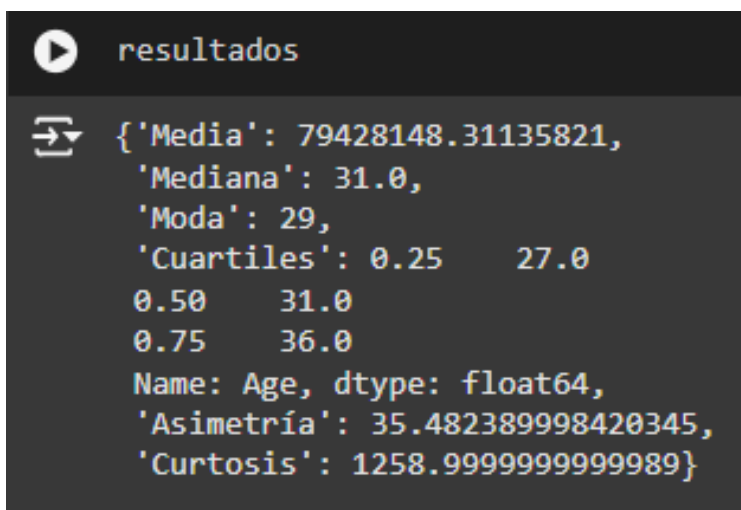


El dataset utilizado proviene de la encuesta "OSMI Mental Health in Tech Survey 2016", cuyo objetivo es evaluar las actitudes hacia la salud mental en el lugar de trabajo dentro del sector tecnológico y analizar la prevalencia de trastornos de salud mental en sus trabajadores. Este conjunto de datos contiene 26 variables que recopilan información demográfica, laboral y de salud mental. Entre estas, se encuentran variables cualitativas como **gender**, **family_history**, **treatment** y **benefits**, así como una variable cuantitativa destacada: **age**. La variable objetivo del análisis es **treatment**, que indica si el encuestado ha buscado tratamiento relacionado con la salud mental.

- **Criterios empleados para la limpieza de datos:**

Uno de los primeros pasos que lleve a cabo fue, revisar el tipo de dato de cada columna, para de esta forma ver si había algún tipo de incongruencia con los datos, por ejemplo: si la columna Age marcaba Object siendo que debería marcar Int64. Después realice las respectivas pruebas de medida de tendencia central y posición a mis variables cuantitativas (Age) para ver si había algún tipo de datos anómalos, dando como resultado una curtosis de 1258, concluyendo que la distribución tiene un pico muy alto con colas gruesas. Esto puede sugiere que hay una gran cantidad de valores extremos, lo que puede ser un indicativo de que existen varios encuestados con edades anómalas o que hay errores en los datos de edad.



```
resultados
{
  'Media': 79428148.31135821,
  'Mediana': 31.0,
  'Moda': 29,
  'Cuartiles': 0.25    27.0
              0.50    31.0
              0.75    36.0
  'Name': 'Age', dtype: float64,
  'Asimetría': 35.482389998420345,
  'Curtosis': 1258.9999999999989
}
```

Imagen 1: Medidas de tendencia central

Aplique el siguiente código para filtrar el dataframe y eliminar los datos outliers y de esta forma realizar un mejor análisis, de la siguiente manera:

```
[ ] q1 = df['Age'].quantile(0.25)
    q3 = df['Age'].quantile(0.75)
    iqr = q3 - q1

    limite_inferior = q1 - 1.5 * iqr
    limite_superior = q3 + 1.5 * iqr

    # Filtrar el dataframe para eliminar outliers
    df_limpio = df[(df['Age'] >= limite_inferior) & (df['Age'] <= limite_superior)]

    # Revisar el nuevo DataFrame
    print(df_limpio['Age'].describe())
```

Imagen 2: Eliminación de los Outliers

Por otra parte, tuve que limpiar todo el apartado de Genre debido a que habían datos que los tomaba de manera diferente, pero eran los mismo, por ejemplo: Male, M y male. Representa lo mismo, pero los usuarios al escribirlo de manera diferente el código lo tomaba como géneros diferentes, así que se aplicó una estandarización, para que de esta forma el código lo tome de la misma manera:

```
[ ] df_limpio.loc[:, 'Gender'] = df_limpio['Gender'].replace({"Female ": "female", 'F': 'female', 'Female': 'female',
    df_limpio.loc[:, 'Gender'] = df_limpio['Gender'].replace({'male': 'Male', 'm': 'Male', "M": "Male", "Male (CIS)":
```

Imagen 3: Estandarizar la columna genero

- **Resultados del análisis de dispersión:**

Rango: 23.96

Un rango de 23.96 horas significa que los datos abarcan casi todo el día (24 horas). Esto implica que las observaciones están distribuidas a lo largo de casi todas las horas del día.

