Proyecto final IA

David Santiago Orozco López

Inteligencia Artificial
Presentado A: Francisco Carlos Calderón Bocanegra

Pontificia Universidad
Javeriana
2022

1. Tabla de contenido

1. INTRODUCCIÓN

1.1 Planteamiento del problema

La inteligencia artificial (IA), está cada vez más presente en nuestras vidas. Una buena definición de la misma sería la combinación de algoritmos, que intentan simular algunas acciones de los humanos o mejor aún, ir más allá de la inteligencia humana.

En este sentido, el siguiente informe va a presentar la aplicación de la inteligencia artificial por medio del uso del Matching Learning el cual busca desarrollar un clasificador de ciertas características de manera automática, a un conjunto de datos que se caracteriza por denominar ciertos parámetros a diferentes artistas musicales con una canción en específico, ejemplo: (Shakira- Waka Waka- Bailable 100%).

1.2 Definición del problema

1.2.1 Problema general

Como clasificar un dataset a partir de las características que tengan canciones musicales.

1.2.2 Especificación del problema

Más específicamente se usarán 3 clasificadores, los cuales son: regresión logística, SVN, KNN. Ya con estos clasificadores se encontrará el que mejor los clasifique por medio de parámetros como MCC y f1_score.

1.1 Objetivos

1.4.1 Objetivo general

1. Encontrar el mejor clasificador de los 3, para parametrizar ciertas canciones en diferentes categorías.

2. MARCO TEÓRICO

2.1 Matching Learning

El aprendizaje automático es una rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para imitar la forma en que los humanos aprenden, mejorando gradualmente su precisión. [1]

2.2 ¿Cómo funciona el Matching Learning?

1. Un proceso de decisión: en general, los algoritmos de aprendizaje automático se utilizan para hacer una predicción o clasificación. En función de algunos datos de entrada, que se pueden etiquetar o no, su algoritmo producirá una estimación sobre un patrón en los datos.

2. Una función de error: una función de error evalúa la predicción del modelo. Si hay ejemplos conocidos, una función de error puede hacer una comparación para evaluar la precisión del modelo.

3. Un proceso de optimización del modelo: si el modelo puede ajustarse mejor a los puntos de datos en el conjunto de entrenamiento, entonces los pesos se ajustan para reducir la discrepancia entre el ejemplo conocido y la estimación del modelo. El algoritmo repetirá este proceso de "evaluar y optimizar", actualizando los pesos de forma autónoma hasta alcanzar un umbral de precisión.

2.3 Métodos de aprendizaje automático

- Aprendizaje automático supervisado.
- Aprendizaje automático no supervisado.
- Aprendizaje semisupervisado.

2.4 Algoritmos comunes de aprendizaje automático

- **Neural networks:** las redes neuronales simulan la forma en que funciona el cerebro humano, con una gran cantidad de nodos de procesamiento vinculados. Las redes neuronales son buenas para reconocer patrones y juegan un papel importante en aplicaciones que incluyen traducción de lenguaje natural, reconocimiento de imágenes, reconocimiento de voz y creación de imágenes.
- **Linear regression:** este algoritmo se utiliza para predecir valores numéricos, basándose en una relación lineal entre diferentes valores. Por ejemplo, la técnica podría usarse para predecir los precios de la vivienda en función de los datos históricos del área.
- **Logistic regression: :** este algoritmo de aprendizaje supervisado hace predicciones para variables de respuesta categóricas, como respuestas de "sí/no" a las preguntas. Se puede utilizar para aplicaciones como la clasificación de spam y el control de calidad en una línea de producción.
- **Clustering:** mediante el aprendizaje no supervisado, los algoritmos de agrupamiento pueden identificar patrones en los datos para poder agruparlos. Las computadoras pueden ayudar a los científicos de datos al identificar las diferencias entre los elementos de datos que los humanos han pasado por alto.
- **Decision trees:** los árboles de decisión se pueden usar tanto para predecir valores numéricos (regresión) como para clasificar datos en categorías. Los árboles de decisión utilizan una secuencia de ramificación de decisiones vinculadas que se pueden representar con un diagrama de árbol. Una de las ventajas de los árboles de decisión es que son fáciles de validar y auditar, a diferencia de la caja negra de la red neuronal.
- **Random forests:** en un bosque aleatorio, el algoritmo de aprendizaje automático predice un valor o categoría al combinar los resultados de varios árboles de decisión.

