FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA

PROYECTO INTELIGENCIA ARTIFICIAL

TITULO	**Capstone Project-IBM Employee Attrition Prediction**
AUTOR	**Nombre y Apellido**
	Churqui Mamani Angel
FECHA	20/05/2023

Carrera	Informática Asignatura
Asignatura	INF - 354
Docente	Lic. Moises Silva

PREPROCESAMIENTO DE DATOS Y CLASIFICACIÓN USANDO UN ÁRBOL DE DECISIÓN PARA "PREDICCIÓN DE DESERCIÓN DE EMPLEADOS DE IBM"

Angel Churqui Mamani

Universidad Mayor de San Andres

La Paz, Bolivia

achurqui1@umsa.bo

RESUMEN

"IBM Employee Attrition Prediction" es un proyecto que utiliza técnicas de aprendizaje automático para predecir la atrición de empleados en una organización. El objetivo es desarrollar un modelo de clasificación que pueda identificar qué empleados son propensos a abandonar la organización. El proyecto implica el análisis de datos proporcionados por IBM, la limpieza y preprocesamiento de los datos, la selección de características relevantes y la construcción de un clasificador de árbol de decisión. El modelo entrenado se evalúa utilizando métricas como la precisión y la matriz de confusión. El objetivo final es ayudar a la organización a tomar medidas proactivas para retener a los empleados clave y reducir la rotación de personal.

Palabras clave: atriccion, preprocesamiento, metricas, medidas proactivas, arbol de decision

ABSTRACT

"IBM Employee Attrition Prediction" is a project that uses machine learning techniques to predict employee attrition in an organization. The goal is to develop a classification model that can identify which employees are likely to leave the organization. The project involves the analysis of data provided by IBM, the cleaning and preprocessing of the data, the selection of relevant features and the construction of a decision tree classifier. The working model is evaluated using metrics such as accuracy and the confusion matrix. The ultimate goal is to help the organization take proactive steps to retain key employees and reduce staff turnover.

Keywords: attrition, preprocessing, metrics, proactive measures, decision tree

1. INTRODUCCION

"IBM Employee Attrition Prediction" es un proyecto que tiene como objetivo predecir la atrición de empleados en una organización utilizando técnicas de aprendizaje automático. La atrición de empleados, o la tasa de rotación de personal, es un desafío común en muchas empresas y puede tener un impacto significativo en la productividad y el rendimiento general de la organización.

En este proyecto, se utiliza un conjunto de datos proporcionado por IBM que contiene información relevante sobre los empleados, como su historial laboral, características personales, rendimiento, satisfacción laboral, entre otros. A partir de estos datos, se desarrolla un modelo de clasificación que puede predecir si un

empleado es propenso a abandonar la organización.

El proceso de desarrollo del modelo implica varias etapas. En primer lugar, se realiza un análisis exploratorio de los datos para comprender mejor las características de los empleados y su relación con la atrición. Luego, se lleva a cabo una limpieza de datos y preprocesamiento para asegurarse de que los datos sean adecuados para su uso en el modelo de aprendizaje automático.

A continuación, se seleccionan las características más relevantes y se dividen los datos en conjuntos de entrenamiento y prueba. Se aplica un algoritmo de aprendizaje automático, como un clasificador de árbol de decisión, para entrenar el modelo utilizando el conjunto de entrenamiento. Luego, se evalúa el rendimiento del modelo utilizando el conjunto de prueba y se calculan métricas como la precisión y la matriz de confusión.

El objetivo final es tener un modelo preciso y confiable que pueda predecir la atrición de empleados con base en los datos disponibles. Esto permitirá a la organización tomar medidas proactivas para retener a los empleados clave y reducir la rotación de personal, lo que a su vez puede tener un impacto positivo en la eficiencia y el éxito de la organización.

