Workshop 03 / ETL

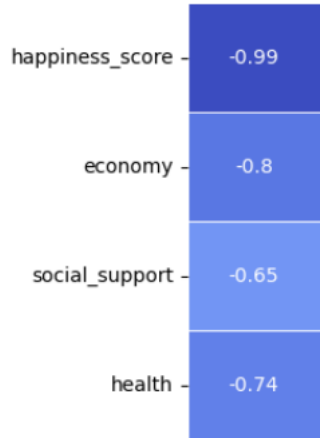
Kevin Artunduaga - 2216155

Introducción

Dados los 5 archivos CSV que contienen información sobre la puntuación de felicidad en diferentes países, entrene un modelo de aprendizaje automático de regresión para predecir la puntuación de felicidad. Cree el EDA/ETL completo para extraer características de los archivos, entrene el modelo usando una división de datos de 70 a 30 (70% para entrenamiento - 30% para pruebas), transmita los datos transformados y luego en el consumidor obtenga los datos y use el modelo entrenado para predecir el puntaje de felicidad y almacenar en una base de datos las predicciones con las respectivas entradas (características), el trabajo completo se presenta en la siguiente figura. Finalmente, extraiga una métrica de rendimiento para evaluar el modelo utilizando los datos de prueba y los datos predichos.

Model training

Para el entrenamiento del modelo hice uso de la librería scikit-learn (sklearn) el cual es una biblioteca de código abierto en Python utilizada para machine learning, para ello hice uso de la técnica de aprendizaje supervisado, la regresión linear en donde busca establecer una relación lineal entre variables en donde predice valores numéricos en este caso el happiness score a partir de variables independientes que las que escogí fueron: health, social support y economy debido a que estas fueron las variables que más correlación tuvieron con el happines score.



Economy: Es la medida respectiva del PIB (producto interno bruto).

Social support: Es la medida de apoyo social que se brinda.

Health: Se refiere a la esperanza de vida o más resumidamente la salud.

Ahora bien para el entrenamiento del modelo se usó el 70% de los datos que en total las filas concatenadas daban un total de 781 en donde esto significaba 547 registros y para el testing se hizo uso de 235 registros osea el 30% de los datos. Cabe recalcar que para poder mantener esta misma selección o split que hacía sklearn se usó el comando de random state que es similar a una semilla y el número es 230. De esta manera se hizo mediante código:

```
X = df_final[["economy", "social_support", "health"]]
y = df_final["happiness_score"]

# Divide los datos en conjuntos de training y testing (70%
training, 30% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=230)

model = LinearRegression()
model.fit(X_train, y_train)
```

Este modelo dispone de un R^2 de aproximadamente el 70.3% esto explica el porcentaje en el que la predicción se ajusta a las variables, no es un mal porcentaje sin embargo no explica la predicción de una manera tan precisa. Para poder hacer uso de este modelo se exportó en el formato pkl y lograr hacer el testing.

Feature selection

En el feature selection como ya expliqué anteriormente la selección de características que escogí más relevantes o útiles para el modelo predictivo fueron la de health, social support y economy. Pero para explicar este proceso de una mejor manera primeramente había que saber que variables disponían todos los csv por igual, para eso se hizo una transformación en donde se eliminaron las variables que difieren estas fueron standar error, region, dystopia residual, lower confidence interval, upper confidence interval, whisker high y whisker low. Aqui enseño una pequeña muestra de las variables que se quedaron luego de la transformación:

country	happiness_rank	happiness_score	economy	social_support	health	freedom	generosity	government_corruption
Norway	1	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964
Denmark	2	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770
Iceland	3	7.504	1.480633	1.610574	0.833552	0.627163	0.475540	0.153527

Como resultado las variables que quedaron fueron: country, happiness rank, happiness score, economy, health, freedom, generosity y goverment corruption. Luego al evaluar la correlación de las variables con happiness score claramente se descartaron el country, el happiness rank y el year porque no eran variables que influyeran en la variable objetivo. Posterior al puntaje de la correlación las que menos destacaron fueron generosity, government corruption y generosity así que estas fueron descartadas en la selección.

Streaming

Para el data streaming se hizo uso de Kafka un broker de mensajería de código abierto que se utiliza para el procesamiento y la transmisión de flujos de datos en

tiempo real (streaming), para hacer uso de este servicio se utilizó docker en donde mediante la imagen de kafka se crearon los contenedores de Kafka y Zookeeper este en especial es un componente central en la arquitectura de Kafka para permitir coordinación de ambos servicios de manera distribuida.

Ahora bien luego de iniciar ambos contenedores es donde entra el funcionamiento en código de Kafka para ello primeramente en el feature selection se seleccionó el 30% de los datos que era para el testing y se exporto en formato de csv, aquí Kafka dispone de dos componente que desempeñan roles específicos, el producer es el responsable de enviar mensajes o datos a los topics, estos topics son canales de mensaje donde se publican los datos para que estén disponibles para el otro componente el consumer.

