**Centro Universitario de Ciencias Exactas e Ingenierías**

**Ingeniería en Computación**
**Practica 2 - ejercicios 2 y 3**


**PRESENTA:**

**Ramirez Gutierrez Hugo Vladimir**
**Código: 220287144**
**Materia:**

**Seminario de Solución de problemas de Inteligencia Artificial 2**

**Docente: Diego Campos Pena**

**Índice:**

**Introducción:**
El objetivo de este reporte es evaluar el rendimiento de varios clasificadores en diferentes conjuntos de datos. Los clasificadores son algoritmos de aprendizaje automático que se utilizan para predecir la clase o categoría de un conjunto de datos. En este estudio, se explorarán tres conjuntos de datos diferentes: Swedish Auto Insurance, Wine Quality y Pima Indians Diabetes. Se evaluarán clasificadores como Regresión Logística, k-Vecinos Más Cercanos (KNN), Máquinas de Soporte Vectorial (SVM), Naive Bayes Gaussiano y Redes Neuronales Artificiales (MLP).

Desarrollo:
Código:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
import warnings

# Desactivar advertencias innecesarias
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=RuntimeWarning)

# Definición de funciones auxiliares

def load_data(file_path):
    """
    Carga los datos desde un archivo CSV.
    """
    return pd.read_csv(file_path)

def categorize_values(data):
    """
    Categoriza los valores de la variable objetivo en función de los
cuartiles.
    """
    quartiles = data['Y'].quantile([0.25, 0.5, 0.75])
    low, medium, high = quartiles.iloc[0], quartiles.iloc[1],
quartiles.iloc[2]
    return data['Y'].apply(lambda x: 'bajo' if x <= low else ('medio'
if x <= medium else 'alto'))

def evaluate_classifier(X, y, classifier):
```

```
    """
    Evalúa el rendimiento de un clasificador en un conjunto de datos
dado.
    """
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
    model = classifier() if classifier != LogisticRegression else
LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted',
zero_division='warn')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    print("\nEvaluación de", classifier.__name__)
    print("Precisión:", accuracy)
    print("Sensibilidad:", recall)
    print("F1 Score:", f1)

def print_dataset_name(name):
    """
    Imprime el nombre del conjunto de datos.
    """
    print("\nConjunto de datos evaluado:", name)


# Carga de datos y evaluación de clasificadores

datasets = {
    "Swedish Auto Insurance": "AutoInsurSweden.csv",
    "Wine Quality": "wine-Quality.csv",
    "Pima Indians Diabetes": "pima-indians-diabetes.csv"
}

