Tim Leonhardt

Dr. Rahimi

CSC 428

29th March 2024

Lab 6

# The source code with all results can also be found here: https://www.kaggle.com/code/tleonhardt/csc-428-lab-6

import pandas as pd

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score

# Load the dataset

data = pd.read\_csv('/kaggle/input/kdd-data/KDD-Data.csv')

# Drop any rows with missing values

data.dropna(inplace=True)

# Remove single quotes from column names

data.columns = data.columns.str.replace("'", "")

le = LabelEncoder()

data['protocol\_type'] = le.fit\_transform(data['protocol\_type'])

data['service'] = le.fit\_transform(data['service'])

data['flag'] = le.fit\_transform(data['flag'])

data['class'] = le.fit\_transform(data['class'])

# Split features and target variable

y = data['class']

X = data.drop('class', axis=1)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,random\_state=42, test\_size=0.30, shuffle=True)

#Standardize features: Transform the features

#such that they have a mean of 0 and a standard deviation of 1

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train logistic regression model

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression(max\_iter = 10000000000000)

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

# Predict on the testing set and calculate accuracy

y\_pred = model.predict(X\_test)

logistic\_accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy: {:.2f}%".format(logistic\_accuracy \* 100))

# Initialize and train KNN model

knn = KNeighborsClassifier(n\_neighbors=5)

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

# Predict on the testing set and calculate accuracy

y\_pred = knn.predict(X\_test)

# Calculate accuracy

knn\_accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy: {:.2f}%".format(knn\_accuracy \* 100))

print("Logistic Regression Accuracy: {:.2f}%".format(logistic\_accuracy \* 100))

print("KNN Accuracy: {:.2f}%".format(knn\_accuracy \* 100))

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA black rectangular object with a black stripe

Description automatically generatedA screenshot of a computer

Description automatically generated