3. ESTADO DEL ARTE

3.1 Casos de uso de aprendizaje automático en el mundo real.

Reconocimiento de voz: también se conoce como reconocimiento de voz automático (ASR), reconocimiento de voz por computadora o conversión de voz a texto, y es una capacidad que utiliza el procesamiento de lenguaje natural (NLP) para traducir el habla humana a un formato escrito. Muchos dispositivos móviles incorporan reconocimiento de voz en sus sistemas para realizar búsquedas por voz, por ejemplo, Siri, o mejorar la accesibilidad para enviar mensajes de texto.

Servicio al cliente: Los chatbots en línea están reemplazando a los agentes humanos a lo largo del viaje del cliente, cambiando la forma en que pensamos sobre la participación del cliente en los sitios web y las plataformas de redes sociales. Los chatbots responden preguntas frecuentes (FAQ) sobre temas como el envío, o brindan asesoramiento personalizado, venta cruzada de productos o sugerencias de tallas para los usuarios. Los ejemplos incluyen agentes virtuales en sitios de comercio electrónico; bots de mensajería, utilizando Slack y Facebook Messenger; y tareas que suelen realizar los asistentes virtuales y los asistentes de voz.

Visión por computadora: esta tecnología de inteligencia artificial permite que las computadoras obtengan información significativa de imágenes digitales, videos y otras entradas visuales, y luego tomen la acción apropiada. Impulsada por redes

neuronales convolucionales, la visión por computadora tiene aplicaciones en el etiquetado de fotografías en las redes sociales, imágenes de radiología en el cuidado de la salud y automóviles autónomos en la industria automotriz.

Motores de recomendación: utilizando datos de comportamiento de consumo anteriores, los algoritmos de IA pueden ayudar a descubrir tendencias de datos que se pueden usar para desarrollar estrategias de venta cruzada más efectivas. Los minoristas en línea utilizan este enfoque para hacer recomendaciones de productos relevantes a los clientes durante el proceso de pago.

Negociación de acciones automatizada: diseñadas para optimizar las carteras de acciones, las plataformas de negociación de alta frecuencia impulsadas por IA realizan miles o incluso millones de transacciones por día sin intervención humana.

Detección de fraude: los bancos y otras instituciones financieras pueden usar el aprendizaje automático para detectar transacciones sospechosas. El aprendizaje supervisado puede entrenar un modelo utilizando información sobre transacciones fraudulentas conocidas. La detección de anomalías puede identificar transacciones que parecen atípicas y merecen una mayor investigación.

4. PROPUESTA DE LA SOLUCION

4.1 Propuesta de la solución

Diseñar un software en Python que permita la clasificación de manera automática de un dataset que es basado en la calificación de caraterististicas, por medio de escalas (numéricas y de porcentaje) a diferentes artistas musicales con sus respectivas canciones.

4.2 Procedimiento de la solución

Para este apartado, se van a hacer descripciones generales del código por medio de un título y una explicación específica a través del uso de los comentarios en cada renglón del código.

Declaración de bibliotecas para usar cada uno de los clasificadores:

```
"from sklearn.model_selection import train_test_split\r\n",
"import pandas as pd\r\n",
"import numpy as np\r\n",
"import matplotlib.pyplot as plt\r\n",
"import csv\r\n",
"from sklearn import metrics\r\n",
"from sklearn.decomposition import PCA\r\n",
"\r\n",
"from sklearn.neighbors import KNeighborsClassifier\r\n",
"from sklearn.svm import SVC\r\n",
"from sklearn.linear_model import LogisticRegression\r\n",
"\r\n",
"from sklearn.preprocessing import minmax_scale\r\n",
"from sklearn.preprocessing import StandardScaler\r\n",
"\r\n",
"from sklearn.metrics import classification_report\r\n",
"from sklearn.metrics import confusion_matrix\r\n",
"from sklearn.metrics import roc_auc_score #***AUC-ROC***\r\n",
"from sklearn.metrics import matthews_corrcoef #***MCC***\r\n",
"from sklearn.metrics import f1_score #***F1***\r\n",
"from sklearn.metrics import plot_roc_curve"
```

