UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Materia: Seminario de Solución de Problemas de Inteligencia Artificial II

Sección: D05

Profesor: Diego Campos Peña

Practica 2 - Ejercicio 2 y 3 – Clasificadores

Daniel Sánchez Zepeda 220286881

ANTACEDENTES

Los clasificadores como la Regresión Logística, K-Vecinos Cercanos (K-Nearest Neighbors), Máquinas de Vectores de Soporte (Support Vector Machines) y Naive Bayes son herramientas fundamentales en el ámbito del aprendizaje automático para la tarea de asignar categorías a diferentes instancias de datos.La Regresión Logística es un modelo estadístico que se utiliza para predecir la probabilidad de una variable binaria. Este método es particularmente útil cuando se necesita entender la relación entre una variable dependiente categórica y una o más variables independientes. Entre sus ventajas, se encuentra su interpretabilidad y simplicidad, aunque puede tener limitaciones con datos no lineales.

El K-Vecinos Cercanos (KNN) es un algoritmo basado en la similitud que clasifica una instancia de datos en función de la mayoría de los K vecinos más cercanos en el espacio de características. KNN es fácil de implementar y no requiere un proceso de entrenamiento explícito, lo que lo hace versátil. Sin embargo, puede ser ineficiente con grandes conjuntos de datos y sensible a la elección del valor de K.

Las Máquinas de Vectores de Soporte (SVM) son métodos de aprendizaje supervisado que se utilizan tanto para clasificación como para regresión. SVM funciona al encontrar el hiperplano que mejor separa las clases en el espacio de características. Este método es poderoso en problemas de clasificación con alta dimensionalidad y es robusto frente a outliers, pero puede ser complejo y consumir muchos recursos computacionales.

El Naive Bayes es un clasificador probabilístico basado en el teorema de Bayes con la suposición de independencia entre las características. Este modelo es rápido y eficiente, especialmente con grandes volúmenes de datos, y funciona bien con datos categóricos. No obstante, su principal desventaja radica en la suposición de independencia, que rara vez se cumple en la práctica.

Para evaluar el rendimiento de estos clasificadores, se utilizarán tres conjuntos de datos distintos: Swedish Auto Insurance, Wine Quality y Pima Indians Diabetes. El conjunto de datos de Swedish Auto Insurance se centra en datos de seguros automovilísticos suecos y es útil para modelos predictivos en el sector de seguros. El conjunto de datos de Wine Quality contiene características químicas del vino y sus respectivas calificaciones de calidad, proporcionando un buen marco para problemas de clasificación en la industria alimentaria y de bebidas. El conjunto de datos de Pima Indians Diabetes incluye información médica y antecedentes de pacientes de la comunidad indígena Pima, siendo relevante para aplicaciones en salud y medicina predictiva.

INTRODUCCION

En el campo del aprendizaje automático, los clasificadores juegan un papel crucial en la tarea de categorizar instancias de datos. Herramientas como la Regresión Logística, K-Vecinos Cercanos (K-Nearest Neighbors), Máquinas de Vectores de Soporte (Support Vector Machines) y Naive Bayes son ampliamente utilizadas debido a sus distintas características y capacidades para manejar diversos tipos de problemas de clasificación. Cada uno de estos métodos ofrece ventajas y desventajas únicas que los hacen adecuados para diferentes escenarios y tipos de datos.

