SageMaker]

Boto3
↓
AWS SDK
(Python)

What SDKs does AWS provide?

AWS provides SDKs for many popular technologies and programming languages. They make it easier for you to call AWS services from within your applications in that language or technology. Additionally, AWS also offers SDKs for AWS SaaS offerings so you can use them more efficiently in your code. Some examples include:

- AWS SDK for .NET provides simplified AWS Services through a set of libraries that are consistent and familiar to .NET developers.
- AWS SDK for Python integrates Python applications, libraries, or scripts with AWS services.
- AWS SDK for Ruby eliminates programming intricacies by providing Ruby classes for many AWS services.
- AWS SDK for Rust simplifies the use of AWS services by providing a set of consistent and familiar libraries to Rust developers.
- AWS WorkDocs SDK removes the complexity of building file collaboration and management capabilities into your solutions and applications, providing full administrator and user-level access to Amazon WorkDocs site resources.
- Amazon Chime SDK allows developers to add real-time voice, video, and messaging powered by machine learning into their applications.