Varianza: 20.73:

Una varianza de 20.73 es considerablemente alta, lo que indica que los datos no están muy concentrados alrededor de la media, sino que están bastante dispersos en diferentes horas del día.

Desviación Estándar: 4.55:

Una desviación estándar de 4.55 horas significa que, en promedio, los valores se desvían de la media en aproximadamente 4.55 horas. Esto también refuerza la idea de que hay una dispersión considerable en los datos.

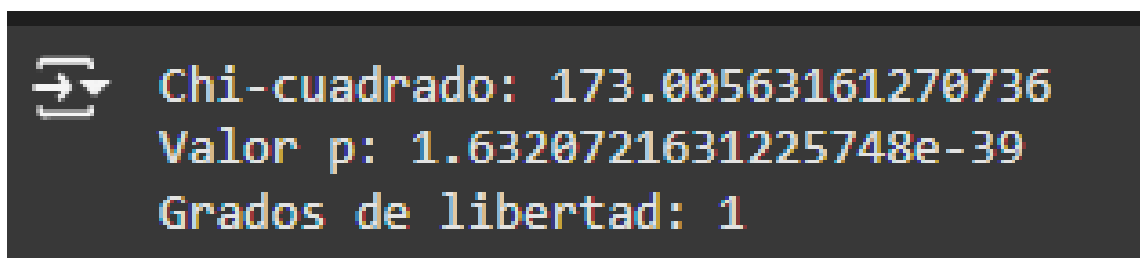
Coeficiente de Variación: 34.12%:

Un coeficiente de variación del 34.12% es moderadamente alto, lo que indica que los datos son bastante variables en comparación con la media. Es decir, hay una gran variabilidad en las horas en las que se realizaron las mediciones o se recogieron los datos.

- **Variable objetivo y variables que afectan a esta**

En primera instancia mi objetivo era realizar el algoritmo que predijera si la persona de la empresa de tecnología se ha realizado un chequeo o se ha hecho algún tratamiento en el apartado de la salud mental, para de esta manera prevenir cualquier tipo de problema a futuro y la variable que mejor encajaba era “**treatment**”. Entonces realice la prueba de chi-cuadrado (Se utiliza para identificar la relación entre una variable categórica) para todos los datos y de esta manera, me base en los 5 datos que tuvieran menor Chi-Cuadrado, dando como resultado lo siguiente:

Family History:

A screenshot of a terminal window showing the results of a Chi-square test. The text is displayed in a monospaced font with a light blue/cyan color on a dark background. It includes a cursor icon on the left. The results are: Chi-cuadrado: 173.00563161270736, Valor p: 1.6320721631225748e-39, and Grados de libertad: 1.

```
→ Chi-cuadrado: 173.00563161270736
Valor p: 1.6320721631225748e-39
Grados de libertad: 1
```

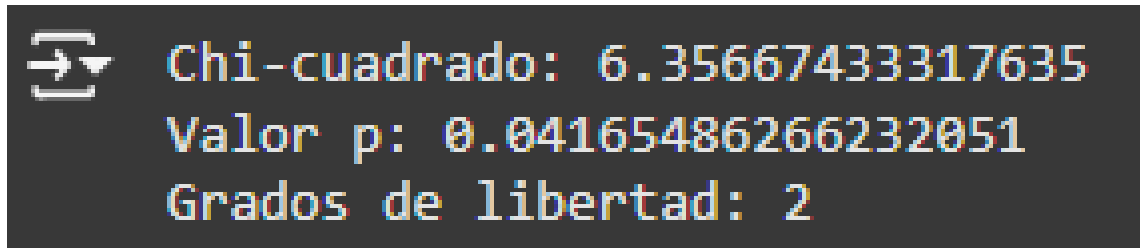
Imagen 4: Chi-Cuadrado de Family History

El valor p es 1.6320721631225748e-39, que es extremadamente bajo (muy cercano a 0). (Generalmente, si el valor p es menor que 0.05, se rechaza la hipótesis nula de que no hay relación entre las variables).

En este caso, como el valor p es mucho menor que 0.05, se puede rechazar la hipótesis nula, lo que significa que existe una relación estadísticamente

significativa entre tener antecedentes familiares de problemas de salud mental y haber buscado tratamiento para la salud mental.

