Bahria University,

Karachi Campus

## LAB EXPERIMENT NO.

12

## LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | Develop a logistic regression model using KNIME that accurately predicts whether a passenger survived the Titanic disaster. The model will be evaluated based on its accuracy, precision. |
| **2** | You are a data analyst working for a healthcare organization, and your task is to predict the likelihood of diabetes in patients using logistic regression in Python. The goal is to identify high - risk individuals to enable early intervention and better management of diabetes. |

Submitted On:

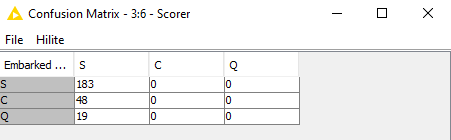
22 may 2024

(Date: DD/MM/YY)

**TASK #1:** Develop a logistic regression model using KNIME that accurately predicts whether a passenger survived the Titanic disaster. The model will be evaluated based on its accuracy, precision.

A diagram of a logistic regression learning

Description automatically generated

 A screenshot of a computer

Description automatically generated

**TASK#2:** You are a data analyst working for a healthcare organization, and your task is to predict the likelihood of diabetes in patients using logistic regression in Python. The goal is to identify high - risk individuals to enable early intervention and better management of diabetes.

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

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

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_diabetes

diabetes\_data = load\_diabetes()

data = pd.DataFrame(diabetes\_data.data, columns=diabetes\_data.feature\_names)

data['Outcome'] = diabetes\_data.target

scaler = StandardScaler()

X = scaler.fit\_transform(data.drop('Outcome', axis=1))

y = data['Outcome']

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

model = LogisticRegression()

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

predictions = model.predict(X\_test)

conf\_matrix = confusion\_matrix(y\_test, predictions)

class\_report = classification\_report(y\_test, predictions, output\_dict=True)

plt.figure(figsize=(10, 7))

sns.heatmap(conf\_matrix, annot=True, fmt='g', cmap='Blues', cbar=False)

plt.title('Confusion Matrix')

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.show()

report = pd.DataFrame(class\_report).T

report = report.round(2)

plt.figure(figsize=(12, 6))

sns.barplot(x=report.index, y=report['precision'], color='blue', label='Precision')

sns.barplot(x=report.index, y=report['recall'], color='red', alpha=0.6, label='Recall')

sns.barplot(x=report.index, y=report['f1-score'], color='green', alpha=0.4, label='F1 Score')

plt.title('Classification Report')

plt.xlabel('Classes')

plt.ylabel('Scores')

plt.xticks(rotation=45)

plt.legend(loc='upper left')

plt.show()

conf\_matrix, class\_report