2. ANALISIS DE PREPROCESAMIENTO

El análisis de preprocesamiento en el proyecto "IBM Employee Attrition Prediction" se refiere a las etapas de limpieza y transformación de los datos antes de ser utilizados para entrenar el modelo de predicción de atrición de empleados. A continuación, se describen las principales tareas de preprocesamiento realizadas:

1. Carga de datos: Se carga el conjunto de datos de empleados de IBM, que contiene información relevante sobre los empleados y su estado de atrición.
2. Limpieza de datos: Se realiza la eliminación de registros incompletos o con valores faltantes para garantizar la integridad de los datos. Esto se logra mediante el uso de la función `dropna()` en Pandas para eliminar filas con valores faltantes.
3. Transformación de datos no numéricos: Si existen columnas con datos no numéricos, se realiza la conversión de estos datos a valores numéricos utilizando métodos como `pd.to_numeric()`.
4. Selección de características: Se eligen las características relevantes que se utilizarán para predecir la atrición de empleados. En el código proporcionado, se utiliza `predictors` para almacenar las características seleccionadas, excluyendo la columna de atrición.
5. División de datos: Los datos se dividen en conjuntos de entrenamiento y prueba utilizando la función `train_test_split()` de Scikit-learn. Esto permite evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento.

En general, el análisis de preprocesamiento se centra en garantizar la calidad de los datos, convertir datos no numéricos a numéricos cuando sea necesario y dividir los datos en conjuntos adecuados para el entrenamiento y la

evaluación del modelo. Estas etapas son fundamentales para garantizar resultados precisos y confiables en el modelo de predicción de atrición de empleados.

3. PREPROCESAMIENTO RELLENAR DATOS

En el proyecto "IBM Employee Attrition Prediction", el análisis de preprocesamiento también incluye el manejo de datos faltantes o valores nulos en el conjunto de datos. A continuación, se describe el proceso de rellenar los datos faltantes:

1. Identificación de datos faltantes: Se realiza una inspección inicial del conjunto de datos para identificar las columnas que contienen valores nulos o faltantes.
2. Evaluación de la naturaleza de los datos faltantes: Se analiza la naturaleza y el patrón de los datos faltantes en cada columna. Esto puede implicar identificar si los datos faltantes son aleatorios o si siguen algún patrón específico.
3. Selección de estrategia de imputación: Dependiendo del tipo de datos y la naturaleza de los datos faltantes, se selecciona una estrategia de imputación adecuada para rellenar los valores faltantes. Algunas estrategias comunes incluyen:
    - **a.** Imputación por valor medio: Se reemplazan los valores faltantes con la media de la columna correspondiente.
    - **b.** Imputación por valor mediano: Se reemplazan los valores faltantes con la mediana de la columna correspondiente.
    - **c.** Imputación por valor más frecuente: Se reemplazan los valores faltantes con el valor más frecuente en la columna correspondiente.
    - **d.** Imputación mediante modelos predictivos: Se utilizan algoritmos de aprendizaje automático para predecir los valores faltantes en función de otras variables.
4. Aplicación de la estrategia de imputación: Se implementa la estrategia seleccionada para rellenar los datos faltantes en el conjunto de datos. Esto se puede lograr utilizando funciones proporcionadas por bibliotecas como Pandas o Scikit-learn.

Es importante tener en cuenta que la selección de la estrategia de imputación debe basarse en el contexto y la naturaleza de los datos. Además, es recomendable evaluar el impacto de la imputación en los resultados del modelo y considerar técnicas adicionales, como el análisis de sensibilidad, para comprender la incertidumbre asociada a los datos imputados.

El proceso de rellenar los datos faltantes forma parte del análisis de preprocesamiento en el proyecto "IBM Employee Attrition Prediction" y tiene como objetivo asegurar la integridad de los datos antes de utilizarlos para el entrenamiento del modelo de predicción de atrición de empleados.

```
#Rellenar los datos faltantes (Numéricos)
medias = df[['Age',
             'DistanceFromHome',
             'MonthlyIncome',
             'NumCompaniesWorked',
             'PercentSalaryHike',
             'TotalWorkingYears',
             'TrainingTimesLastYear',
             'YearsAtCompany',
             'YearsInCurrentRole',
             'YearsSinceLastPromotion',
             'YearsWithCurrManager']].mean()
print(df)
```

FIGURA 1.