Age:

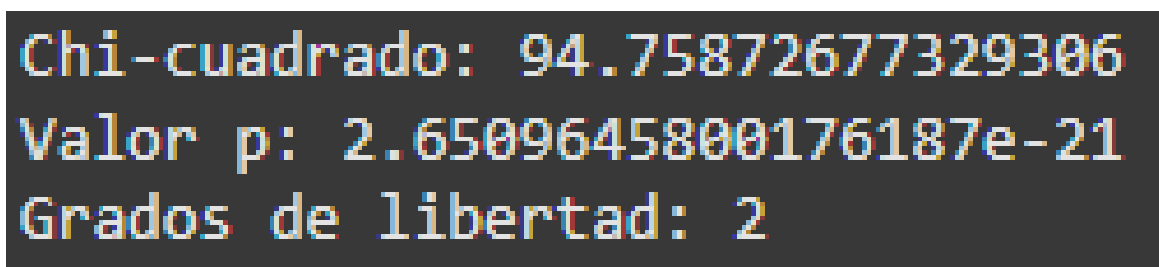
A screenshot of a statistical software output showing the results of a Chi-square test for the variable 'Age'. The text is displayed in a monospaced font on a dark background. It includes a small icon of a box with an arrow pointing right, followed by the Chi-square value, the p-value, and the degrees of freedom.

Chi-cuadrado: 6.35667433317635
Valor p: 0.04165486266232051
Grados de libertad: 2

Imagen 5: Chi-Cuadrado de Age

Al arrojar un valor menor a 0.05 podemos comprobar que no es hipótesis nula, o sea que la Variable Age con respecto a mi variable Objetivo “Treatment” son dependientes.

Care_Options:

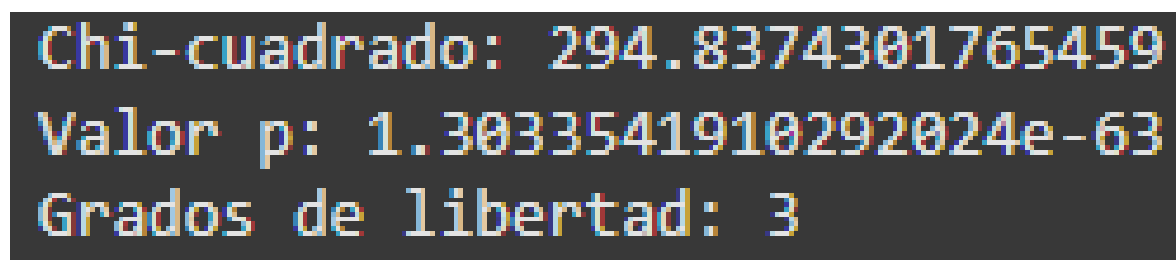
A screenshot of a statistical software output showing the results of a Chi-square test for the variable 'Care_Options'. The text is displayed in a monospaced font on a dark background. It includes the Chi-square value, the p-value, and the degrees of freedom.

Chi-cuadrado: 94.75872677329306
Valor p: 2.6509645800176187e-21
Grados de libertad: 2

Imagen 6: Chi-Cuadrado de Care_Options

Dado que el valor p es mucho menor que 0.05, rechazamos la hipótesis nula. Esto indica que existe una relación estadísticamente significativa entre care_options y treatment.

Work_interfere:



```
Chi-cuadrado: 294.8374301765459
Valor p: 1.3033541910292024e-63
Grados de libertad: 3
```

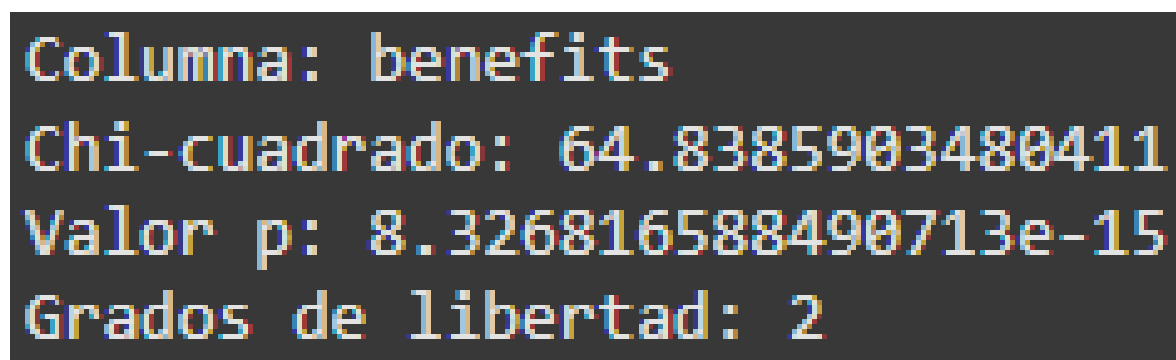
Imagen 7: Chi-Cuadrado de Work_Interfere

Dado el valor p tan bajo, rechazamos la hipótesis nula. Esto confirma que hay una relación estadísticamente significativa entre work_interfere y treatment.

