KNN:

# Import necessary libraries

from sklearn.neighbors import KNeighborsClassifier

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

from sklearn.metrics import accuracy\_score, classification\_report

# Sample data (replace this with your own dataset)

# Assume X contains features and y contains labels

X = [[2.0, 3.0], [1.0, 2.0], [4.0, 2.0], [3.0, 3.0]]

y = [0, 0, 1, 1] # Classes corresponding to the data points

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the KNN classifier

knn\_classifier = KNeighborsClassifier(n\_neighbors=3) # You can adjust the value of k

# Train the classifier on the training set

knn\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

predictions = knn\_classifier.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, predictions)

report = classification\_report(y\_test, predictions)

# Print the results

print("Accuracy:", accuracy)

print("Classification Report:\n", report)

Logistic Regression

# Import necessary libraries

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import log\_loss

import pandas as pd

# Load your 'customer churn' dataset (replace 'your\_dataset.csv' with the actual file path)

# Assuming the dataset has columns like 'MonthlyCharges' and 'Churn'

df = pd.read\_csv('your\_dataset.csv')

# Select the independent variable 'MonthlyCharges' and the dependent variable 'Churn'

X = df[['MonthlyCharges']]

y = df['Churn']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the Logistic Regression model

logreg\_model = LogisticRegression()

# Train the model on the training set

logreg\_model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred\_prob = logreg\_model.predict\_proba(X\_test)[:, 1] # Probability of class 1 (Churn)

# Calculate log loss

logloss = log\_loss(y\_test, y\_pred\_prob)

# Print the log loss

print("Log Loss:", logloss)