Predicción de la categoría de Índice de Masa Corporal (IMC)

Integrantes:

- Choque Callisaya Fernando Hugo
- Condori Rojas Eugenio Francisco
- Mamani Poma Luis Samuel

INTRODUCCION

Descripción detallada de los campos del dataset

- **ID de persona:** Identificador único para cada individuo del estudio.
- **Género:** Sexo del individuo, categorizado como Masculino o Femenino.
- **Edad:** Edad en años del participante.
- **Ocupación:** Profesión u ocupación del individuo.
- **Duración del sueño (horas):** Número de horas que el participante duerme diariamente.
- **Calidad del sueño (escala: 1-10):** Puntuación subjetiva de la calidad del sueño, donde 1 representa "muy mala" y 10 "excelente".
- **Nivel de actividad física (minutos/día):** Tiempo promedio diario que dedica el participante a actividades físicas.
- **Nivel de estrés (escala: 1-10):** Autoevaluación subjetiva del nivel de estrés, donde 1 indica "muy bajo" y 10 "muy alto".
- **Categoría de IMC:** Variable objetivo. Clasificaca el Índice de Masa Corporal del individuo, por ejemplo:
    - Bajo peso
    - Normal
    - Sobrepeso
- **Presión arterial (sistólica/diastólica):** Medición de la presión arterial, expresada como dos valores (sistólica sobre diastólica).
- **Frecuencia cardíaca (lpm):** Frecuencia cardíaca en reposo del participante, en latidos por minuto.
- **Pasos diarios:** Promedio del número de pasos que realiza el individuo cada día.
- **Trastorno del sueño:** Indica la presencia o ausencia de trastornos del sueño, categorizada como:
    - Ninguno
    - Insomnio
    - Apnea del sueño

Contextualización del problema

La obesidad es uno de los problemas de salud más prevalentes a nivel mundial, afectando a millones de personas y generando complicaciones a largo plazo, como enfermedades cardíacas, diabetes tipo 2 y trastornos musculoesqueléticos. La identificación temprana del riesgo de obesidad y el monitoreo constante del Índice de Masa Corporal (IMC) son esenciales para prevenir y tratar esta condición. Sin embargo, el proceso de diagnóstico puede resultar complejo y subjetivo, dependiendo de los métodos tradicionales utilizados para calcular el IMC.

Objetivo principal del proyecto

El objetivo de este proyecto es desarrollar un modelo de clasificación utilizando técnicas de aprendizaje automático para predecir el riesgo de obesidad en individuos a partir de diversas variables biométricas, como el peso, la altura, la edad, y el género. Este modelo ayudará a automatizar la evaluación del IMC, proporcionándole una herramienta precisa y eficiente a los profesionales de la salud para el diagnóstico temprano de la obesidad.

ALCANCE

Definición clara del alcance del proyecto

Este proyecto está orientado a la creación de un modelo de clasificación que permita predecir el riesgo de obesidad basado en parámetros biométricos sencillos. Se llevará a cabo un análisis de datos, seguido de la implementación y evaluación del modelo utilizando técnicas de aprendizaje automático. El modelo se desarrollará y probará en un entorno de laboratorio, con un conjunto de datos preexistente, y no se realizarán pruebas con datos en tiempo real ni con usuarios externos.

Actividades principales cubiertas por el proyecto

- **Recolección de datos:** Obtención de un conjunto de datos biométricos que incluya variables como peso, altura, edad y género.
- **Preprocesamiento de datos:** Limpieza y preparación de los datos, incluyendo la normalización y la transformación de los valores para el análisis.
- **Desarrollo del modelo:** Implementación de un modelo de clasificación utilizando Random Forest, junto con una prueba de reducción de dimensionalidad con PCA.
- **Evaluación y pruebas:** Medición del rendimiento del modelo utilizando métricas estándar como precisión.
- **Optimización:** Ajuste y mejora de los parámetros del modelo para obtener el mejor rendimiento posible.