Fuerza de la asociación:

Aunque el estadístico de chi-cuadrado no mide la fuerza de la asociación, el valor tan alto sugiere que el impacto del trabajo (work interfere) influye fuertemente en si una persona busca tratamiento o no.

Benefits:



```
Columna: benefits
Chi-cuadrado: 64.8385903480411
Valor p: 8.326816588490713e-15
Grados de libertad: 2
```

Imagen 8: Chi-Cuadrado de Benefits

Rechazo de la hipótesis nula:

Dado que el valor p es mucho menor que 0.05, rechazamos la hipótesis nula.

Esto confirma que existe una relación estadísticamente significativa entre benefits y treatment.

Significado del resultado:

Las diferencias observadas en las categorías de benefits (por ejemplo, acceso a beneficios, apoyo laboral en salud mental, etc.) tienen un impacto importante en si una persona busca tratamiento o no.

- **Análisis de las gráficas (heatmap):**

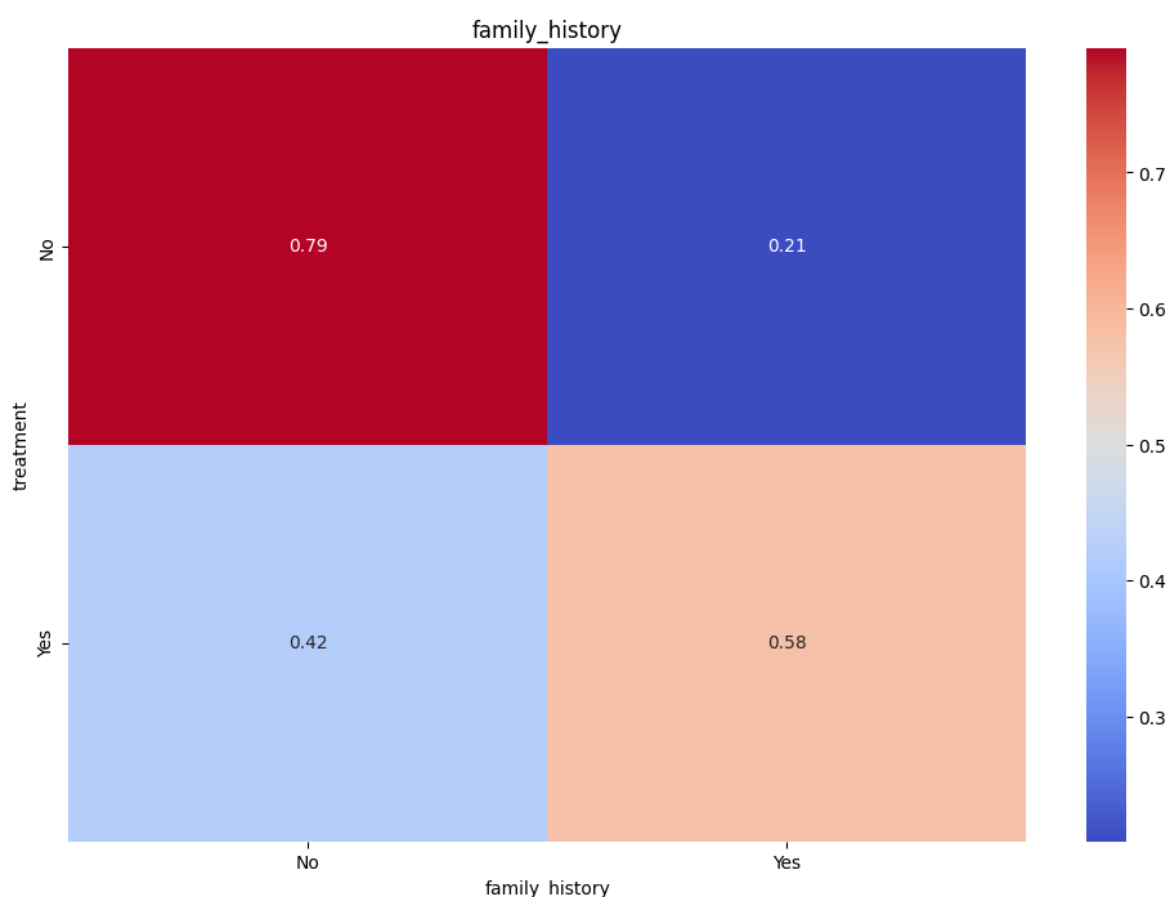


Imagen 9: Mapa de calor Treatment

La gráfica muestra una fuerte relación entre las personas que indicaron no tener familiares con antecedentes de salud mental y aquellas que nunca se han realizado un chequeo o tratamiento relacionado con la salud mental. Esto sugiere que quienes no tienen familiares con antecedentes de este tipo tienden a preocuparse menos por su propia evaluación en este ámbito.