for name, file_path in datasets.items():
    print_dataset_name(name)
    data = load_data(file_path)
    if name == "Swedish Auto Insurance":
        data['Y_category'] = categorize_values(data)
        X, y = data[['X']], data['Y_category']
    elif name == "Wine Quality":
        X, y = data.drop('quality', axis=1), data['quality']
    else:
        X, y = data.drop('Class variable (0 or 1)', axis=1),
data['Class variable (0 or 1)']
    classifiers = [LogisticRegression, KNeighborsClassifier, SVC,
GaussianNB, MLPClassifier]
    for classifier in classifiers:
        evaluate_classifier(X, y, classifier)
```

El proceso de evaluación de los clasificadores se lleva a cabo en varias etapas:

1. Preparación de los datos: Se cargan los conjuntos de datos desde archivos CSV utilizando la biblioteca de pandas. Además, en el caso del conjunto de datos Swedish Auto Insurance, se categorizan los valores de la variable objetivo en función de los cuartiles.
2. Definición de clasificadores y métricas: Se importan los clasificadores de scikit-learn y las métricas de evaluación, como precisión, sensibilidad (recall) y F1-Score. Se define una función para evaluar el rendimiento de cada clasificador en un conjunto de datos dado.
3. Evaluación de clasificadores: Se evalúa cada clasificador en los tres conjuntos de datos mencionados. Para cada conjunto de datos, se dividen los datos en conjuntos de entrenamiento y prueba, se entrena el clasificador y se calculan las métricas de evaluación mencionadas anteriormente.
4. Presentación de resultados: Los resultados de la evaluación se presentan en forma de métricas de precisión, sensibilidad y F1-Score para cada clasificador en cada conjunto de datos.

## Resultados:

| | Dataset | Classifier | Accuracy | Precision | Recall | \ |
|---|---|---|---|---|---|---|
| 0 | Swedish Auto Insurance | LogisticRegression | 0.6923 | 0.6923 | 0.6923 | |
| 1 | Swedish Auto Insurance | KNeighborsClassifier | 0.6154 | 0.6154 | 0.6154 | |
| 2 | Swedish Auto Insurance | SVC | 0.6923 | 0.6923 | 0.6923 | |
| 3 | Swedish Auto Insurance | GaussianNB | 0.5385 | 0.5385 | 0.5385 | |
| 4 | Swedish Auto Insurance | MLPClassifier | 0.8462 | 0.8462 | 0.8462 | |
| 5 | Wine Quality | LogisticRegression | 0.5750 | 0.5750 | 0.5750 | |
| 6 | Wine Quality | KNeighborsClassifier | 0.4563 | 0.4563 | 0.4563 | |
| 7 | Wine Quality | SVC | 0.5094 | 0.5094 | 0.5094 | |
| 8 | Wine Quality | GaussianNB | 0.5500 | 0.5500 | 0.5500 | |
| 9 | Wine Quality | MLPClassifier | 0.5563 | 0.5563 | 0.5563 | |
| 10 | Pima Indians Diabetes | LogisticRegression | 0.7468 | 0.7468 | 0.7468 | |
| 11 | Pima Indians Diabetes | KNeighborsClassifier | 0.6623 | 0.6623 | 0.6623 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | Pima Indians Diabetes | SVC | 0.7662 | 0.7662 | 0.7662 |
| 13 | Pima Indians Diabetes | GaussianNB | 0.7662 | 0.7662 | 0.7662 |
| 14 | Pima Indians Diabetes | MLPClassifier | 0.6494 | 0.6494 | 0.6494 |

| | F1 Score |
|---|---|
| 0 | 0.7433 |
| 1 | 0.6838 |
| 2 | 0.7510 |
| 3 | 0.6357 |
| 4 | 0.7756 |
| 5 | 0.5405 |
| 6 | 0.4299 |
| 7 | 0.4618 |
| 8 | 0.5455 |
| 9 | 0.5181 |
| 10 | 0.7482 |
| 11 | 0.6658 |
| 12 | 0.7586 |
| 13 | 0.7679 |
| 14 | 0.6297 |

| | Dataset | Classifier | Accuracy | Precision | Recall | |
|---|---|---|---|---|---|---|
| 13 | Pima Indians Diabetes | GaussianNB | 0.7662 | 0.7662 | 0.7662 |
| 4 | Swedish Auto Insurance | MLPClassifier | 0.8462 | 0.8462 | 0.8462 |
| 8 | Wine Quality | GaussianNB | 0.5500 | 0.5500 | 0.5500 |

En resumen:

1. Swedish Auto Insurance Dataset:
    - El MLPClassifier tiene el puntaje F1 más alto (0.8462), lo que indica un mejor equilibrio entre precisión y sensibilidad en comparación con otros clasificadores.
    - Los clasificadores LogisticRegression y SVC también muestran un rendimiento sólido, con puntajes F1 de 0.7433 y 0.7510 respectivamente.
    - El KNeighborsClassifier tiene el rendimiento más bajo en este conjunto de datos, con un puntaje F1 de 0.6838.
2. Wine Quality Dataset:
    - Todos los clasificadores tienen puntajes F1 relativamente bajos en este conjunto de datos, con valores que oscilan entre 0.4299 y 0.5563.
    - El GaussianNB tiene el puntaje F1 más alto (0.5563), seguido de cerca por el MLPClassifier (0.5563).

- El KNeighborsClassifier tiene el peor rendimiento en este conjunto de datos, con un puntaje F1 de 0.4299.

3. Pima Indians Diabetes Dataset:
    - El SVC tiene el puntaje F1 más alto (0.7662) en este conjunto de datos, seguido de cerca por GaussianNB (0.7662) y LogisticRegression (0.7482).
    - El MLPClassifier tiene el puntaje F1 más bajo (0.6297), aunque sigue siendo bastante competitivo en comparación con los otros conjuntos de datos.

## Conclusión:

El rendimiento de los clasificadores varía según el conjunto de datos, lo que sugiere que no existe un clasificador universalmente superior en todos los casos. La elección del clasificador óptimo puede depender de las características específicas del conjunto de datos y los requisitos del problema. Es importante considerar no solo el puntaje F1, sino también otras métricas como precisión, sensibilidad y exactitud para comprender completamente el rendimiento de un clasificador en un contexto dado.