Limitaciones y lo que no se incluye en el estudio

Este proyecto se limita a la predicción de la obesidad utilizando únicamente las variables biométricas disponibles en el conjunto de datos. No se considerarán factores adicionales como hábitos alimenticios, actividad física o antecedentes médicos, que también son relevantes para el diagnóstico de la obesidad. Además, el modelo no se validará con datos de pacientes en tiempo real, lo que implica que los resultados no serán aplicables directamente a la práctica clínica sin más pruebas.

DESARROLLO

Descripción de las fases de desarrollo del proyecto

El desarrollo del proyecto se estructuró en varias fases clave, comenzando con la recopilación del conjunto de datos hasta la implementación y evaluación del modelo predictivo. Cada fase fue esencial para garantizar que el modelo estuviera correctamente entrenado y validado.

Recopilación de datos

El conjunto de datos utilizado en este proyecto proviene de Kaggle, específicamente el Sleep Health and Lifestyle Dataset, que consta de 400 filas y 13 columnas. Este dataset abarca una variedad de variables relacionadas con los hábitos de sueño y las condiciones de salud asociadas, como la duración del sueño, la calidad del sueño, los niveles de estrés, y la actividad física diaria.

Preprocesamiento de datos

Una vez obtenido el conjunto de datos, se procedió a la limpieza y preparación de los datos. Esta fase incluyó la eliminación de valores nulos, la transformación de variables categóricas en variables numéricas (como la columna BMI Category), y la normalización de las características numéricas, como la duración del sueño y los niveles de actividad física, para que estuvieran en una escala comparable.

Análisis exploratorio de datos (EDA)

Se realizó un análisis preliminar de los datos para comprender las relaciones entre las variables, identificar patrones relevantes y detectar posibles valores atípicos. Durante esta fase, se emplearon visualizaciones y estadísticas descriptivas para examinar cómo factores como la calidad del sueño o los niveles de estrés podrían influir en el diagnóstico de trastornos del sueño.

Desarrollo del modelo

El modelo predictivo se desarrolló utilizando técnicas de aprendizaje automático, específicamente un modelo de clasificación. Se exploraron algoritmos como Random Forest y Support Vector Machines (SVM) debido a su capacidad para manejar tanto variables categóricas como continuas, así como su robustez frente a datos desequilibrados. Además, se utilizó Cross-validation para garantizar una evaluación confiable del modelo.

Evaluación y ajuste del modelo

Después de entrenar el modelo, se evaluó su rendimiento utilizando métricas estándar como precisión, recall, F1-score y la matriz de confusión. Con base en los resultados obtenidos, se ajustaron los hiperparámetros del modelo para optimizar su desempeño, utilizando técnicas como la búsqueda de cuadrícula (grid search).

Herramientas y tecnologías utilizadas

El proyecto se desarrolló utilizando herramientas y tecnologías de última generación para garantizar un flujo de trabajo eficiente y robusto:

- **Lenguaje de programación:** Python, debido a su popularidad en el ámbito de análisis de datos y machine learning.
- **Librerías de machine learning:**
    - scikit-learn: Para el desarrollo y la evaluación de los modelos de clasificación.
    - Pandas: Para la manipulación y el análisis de datos estructurados.

    - Matplotlib y Seaborn: Para la creación de visualizaciones de los datos y resultados.
- **Plataforma de desarrollo:** Google Colab, que permite realizar el desarrollo y la evaluación del modelo de manera eficiente en la nube sin necesidad de configurar un entorno local complejo.

Recolección de datos: El conjunto de datos fue descargado desde Kaggle, proporcionando información relevante sobre el sueño, la actividad física y la salud cardiovascular de los individuos.

Preprocesamiento: Incluye la limpieza de datos, la imputación de valores faltantes, y la conversión de datos categóricos en valores numéricos.

Modelado: Se implementaron diversos algoritmos de clasificación, como Random Forest, para predecir la presencia de trastornos del sueño en función de las variables introducidas en el conjunto de datos.