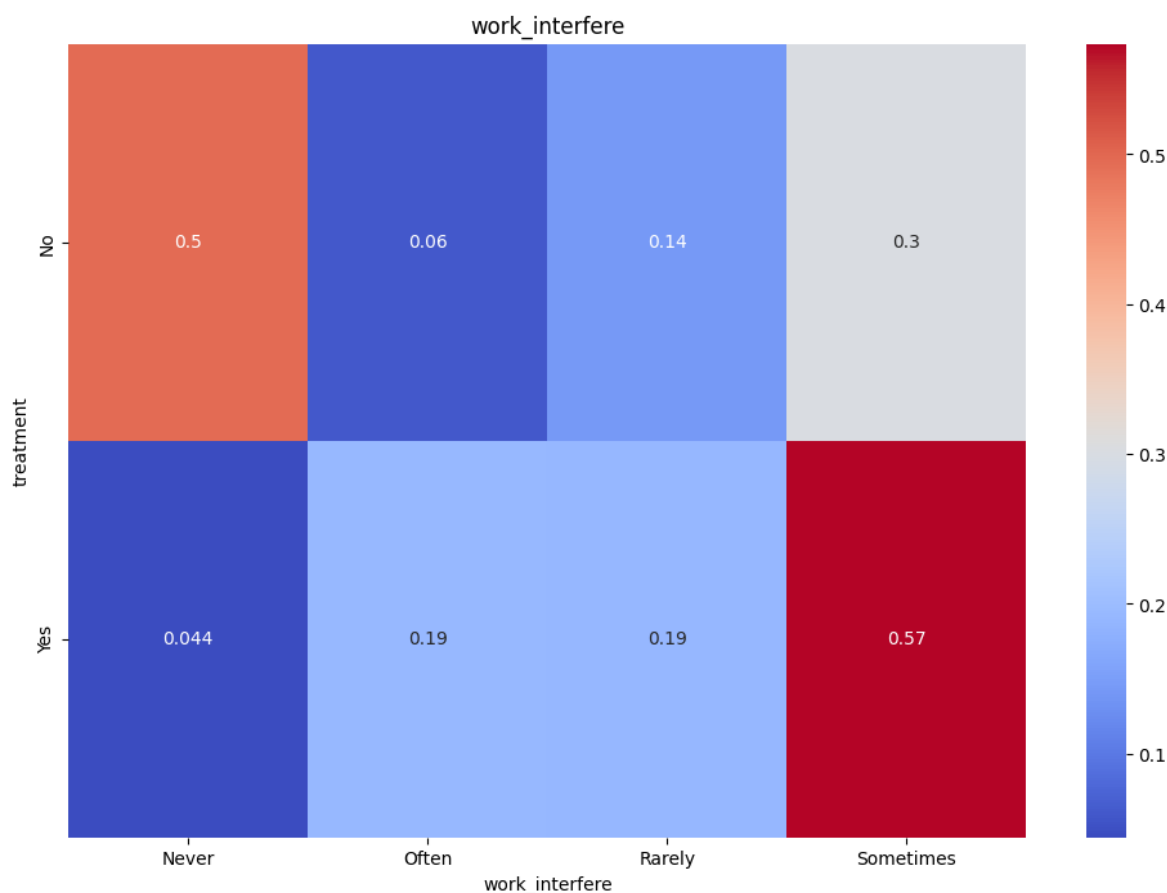


Imagen 10: Mapa de calor de Work_Interfere

Se observa una relación significativa entre los problemas de salud mental y la eficiencia laboral. En la mayoría de los casos, las personas reportan interferencias en su desempeño laboral como resultado de haberse sometido a algún tipo de revisión.

Esto podría explicarse porque quienes deciden realizarse una evaluación de salud mental suelen haber identificado previamente problemas que afectan su vida cotidiana.

