## Overview

This 3-hour session introduces students to the concept of derived features and transformations in data analysis and machine learning using Python. Students will learn how to create new features from existing data, apply various transformations, and understand when and why these techniques are useful.

## Prerequisites

* Basic Python syntax knowledge
* Familiarity with variables, functions, and loops
* Basic understanding of NumPy and Pandas libraries

## Learning Objectives

By the end of this session, students will be able to:

1. Understand what derived features are and why they're important
2. Create basic numerical transformations (log, square root, etc.)
3. Implement binary and categorical feature encoding
4. Apply normalization and standardization techniques
5. Create interaction features and polynomial features
6. Apply these concepts to simple datasets

## Materials Needed

* Python 3.x installed
* Jupyter Notebook or similar IDE
* Required libraries: NumPy, Pandas, Matplotlib, Scikit-learn

## Schedule

### Section 1 (90 minutes)

#### Topic 1: Introduction to Derived Features (20 minutes)

* Definition and importance of feature engineering
* Types of derived features
* When and why to create derived features
* Real-world examples and applications

#### Topic 2: Basic Numerical Transformations (25 minutes)

* Understanding data distributions
* Log transformations
* Square root and power transformations
* Binning continuous variables
* Handling skewed data

#### Topic 3: Categorical Feature Encoding (20 minutes)

* One-hot encoding
* Label encoding
* Binary encoding
* Target encoding
* Handling high-cardinality features

#### Breakout Room Activity 1 (25 minutes)

* Exploratory data analysis and transformation practice
* Students will analyze a dataset, identify skewed features, and apply appropriate transformations
* Implement categorical encoding on different feature types

### Section 2 (90 minutes)

#### Topic 4: Normalization and Standardization (35 minutes)

* Understanding feature scaling
* Min-max scaling
* Z-score standardization
* Robust scaling for outliers
* When to use each scaling method
* Impact on model performance

#### Topic 5: Advanced Feature Engineering (30 minutes)

* Creating interaction features
* Polynomial features
* Time-based features
* Aggregation features
* Feature selection after transformation

#### Breakout Room Activity 2 (25 minutes)

* Applied feature engineering challenge
* Students will implement a complete feature engineering pipeline
* Compare model performance before and after feature engineering

### Wrap-up (10 minutes)

* Review key concepts
* Common pitfalls and best practices
* Resources for further learning
* Q&A session

## Assessment

* Participation in breakout room activities
* Quality of transformed features
* Ability to explain transformation choices
* Improvement in model performance after feature engineering

## Resources

* Scikit-learn documentation: https://scikit-learn.org/stable/modules/preprocessing.html
* Feature Engineering for Machine Learning (O'Reilly book)
* Kaggle feature engineering tutorials