FUNCIONALIDAD

Explicación del modelo Random Forest

El Random Forest es un modelo de aprendizaje supervisado basado en un conjunto de árboles de decisión. Un árbol de decisión es un modelo predictivo que utiliza una estructura en forma de árbol para clasificar o predecir resultados a partir de un conjunto de características. Sin embargo, un solo árbol puede ser susceptible al sobreajuste, especialmente cuando los datos son ruidosos o complejos. El Random Forest aborda esta limitación creando varios árboles de decisión de forma aleatoria y promediando sus predicciones para mejorar la generalización del modelo.

Este enfoque tiene varias características destacables:

- **Conjunto de Árboles**: El modelo crea múltiples árboles de decisión basados en subconjuntos aleatorios de los datos, lo que reduce el riesgo de sobreajuste.
- **Agregación de Resultados**: Al utilizar el promedio (en el caso de regresión) o el voto mayoritario (en clasificación), el modelo obtiene una mayor robustez y estabilidad en comparación con un único árbol de decisión.
- **No Linealidad**: Los árboles de decisión, y por ende el Random Forest, pueden modelar relaciones no lineales entre las variables de entrada y la salida, lo que lo convierte en un modelo adecuado para tareas complejas como la clasificación de categorías, donde la relación entre las características y la variable objetivo no necesariamente es lineal.

Ventajas del Random Forest

- **Robustez frente al sobreajuste**: Como cada árbol de decisión se entrena con un subconjunto aleatorio de los datos, el Random Forest es mucho menos propenso al sobreajuste que un solo árbol de decisión. Además, el promedio de los resultados de varios árboles ayuda a reducir la varianza del modelo.
- **Manejo de características correlacionadas**: A diferencia de otros modelos como la regresión logística o los modelos lineales, el Random Forest puede manejar características altamente correlacionadas sin perder poder predictivo. Esto es útil en problemas donde las características pueden no ser independientes.
- **Capacidad para manejar datos desbalanceados**: Aunque el Random Forest no está completamente libre de dificultades con clases desbalanceadas, su naturaleza de agregado de árboles lo hace menos sensible a este problema comparado con modelos más simples como el árbol de decisión o la regresión logística.
- **Importancia de las características**: El Random Forest puede proporcionar estimaciones sobre la importancia relativa de las características para la predicción. Esto es útil en problemas donde queremos entender qué factores influyen más en la predicción de una categoría.

- **Robustez a datos ruidosos**: Debido a que los árboles de decisión tienden a ser robustos a datos ruidosos, y el Random Forest agrega múltiples árboles, el modelo es más resiliente al ruido en los datos.

Aplicación en el contexto del IMC (BMI)

En el contexto de la clasificación de categorías de IMC, el Random Forest es una excelente opción debido a las siguientes razones:

- **Diversidad de características**: Los datos que se utilizan para predecir el IMC incluyen tanto características numéricas (como la edad, la duración del sueño, la presión arterial, etc.) como categóricas (género, ocupación, trastornos del sueño, etc.). El Random Forest puede manejar de manera eficiente estas diversas características, tanto continuas como discretas, sin necesidad de una transformación compleja.
- **Modelo no paramétrico**: No es necesario hacer suposiciones sobre la distribución de los datos (como ocurre en modelos paramétricos, como la regresión logística). Esto es especialmente útil cuando los datos tienen distribuciones complejas o no lineales.
- **Interpretabilidad de resultados**: A pesar de ser un modelo "complejo", el Random Forest permite entender la importancia relativa de cada característica en el modelo mediante el cálculo de la importancia de las variables, lo que puede ayudar a los investigadores o profesionales a identificar qué factores son más relevantes para predecir la categoría del IMC.