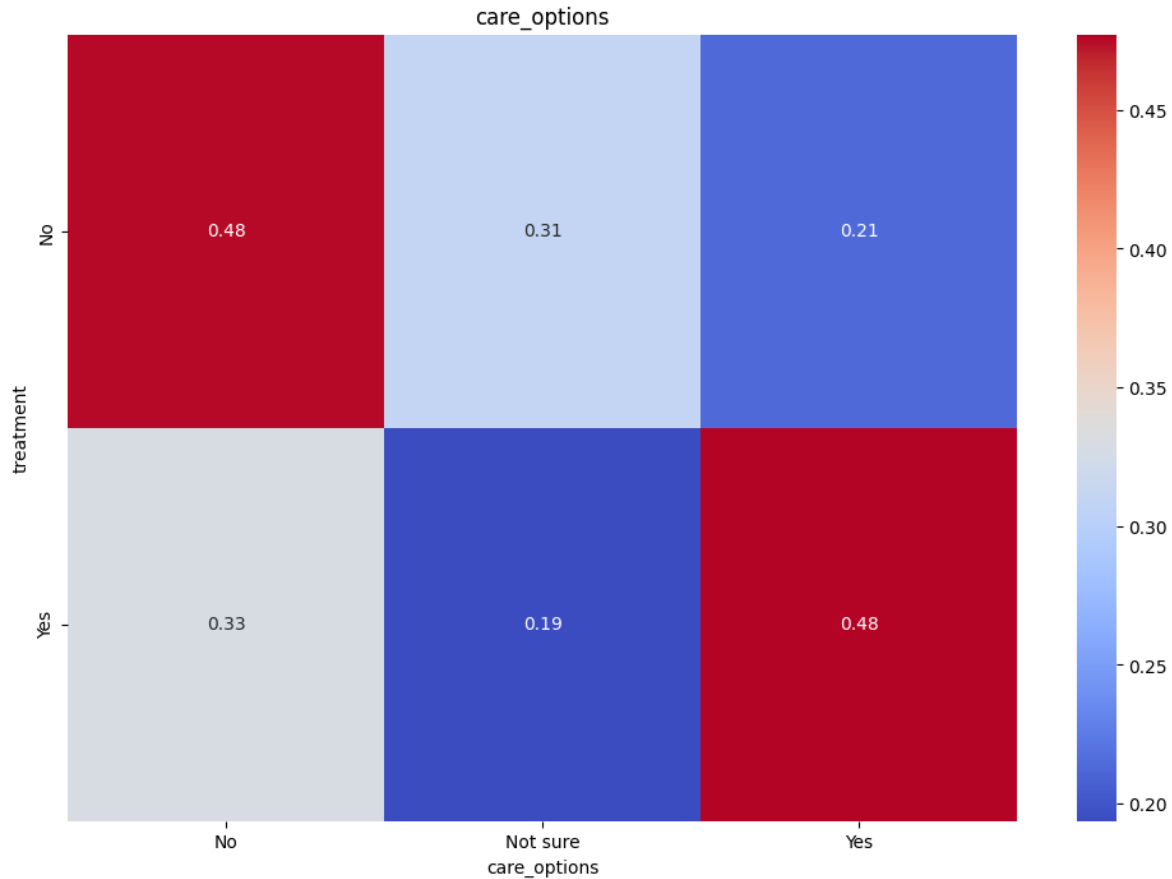


Imagen 11: Mapa de calor de Care_Options

Este análisis resulta bastante interesante, ya que muestra un porcentaje igual entre las personas que conocen las opciones de ayuda relacionadas con la salud mental en su empresa y aquellas que no las conocen. Esto revela que cerca del 50% de los empleados no se han realizado ningún chequeo en el ámbito de la salud mental y, al mismo tiempo, desconocen las opciones de apoyo disponibles en su lugar de trabajo.

- **Pasos para la optimización del modelo de predicción**

En este punto se utilizó la librería sklearn y el método de Regresión logística, debido a que se quiere predecir si el paciente recibió tratamiento o no (treatment). Estos fueron las variables que se utilizaron para el modelo:


```
[32] dataset_modelo = df_limpio[["family_history", "care_options", "Age", "treatment", "work_interfere", "benefits"]]
dataset_modelo = dataset_modelo[dataset_modelo["care_options"].isin(["Yes", "No"])]
dataset_modelo = dataset_modelo[dataset_modelo["work_interfere"].isin(["Never", "Often", "Rarely", "Sometimes"])]
dataset_modelo = dataset_modelo[dataset_modelo["benefits"].isin(["Yes", "No"])]
```

Imagen 12: Variables que afectan a mi variable objetivo

Se aplico el modelo de regresión logística y se utilizo el Grid Search (búsqueda exhaustiva. La idea de la búsqueda exhaustiva es sencilla: simplemente debemos probar TODAS las posibles combinaciones de hiperparámetros, entrenar y validar el modelo con cada una de estas combinaciones y elegir aquella que tenga el mejor desempeño.) Teniendo como resultado lo siguiente:

```
param_grid = {
    "C": [0.001, 0.01, 0.1, 1, 10, 100],
    "penalty": ["l1", "l2"],
    "solver": ["liblinear"]
}

grid_search = GridSearchCV(estimator=modelo, param_grid=param_grid, cv=5, scoring="accuracy")
grid_search.fit(xc, yc)

print("Best parameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)

best_modelo1 = grid_search.best_estimator_

Best parameters: {'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}
Best score: 0.8047303271441203
```

Imagen 13: Resultados combinaciones hiperparametros

Rendimiento aceptable:

El puntaje obtenido demuestra que el modelo tiene un buen equilibrio entre precisión y generalización. Esto sugiere que los datos utilizados tienen patrones predictivos claros relacionados con la variable objetivo (treatment).