donde posterior mente se realiza lo siguiente:

```
#Reemplazar datos categóricos con valores numéricos
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

FIGURA 2

Rellenando de esta manera las columnas que no contengan ni un dato.

4. LIBRERIAS USADAS

Para el preprocesamiento de rellenar datos faltantes en el proyecto "IBM Employee Attrition Prediction", se pueden utilizar varias librerías en Python. A continuación, se mencionan algunas de las librerías comúnmente utilizadas:

- Pandas: Pandas es una biblioteca de manipulación y análisis de datos que proporciona estructuras de datos flexibles y eficientes para trabajar con conjuntos de datos. Se utiliza para identificar y manipular los datos faltantes en el conjunto de datos, así como para aplicar estrategias de imputación.
- NumPy: NumPy es una biblioteca fundamental para la computación científica en Python. Proporciona una amplia gama de funciones y herramientas para el manejo de matrices y cálculos numéricos. Se puede utilizar junto con Pandas para realizar operaciones numéricas en los datos faltantes, como calcular la media o la mediana.
- Scikit-learn: Scikit-learn es una biblioteca de aprendizaje automático en Python que ofrece una variedad de algoritmos y herramientas para tareas de minería de datos y análisis predictivo. Se utiliza para aplicar técnicas de imputación basadas en modelos predictivos, como regresión o clasificación, para rellenar los datos faltantes.
- SimpleImputer (de Scikit-learn): SimpleImputer es una clase de Scikit-learn que proporciona estrategias predefinidas para la imputación de datos faltantes. Se puede utilizar para reemplazar los valores faltantes con medidas estadísticas, como la media, la mediana o el valor más frecuente.

Estas son solo algunas de las librerías utilizadas comúnmente para el preprocesamiento de rellenar datos faltantes en Python. La elección de la librería depende del contexto y los requisitos específicos del proyecto. Es recomendable revisar la documentación y ejemplos de uso de cada librería para aplicar las técnicas adecuadas en función de las necesidades del proyecto.

5. CLASIFICACION USANDO EL ÁRBOL DE DECISION

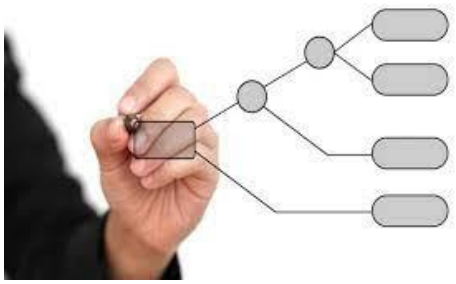

FIGURA 3.

Clasificación utilizando el árbol de decisión es una técnica común en el análisis de datos y aprendizaje automático. El árbol de decisión es un algoritmo de aprendizaje supervisado que construye un modelo de predicción en forma de estructura de árbol. Se utiliza para realizar clasificaciones basadas en la evaluación de características del conjunto de datos.

Aquí hay una descripción general de cómo realizar clasificación utilizando el árbol de decisión:

Preparación de los datos: Antes de construir el árbol de decisión, es necesario preparar los datos. Esto incluye la limpieza de los datos, la selección de características relevantes y la división del conjunto de datos en conjuntos de entrenamiento y prueba.

```
data = data_clean(data)
X = data.drop(['Attrition'], axis=1)
y = data['Attrition']

# Aplicar muestreo excesivo para equilibrar las clases
oversampler = RandomOverSampler()
X_resampled, y_resampled = oversampler.fit_resample(X, y)

# Escalar características
scaler = StandardScaler()
X_resampled = scaler.fit_transform(X_resampled)

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_

# Construir el clasificador de árbol de decisión
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

FIGURA 4.

6. BIBLIOGRAFIA

APRENDEIA.(2023). Limpieza y procesamiento de datos. Recuperado de: https://aprendeia.com/limpieza-y-procesamiento-de-datos-con-codigo-en-python/

ARBOL.(2022). Arboles de decision.Recuperado de:

LUCIDCHART.(2022). Arbol de decisión. Recuperado de: https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-arbol-de-decision

SHENGHUAN YANG.(2021). IBM Employee Attrition Analysis. Recuperado de: https://arxiv.org/pdf/2012.01286.pdf

SOTAQUIRA MIGUEL.(2022). Ejemplos de preprocesamiento. Recuperado de: https://www.codificandobits.com/curso/introduccion-ciencia-de-datos/10-ejemplo-preprocesamiento-de-datos/