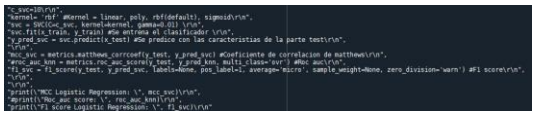
Lectura del dataset y su respectiva limpieza con PCA la cual se utiliza para describir un conjunto de datos en términos de nuevas variables no correlacionadas.

```
"dataset = pd.read_csv('train.csv') #Se extraen los datos \r\n",
"\r\n",
"#Procesamiento de los datos\r\n",
"dataset = dataset.dropna() #Quita las filas con valores NaN, que segun Kaggle es de un procentaje de 2%\r\
"dataset = dataset.drop(['Artist Name'], axis =1) #Quita la columna del nombre del artista\r\n",
"dataset = dataset.drop(['Track Name'], axis=1) #Quita la columna del nombre de la cancion\r\n",
"\r\n",
"x = dataset.drop([\"Class\"], axis = 1).values\r\n",
"\r\n",
"y = dataset[\"Class\"].values\r\n",
"\r\n",
"scaler = StandardScaler()\r\n",
"scaler.fit(x)\r\n",
"x= scaler.fit_transform(x)\r\n",
"\r\n",
"#PCA\r\n",
"pca = PCA(n_components=14, svd_solver='full')\r\n",
"pca.fit(x)\r\n",
"#print(pca.explained_variance_ratio_)\r\n",
"\r\n",
"x_train, x_test, y_train, y_test = train_test_split(x, y, random_state = 0, test_size = 0.2)\r\n",
"\r\n",
"x_train = pca.fit_transform(x_train)\r\n",
"x_test = pca.fit_transform(x_test)"
```

Primer método de clasificación regresión logística.

```
"logistic_regression = LogisticRegression(solver='newton-cg', max_iter=1000, multi_class='multinomial') #Se crea el objeto de logistic regression\r\n",
"#solver('newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga')\r\n",
"logistic_regression.fit(x_train,y_train) #Se entrena el clasificador \r\n",
"\r\n",
"y_pred_lr= logistic_regression.predict(x_test) #Se predice con las caracteristias de la parte test\r\n",
"y_pred=np.round_(x= logistic_regression.predict_proba(x_test) #Se predice las probabilidades con las caracteristias de la parte test\r\n",
"\r\n",
"mcc_lr = metrics.matthews_corrcoef(y_test, y_pred_lr) #Coeficiente de correlacion de matthews\r\n",
"#roc_auc_lr = metrics.roc_auc_score(y_test, y_pred_lr, multi_class='ovr') #Roc auc\r\n",
"f1_lr = f1_score(y_test, y_pred_lr, labels=None, pos_label=1, average='micro', sample_weight=None, zero_division='warn') #F1 score\r\n",
"\r\n",
"\"\\\"\\\"\\r\\n",
"roc = [label: 1] for label in multi_class_series.unique()}\r\n",
"for label in multi_class_series.unique():\r\n",
"    logistic_regression.fit(x_train, y_train == label)\r\n",
"    predictions_proba = logistic_regression.predict_proba(x_test)\r\n",
"    roc[label] += roc_auc_score(y_test, predictions_proba[:,1]) \"\\\"\\\"\\r\\n",
"\r\n",
"print(\"MCC Logistic Regression: \", mcc_lr)\r\n",
"#print(\"Roc_auc score: \", roc_auc_lr)\r\n",
"print(\"F1 score Logistic Regression: \", f1_lr)"
```

Segundo método de clasificación SVM.

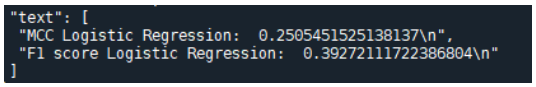


Tercer método de clasificación KNN.

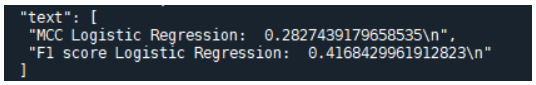
4.1 Resultados de la solución

Parámetros de evaluación de la regresión logística.

Parámetros de evaluación de la SVM



Parámetros de evaluación de la KNN.



5. ANÁLISIS DE LOS RESULTADOS.

A partir de los resultados mostrados anteriormente se puede analizar que los datos están demasiado dispersos entre si ya que muestra una correlación en cada una de ellas bastante baja, esto se debe principalmente a que las variables que se están, son subjetivas y se concentran en todo los rangos posibles, es decir que al ser tantas variables medidas, cada decil hablando en términos estadísticos contiene por lo menos 1800 datos, por lo cual la correlación de cada uno de los datos va a ser bastante grande. También si se habla del valor F vemos que es más grande en todos los casos con respecto a la correlación esto quiere decir que la precisión y la recuperación del modelo es más mayor, con respecto a la correlación de los datos.

6. CONCLUSIONES.

Con este proyecto se puede llegar a la conclusión de que el mejor clasificador que hay para este caso es KNN teniendo un 0.29 de correlación de datos y 0.42 de F1, sin embargo, estos números son bastante bajos, lo que quiere decir que es un modelo bastante malo y que al ser tan subjetivo para cada valoración no se puede clasificar de mejor manera.

7. REFERENCIAS

[1] https://www.ibm.com/cloud/learn/machine-learning