Hiperparámetros óptimos:

Los hiperparámetros seleccionados (C=1, l2, liblinear) indican que el modelo está suficientemente regularizado para evitar sobreajuste, mientras que el solver garantiza una optimización eficiente.

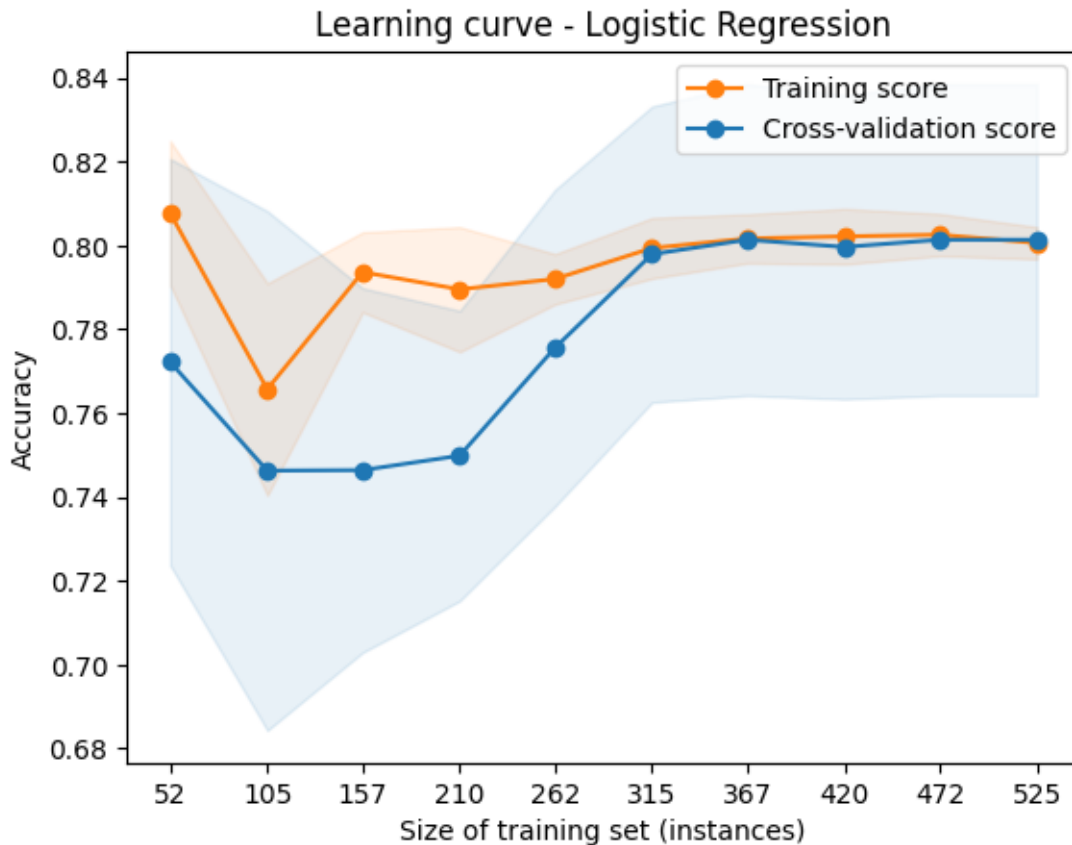


Imagen 14: Curva de aprendizaje

Se puede decir que el modelo cuando tiene pocos datos su precisión es relativamente bajo, pero a medida que sus datos aumentan su nivel de precisión también aumenta, quedando al final en un 80% de esta misma precisión, por otra parte no presenta un sesgo muy grande.

```
[50] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, precision_recall_fscore_support

accuracy = accuracy_score(y_test, y_predict)
precision = precision_score(y_test, y_predict)
recall = recall_score(y_test, y_predict)
f1 = f1_score(y_test, y_predict)

print(f"Precision: {precision}")
print(f"Accuracy: {accuracy}")
print(f"Recall: {recall}")
print(f"F1-Score: {f1}")
```

```
Precision: 0.757396449704142
Accuracy: 0.7642487046632125
Recall: 0.9660377358490566
F1-Score: 0.8490878938640133
```

Imagen 15: Resultado final

Precision: 75%

La precisión del 75.7% indica que, de todas las veces que el modelo predijo la clase positiva, fue correcto el 75.7% de las veces.

