#!/usr/bin/env python

# coding: utf-8

# In[1]:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, classification\_report

from sklearn.datasets import load\_iris

import numpy as np

import statsmodels.api as sm

# Load the dataset

iris = load\_iris()

X = pd.DataFrame(iris.data, columns=iris.feature\_names)

y = iris.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Create and train the logistic regression model

model = LogisticRegression(max\_iter=10000, multi\_class='multinomial')

model.fit(X\_train, y\_train)

# Make predictions and evaluate the model

predictions = model.predict(X\_test)

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, predictions))

print("\nClassification Report:\n", classification\_report(y\_test, predictions))

# Analyze the coefficients

coefficients = pd.DataFrame(model.coef\_, columns=X.columns)

coefficients['intercept'] = model.intercept\_

print("\nCoefficients of the model:\n", coefficients)

# If you want p-values and more statistics, use statsmodels

X\_train\_const = sm.add\_constant(X\_train) # adding a constant

model\_sm = sm.MNLogit(y\_train, X\_train\_const)

result = model\_sm.fit()

print(result.summary())

# In[ ]: