Use Case with python

Use Case: Maximum Likelihood Estimation in Statistical Inference

In the context of the course on Set Theory, Logic, Linear Algebra, Calculus, and Probability and Statistics, let's explore a use case related to statistical inference, specifically Maximum Likelihood Estimation (MLE). MLE is a fundamental concept in statistics that involves finding the parameters of a statistical model that maximize the likelihood of observed data. This use case integrates various mathematical and statistical concepts from the course.

Description:

In this use case, we will focus on applying Maximum Likelihood Estimation to estimate parameters for a probability distribution. MLE is commonly used in machine learning and statistical modeling to find the best-fitting model parameters, such as the mean and variance of a normal distribution.

Key Components:

1. Probability Distributions: Understanding different probability distributions (e.g., normal, binomial) and their probability density functions is crucial for setting up a statistical model.

2. Likelihood Function: The likelihood function expresses how likely the observed data is for various values of the model parameters. It is based on the chosen probability distribution and serves as the foundation for MLE.

3. Derivatives and Calculus: Calculus concepts, such as differentiation, are used to find the maximum of the likelihood function. The first and second derivatives may be involved in optimization.

4. Statistical Inference: MLE is a common method for estimating parameters in statistical inference, which includes hypothesis testing and confidence intervals.

Python Code Example (Maximum Likelihood Estimation):

```python
import numpy as np
from scipy.stats import norm
from scipy.optimize import minimize

# Generate synthetic data from a normal distribution
np.random.seed(0)
data = np.random.normal(loc=3.0, scale=2.0, size=100)

# Likelihood function for a normal distribution
def likelihood(params, data):
    mean, variance = params
    return -np.sum(norm.logpdf(data, loc=mean, scale=np.sqrt(variance)))

# Initial guess for mean and variance
initial_params = [0, 1]

# Find MLE estimates using optimization
result = minimize(likelihood, initial_params, args=(data,))
mle_mean, mle_variance = result.x

print("MLE Estimated Mean:", mle_mean)
print("MLE Estimated Variance:", mle_variance)
```

In this code, we use the likelihood function for a normal distribution to estimate the mean and variance of the distribution that best fits the synthetic data. The scipy.optimize.minimize function is employed to find the parameters that maximize the likelihood.

This use case demonstrates the practical application of mathematical concepts from the course, including probability, calculus, and optimization, in the field of statistical inference using Maximum Likelihood Estimation.