Accuracy: 76%

La exactitud del 76.4% significa que el modelo clasifica correctamente el 76.4% de todas las instancias, independientemente de la clase.

Recall: 96%

Un valor de recall de 96.6% es muy alto, lo cual es positivo. Indica que el modelo está capturando casi todos los casos positivos (la mayoría de los verdaderos positivos).

F1-Score: 0.84

El F1-Score de 0.849 indica un buen equilibrio entre precisión y recall. Al ser la media armónica de ambas métricas, refleja que el modelo tiene un buen rendimiento general en la clasificación de la clase positiva.

CONCLUSIÓN

Este análisis y modelo de aprendizaje automático permitieron identificar patrones significativos en los datos relacionados con la salud mental en el ámbito laboral, utilizando técnicas de limpieza, preprocesamiento y optimización de hiperparámetros. A continuación, se resumen los hallazgos principales y las implicaciones:

Importancia de las Variables:

Se identificaron variables clave como antecedentes familiares, opciones de ayuda en la empresa y percepciones sobre la salud mental como factores relacionados con la decisión de buscar tratamiento. Estas variables aportan un entendimiento más profundo de los factores que influyen en la salud mental de los empleados.

Estrategias Efectivas de Modelado:

La regresión logística con regularización L2 fue seleccionada como el modelo más adecuado, logrando un desempeño robusto con una precisión del 80.47%. La optimización de los hiperparámetros garantizó un balance entre simplicidad y eficacia, evitando el sobreajuste.

Insights sobre el Comportamiento de los Datos:

El análisis estadístico reveló que muchas personas desconocen las opciones de ayuda en sus empresas y, a menudo, no buscan tratamiento hasta identificar un impacto en su vida cotidiana. Este hallazgo resalta la necesidad de una mayor concienciación y accesibilidad a programas de bienestar mental en el entorno laboral.

Implicaciones Prácticas:

Este trabajo proporciona una base sólida para que las empresas desarrollen políticas orientadas a mejorar la accesibilidad a recursos de salud mental, enfatizando la importancia de la educación y el apoyo preventivo.

Limitaciones y Oportunidades Futuras:

Aunque el modelo ofrece buenos resultados, sería valioso explorar otros enfoques, como métodos basados en árboles de decisión o redes neuronales, para comparar resultados.

También sería interesante considerar análisis longitudinales o incluir más variables que puedan captar mejor los factores culturales y sociales.

REFERENCIAS BIBLIOGRAFICAS

Fandango, A. (2017). Python Data Analysis - Second Edition: Vol. Second edition. Packt Publishing. (pp 56-66) <https://research-ebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=3b9baf5ea554-36f8-bc46-c7fda76f836c>

Fandango, A. (2017). Python Data Analysis - Second Edition: Vol. Second edition. Packt Publishing. (pp 76-81) <https://research-ebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=3130b5d4-2217-3029-9174-dd119a3a39cc>

Llinás Solano, H. (2017). Estadística descriptiva y distribuciones de probabilidad (pp. 33 - 61). Universidad del Norte. <https://elibronet.bibliotecavirtual.unad.edu.co/es/ereader/unad/70059?page=51>

Llinás Solano, H. (2017). Estadística descriptiva y distribuciones de probabilidad (pp. 66 - 72). Universidad del Norte <https://elibronet.bibliotecavirtual.unad.edu.co/es/ereader/unad/70059?page=84>

Raschka, S., Mirjalili, V. (2017). Python Machine Learning – Second Edition: Vol. 2nd ed. Packt Publishing. (pp 316-318). <https://researchebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=7dd94896-04b3-32bc-afba-c837414115f1>

Aldrin Yim, Claire Chung, & Allen Yu. (2018). Matplotlib for Python Developers, 2nd Edition. Packt Publishing. (pp 109-112) <https://research-ebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=41dc1e4fee2-3bc0-ad97-c2c4388bc71c>

IBM (2024) Data Analysis with Python. Recuperado de <https://cognitiveclass.ai/courses/data-analysis-python>

George Kyriakides, & Konstantinos G. Margaritis. (2019). Hands-On Ensemble Learning with Python : Build Highly Optimized Ensemble Machine Learning Models Using Scikit-learn and Keras. Packt Publishing. (pp 15-20). <https://research-ebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=6d68e3eeac5-3a67-9686-1cd7f819b6b3>

Raschka, S., Mirjalili, V. (2017). Python Machine Learning – Second Edition: Vol. 2nd ed. Packt Publishing. (pp 59-73). <https://researchebscocom.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=66bca30a-5191-322e-8eaa-a22e2e9c1fb1>