Comparación con otros clasificadores

El Random Forest es más robusto que otros clasificadores comunes como la regresión logística o los árboles de decisión individuales debido a las siguientes razones:

- **Reducción de la varianza**: A diferencia de los árboles de decisión individuales, que pueden ser muy sensibles a los cambios en los datos (lo que lleva al sobreajuste), el Random Forest es menos susceptible al sobreajuste gracias a su método de agregación de resultados.
- **Mayor precisión**: En tareas de clasificación complejas, el Random Forest suele ofrecer una mayor precisión que los clasificadores más simples debido a su capacidad para manejar interacciones complejas y relaciones no lineales entre las características.
- **Escalabilidad**: A pesar de ser un modelo relativamente complejo, Random Forest puede ser bastante eficiente incluso cuando se trabaja con grandes conjuntos de datos, lo cual es beneficioso en escenarios con grandes volúmenes de datos como el caso de los estudios de salud.

Referencias

- **Breiman, L.** (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

- **Hastie, T., Tibshirani, R., & Friedman, J.** (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. ISBN: 978-0387848570

PROCESO DEL ANALISIS DE COMPONENTES PRINCIPALES (PCA)

1. Estandarización de los Datos

Antes de aplicar PCA, los datos deben ser estandarizados. Esto asegura que todas las características tengan igual peso al calcular la varianza.

Para cada característica x_i:

$$z_i = (x_i - \mu) / \sigma$$

Donde:

μ: $Media\ de\ la\ característica$
σ: $Desviación\ estándar\ de\ la\ característica$

2. Matriz de Covarianza

El PCA identifica la correlación entre las características calculando la matriz de covarianza C.
Dado un conjunto de datos con n observaciones y p características, la matriz de covarianza es de tamaño p x p y se calcula como:

$$C = \frac{1}{n-1} Z^T Z$$

Donde Z es la matriz de datos estandarizados (n x p). Cada entrada c_{ij} en C representa la covarianza entre las características i y j.

3. Autovalores y Autovectores

Para identificar las direcciones principales de la varianza, calculamos los autovalores y autovectores de la matriz de covarianza.

- Un autovector v es una dirección en el espacio, y un autovalor λ mide cuánta varianza tienen los datos en esa dirección.
- Resolver el problema de autovalores:

  $$Cv = \lambda v$$

  Aquí:
  V es un vector no nulo
  Λ es el autovalor asociado

4. Selección de componentes principales

1. **Ordenar autovalores:** Los autovalores λ1,λ2,…,λp se ordenan en orden descendente.
2. **Seleccionar los más significativos:** Los autovalores más grandes corresponden a las direcciones (autovectores) con mayor varianza.

Si seleccionamos los k primeros autovalores, los autovectores correspondientes forman una matriz V_k de tamaño p×k.

5. Proyección de los Datos

Los datos originales estandarizados se proyectan en el espacio reducido utilizando los autovectores seleccionados:

$$Z_{reducida} = ZV_k$$

$Z_{reducida}$: Matriz de datos transformada con (k dimensiones).

V_k: Matriz de autovectores seleccionados.

RESUMEN DEL TRABAJO REALIZADO

1. Carga y Exploración Inicial del Dataset

```
df = pd.read_csv('Sleep_health_and_lifestyle_dataset.csv')
print(df.head())
```

- **Objetivo:** Cargar los datos en un DataFrame de pandas para inspeccionar su estructura y contenido.
- **Tareas:** Visualizar las primeras filas para identificar el formato de los datos y las columnas disponibles.

2. Preprocesamiento del Dataset

El preprocesamiento es crucial para limpiar y transformar los datos antes del análisis.