CODIGO IMPLEMENTADO EN PYTHON COMENTADO

```
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 las advertencias de Sklearn
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=RuntimeWarning)

# Cargar los datasets desde los archivos CSV
swedish_auto_data = pd.read_csv('AutoInsurSweden.csv')
wine_quality_data = pd.read_csv('wine-Quality.csv')
pima_diabetes_data = pd.read_csv('pima-indians-diabetes.csv')

# Categorizar los valores de Y en Swedish Auto Insurance Dataset
quartiles = swedish_auto_data['Y'].quantile([0.25, 0.5, 0.75])
low_limit = quartiles.iloc[0]
medium_limit = quartiles.iloc[1]
high_limit = quartiles.iloc[2]
```

```
def categorize_y(y):
    """Categoriza el valor de Y en bajo, medio o alto según los cuartiles."""
    if y <= low_limit:
        return 'bajo'
    elif y <= medium_limit:
        return 'medio'
    else:
        return 'alto'

swedish_auto_data['Y_category'] = swedish_auto_data['Y'].apply(categorize_y)

def evaluate_classifier(X, y, classifier):
    """
    Entrena y evalúa un clasificador en los datos proporcionados, imprimiendo
    las métricas de evaluación.

    Parámetros:
    X (DataFrame): Características del conjunto de datos.
    y (Series): Etiquetas del conjunto de datos.
    classifier (Class): Clase del clasificador a usar.
    """
    # Dividir los datos en conjunto de entrenamiento y conjunto de prueba
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Instanciar y entrenar el clasificador
    model = classifier(max_iter=1000) if classifier == LogisticRegression else classifier()
    model.fit(X_train, y_train)

    # Hacer predicciones
    y_pred = model.predict(X_test)

    # Calcular métricas de evaluación
```

```
    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')

    # Imprimir métricas
    print(f"\nMétricas para {classifier.__name__}:")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1 Score: {f1:.4f}")

# Swedish Auto Insurance Dataset
print("\nSwedish Auto Insurance Dataset:")
X_swedish = swedish_auto_data[['X']]
y_swedish = swedish_auto_data['Y_category']
evaluate_classifier(X_swedish, y_swedish, LogisticRegression)
evaluate_classifier(X_swedish, y_swedish, KNeighborsClassifier)
evaluate_classifier(X_swedish, y_swedish, SVC)
evaluate_classifier(X_swedish, y_swedish, GaussianNB)
evaluate_classifier(X_swedish, y_swedish, MLPClassifier)

# Wine Quality Dataset
print("\nWine Quality Dataset:")
X_wine = wine_quality_data.drop('quality', axis=1)
y_wine = wine_quality_data['quality']
evaluate_classifier(X_wine, y_wine, LogisticRegression)
evaluate_classifier(X_wine, y_wine, KNeighborsClassifier)
evaluate_classifier(X_wine, y_wine, SVC)
evaluate_classifier(X_wine, y_wine, GaussianNB)
evaluate_classifier(X_wine, y_wine, MLPClassifier)

# Pima Indians Diabetes Dataset
```

```
print("\nPima Indians Diabetes Dataset:")
X_pima = pima_diabetes_data.drop('Class variable (0 or 1)', axis=1)
y_pima = pima_diabetes_data['Class variable (0 or 1)']
evaluate_classifier(X_pima, y_pima, LogisticRegression)
evaluate_classifier(X_pima, y_pima, KNeighborsClassifier)
evaluate_classifier(X_pima, y_pima, SVC)
evaluate_classifier(X_pima, y_pima, GaussianNB)
evaluate_classifier(X_pima, y_pima, MLPClassifier)
```

Funcionalidad del código:

- Carga y prepara los datos.
- Categorización de la variable objetivo en el conjunto de datos de seguros de auto sueco.
- Implementación de una función para entrenar y evaluar varios clasificadores utilizando distintas métricas.
- Evaluación del rendimiento de cinco clasificadores en tres conjuntos de datos diferentes.

- Accuracy: La proporción de predicciones correctas sobre el total de predicciones.
- Precision: La proporción de verdaderos positivos sobre el total de predicciones
- positivas.
- Recall: La proporción de verdaderos positivos sobre el total de valores verdaderos
- en el conjunto de datos.
- F1-score: La media armónica de precisión y recall, que proporciona una medida de
- precisión equilibrada.

Resultados

En términos generales, el clasificador óptimo depende del conjunto de datos y la tarea específica que se desea realizar. Es fundamental analizar las métricas de evaluación y las particularidades de cada conjunto de datos para elegir el clasificador que mejor se adapte a las necesidades del problema. Cada conjunto de datos tiene características únicas, como la distribución de las clases, la cantidad de ruido y la dimensionalidad, que pueden influir significativamente en el rendimiento de los clasificadores.

Por ejemplo, un clasificador como la Regresión Logística puede ser muy efectivo en situaciones donde las relaciones entre las variables independientes y la dependiente son lineales. Sin embargo, en conjuntos de datos con relaciones no lineales, algoritmos como K-Vecinos Cercanos o Máquinas de Vectores de Soporte pueden ofrecer un mejor rendimiento. Por otro lado, clasificadores como Naive Bayes pueden ser más adecuados para conjuntos de datos con características independientes y categóricas.

Además de la precisión, otras métricas de evaluación como la precisión, el recall y el F1 score deben ser consideradas, especialmente en problemas donde el costo de los falsos positivos o falsos negativos es alto. Estas métricas proporcionan una visión más completa del desempeño del clasificador en diferentes aspectos.

También es crucial tener en cuenta factores como la interpretabilidad del modelo, el tiempo de entrenamiento y predicción, y la escalabilidad del algoritmo cuando se selecciona un clasificador. Por ejemplo, mientras que los modelos de redes neuronales pueden ofrecer alta precisión en conjuntos de datos complejos, su interpretabilidad y los recursos computacionales que requieren pueden ser desventajas en ciertos contextos.

En resumen, no existe un clasificador universalmente mejor para todas las situaciones. La selección del clasificador más adecuado debe basarse en un análisis cuidadoso de las características del conjunto de datos y las métricas de evaluación, así como en consideraciones prácticas y contextuales específicas de la tarea en cuestión.

DATASET	ALGORITMO	ACCURACY	PRECISION	RECALL	F1-SCORE
Swedish Auto Insurance	Logistic Regresión	0.6923	0.8654	0.6923	0.7433
	KNeighborsClassifier	0.6154	0.8615	0.6154	0.6838
	SVC	0.6923	0.8718	0.6923	0.7510
	Naive Bayes	0.5385	0.8718	0.5385	0.6357
	MLPClassifier	0.8462	0.7160	0.8462	0.7756
Wine Quality	Logistic Regresión	0.5750	0.5287	0.5750	0.5405
	KNeighborsClassifier	0.4563	0.4223	0.4563	0.5645
	SVC	0.5094	0.5645	0.5094	0.4618
	Naive Bayes	0.5500	0.5423	0.5500	0.5455
	MLPClassifier	0.5656	0.5435	0.5656	0.5268
Pima Indians Diabetes	Logistic Regresión	0.7468	0.7502	0.7468	0.7482
	KNeighborsClassifier	0.6623	0.6712	0.6623	0.6658
	SVC	0.7662	0.7613	0.7662	0.7586
	Naive Bayes	0.7662	0.7707	0.7662	0.7679
	MLPClassifier	0.7078	0.6967	0.7078	0.6901

Mejor método para cada dataset:

Dataset	Accuracy	Recall	Best Classifier (Based on F1-Score)	Precision
Swedish Auto Insurance	0.6923	0.6923	SVC	0.8718
Wine Quality	0.5094	0.5094	SVC	0.5645
Pima Indians Diabetes	0.7662	0.7662 and 0.7662	SVC and Naive Bayes (tied)	0.7613 and 0.7707

CONCLUSION

Para evaluar el rendimiento de estos clasificadores, se utilizaron tres conjuntos de datos distintos: Swedish Auto Insurance, Wine Quality y Pima Indians Diabetes. Cada uno de estos conjuntos de datos presenta características únicas que influyen en el rendimiento de los clasificadores. El conjunto de datos de Swedish Auto Insurance, centrado en datos de seguros automovilísticos suecos, es útil para modelos predictivos en el sector de seguros. El conjunto de datos de Wine Quality, que contiene características químicas del vino y sus respectivas calificaciones de calidad, proporciona un buen marco para problemas de clasificación en la industria alimentaria y de bebidas. El conjunto de datos de Pima Indians Diabetes incluye información médica y antecedentes de pacientes de la comunidad indígena Pima, siendo relevante para aplicaciones en salud y medicina predictiva.

La elección del mejor clasificador para un problema específico debe basarse en un análisis exhaustivo de las características del conjunto de datos y en la evaluación de múltiples métricas de desempeño. Métricas como la precisión, el recall y el F1 score ofrecen una visión más completa del rendimiento del clasificador, especialmente en contextos donde el costo de los errores de clasificación es significativo.

Además de las métricas de desempeño, factores como la interpretabilidad del modelo, el tiempo de entrenamiento y predicción, y la escalabilidad del algoritmo también deben ser considerados. Por ejemplo, aunque los modelos de redes neuronales pueden proporcionar alta precisión en conjuntos de datos complejos, su interpretabilidad y los recursos computacionales que requieren pueden ser desventajas en ciertos contextos.

En conclusión, no existe un clasificador universalmente superior para todas las situaciones. La selección del clasificador más adecuado debe basarse en un análisis cuidadoso y equilibrado de las características del conjunto de datos, las métricas de evaluación relevantes y las consideraciones prácticas específicas de la tarea en cuestión. Solo a través de un enfoque meticuloso y bien informado se puede lograr una clasificación óptima que satisfaga las necesidades del problema específico.