Codificación de Variables Categóricas:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender'])
```

- **Objetivo:** Convertir la variable categórica Gender en valores numéricos (0 y 1) para que sea compatible con los algoritmos de ML.
- **Herramienta:** LabelEncoder asigna un número entero a cada categoría única.

One-Hot Encoding para la Columna Occupation

```
df = pd.get_dummies(df, columns=['Occupation'], drop_first=True)
```

- **Objetivo:** Representar la variable categórica Occupation como múltiples columnas binarias.
- **Motivación:** Evitar que los modelos interpreten relaciones ordinales inexistentes entre las categorías.

Separación de Blood Pressure en Columnas

```
df[['Systolic BP', 'Diastolic BP']] = df['Blood Pressure'].str.split('/', expand=True)
df['Systolic BP'] = df['Systolic BP'].astype(int)
df['Diastolic BP'] = df['Diastolic BP'].astype(int)
df = df.drop(columns=['Blood Pressure'])
```

- **Objetivo:** Dividir Blood Pressure en dos columnas numéricas (Systolic BP y Diastolic BP) para análisis detallado.
- **Tareas:** Separar las mediciones sistólica y diastólica y eliminar la columna original.

Manejo de Valores Nulos en Sleep Disorder

```
df['Sleep Disorder'] = df['Sleep Disorder'].fillna('None')
df['Sleep Disorder'] = label_encoder.fit_transform(df['Sleep Disorder'])
```

- **Objetivo:** Imputar valores faltantes con una categoría (None) y codificar la columna.

Estandarización de Variables Numéricas

```
from sklearn.preprocessing import StandardScaler
numeric_columns = ['Age', 'Sleep Duration', 'Physical Activity Level', 'Stress Level',
                   'Systolic BP', 'Diastolic BP', 'Heart Rate', 'Daily Steps']
scaler = StandardScaler()
df[numeric_columns] = scaler.fit_transform(df[numeric_columns])
```

- **Objetivo:** Escalar las variables numéricas para que tengan una media de 0 y una desviación estándar de 1.
- **Motivación:** Mejorar la convergencia y el rendimiento de los modelos sensibles a la escala.

3. Entrenamiento del Modelo Supervisado

División del Dataset

```
from sklearn.model_selection import train_test_split
X = df.drop(['BMI Category'], axis=1)
y = df['BMI Category']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=None
```

- **Objetivo:** Dividir los datos en conjuntos de entrenamiento y prueba.
- **Parámetros:**
    - test_size=0.2: 20% para prueba.
    - random_state=None: Generar diferentes divisiones en cada ejecución.

Entrenamiento y Evaluación

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy_score(y_test, y_pred)
```

- **Modelo:** Random Forest, un modelo de clasificación basado en árboles de decisión.

- **Evaluación:** accuracy_score mide la proporción de predicciones correctas.

Repetición con Varias Divisiones

El código realiza este proceso 100 veces para dos configuraciones de división:

- **80/20**: 80% entrenamiento y 20% prueba.
- **50/50**: 50% entrenamiento y 50% prueba.
- **Motivación:** Estimar la confiabilidad promedio mediante la mediana de los accuracy.

4. Reducción de Dimensionalidad con PCA

```
from sklearn.decomposition import PCA
pca = PCA(n_components=n)
X_pca = pca.fit_transform(X)
```

- **Objetivo:** Reducir el número de variables manteniendo la mayor cantidad posible de varianza.
- **Proceso:**
    1. Calcular los componentes principales (vectores propios) a partir de la matriz de covarianza.
    2. Transformar los datos proyectándolos en las nuevas dimensiones.
- **Evaluación:** Reentrenar el modelo usando los datos reducidos y medir el accuracy para cada configuración de componentes principales.

5. Clustering sin Supervisión con K-Means

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_scaled)
df['Cluster'] = kmeans.labels_
```

- **Objetivo:** Agrupar los datos en 3 clusters basados en similitudes.
- **Proceso:**
    1. Escalar los datos.
    2. Aplicar K-Means para dividir los datos en k=3k = 3k=3 clusters.
    3. Etiquetar los puntos según su cluster asignado.
- **Visualización:** Representar gráficamente los clusters en un plano bidimensional.

6. Integración de PCA con K-Means

```
X_pca = PCA(n_components=2).fit_transform(X_scaled)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans.labels_, cmap='viridis')
```

- **Objetivo:** Reducir la dimensionalidad a 2 componentes principales y visualizar los clusters.
- **Motivación:** Interpretar los resultados del clustering en un espacio reducido.