En cuanto al código del producer se le da el csv del testing data y mediante la librería de kafka-python se usa la instancia del producer para serializar los mensaje en formato JSON y codificarlos en utf-8 y se le asigna una dirección de los brokers de Kafka al que se conectaran. Luego se hace un ciclo para que por cada fila del testing data envíe los mensajes de economy, social support, health y happiness score o el Y test, se le asigna un topic ya previamente creado y se envía el mensaje, este proceso se repite por cada fila que disponga. Aquí enseño como está hecho el código para entenderlo de una mejor manera:

```
def kafka_producer():
    df =
pd.read_csv("C:/Users/kevin/ETL/workshop_03/testing_data.csv")
    producer = KafkaProducer(
        value_serializer=lambda m: json.dumps(m).encode('utf-8'),
        bootstrap_servers=['localhost:9092']
    )

    for index, row in df.iterrows():
        data = {
            "economy": row["economy"],
            "social_support": row["social_support"],
            "health": row["health"],
            "happiness_test": row["happiness_score"]
        }
        producer.send("workshop_03", value=data)
        print("Mensaje enviado")
        time.sleep(0.01)
```

Kafka Consumer

Por otro lado el Consumer de kafka se conecta al topic y recupera los mensajes o datos mediante estos, los consumer se suscriben a los topics para leer los mensajes del producer, y pueda procesar, analizar o almacenar mensajes, en este caso el consumer está encargado de leer los mensajes y extraer cada una de las características de los mensajes y realizar la predicción del happiness score mediante el archivo pkl del modelo entrenado y hacer la predicción mediante las tres características seleccionada en el feature selection.

Luego de hacer esto se ejecuta la conexión a la base de datos de Postgres y se hace la inserción de las características, el Y test y el Y predicción a esta, repitiendo este proceso con cada uno de los mensajes del producer al inicio de la función del consumer se ejecuta el llamado a la librería de kafka-python del Kafka Consumer y se serializan los mensajes en json con el codificado de utf-8, se le asigna la misma dirección a la que estaba el producer y se le asigna el topic al que se conectó el producer. Aqui muestro la función que use en el consumer el cual explica en gran manera el proceso que este hace con los mensajes:

```
def process_and_store_message(message, loaded_model, db_config):
    try:
        # Extrae las características del mensaje
        economy = message["economy"]
        social_support = message["social_support"]
        health = message["health"]
        happiness_score = message["happiness_test"]
        prediction = loaded_model.predict([[economy,
social_support, health]])

        conn = psycopg2.connect(
            host='localhost',
            user=db_config['user'],
            password=db_config['password'],
            database='happiness'
        )
        cursor = conn.cursor()

        insert_data_sql = """
        INSERT INTO happiness_predictions (economy,
social_support, health, happiness_test, happiness_prediction)
        VALUES (%s, %s, %s, %s, %s);
        """
        cursor.execute(insert_data_sql, (economy, social_support,
```

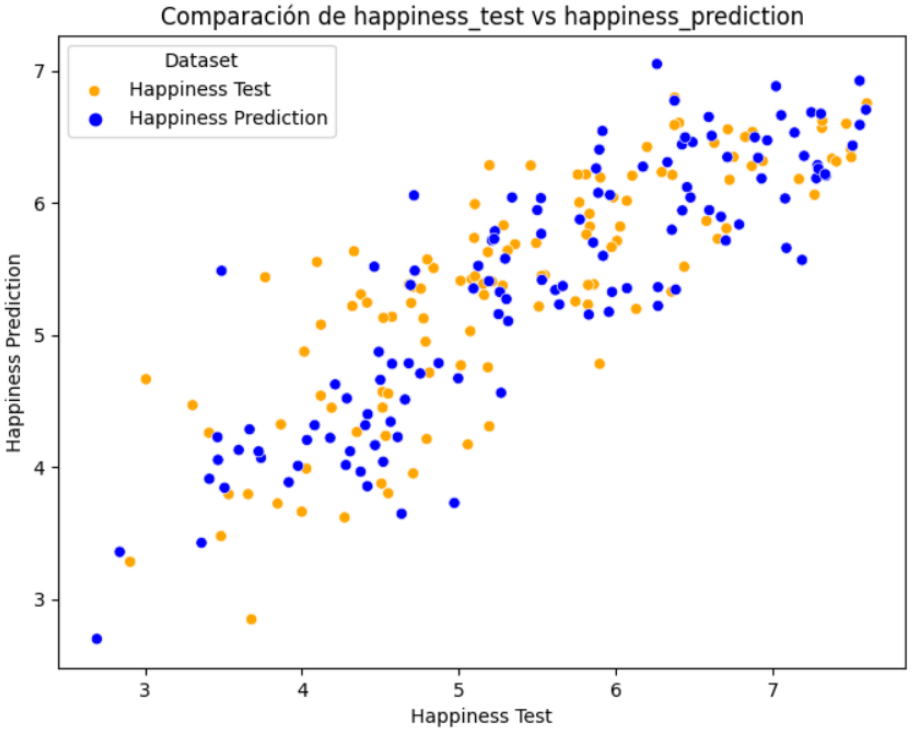
```
health, happiness_score, prediction[0]))

        conn.commit()
        conn.close()
        print("Datos almacenados con éxito en la base de datos.")
```

Model prediction / load

Luego de haber ejecutado el proceso del data streaming quedaron finalmente las 235 filas en la base de datos con las características, el happiness score de testing y el happiness score prediction. Ejemplo de las 5 primeras filas de la base de datos:

id [PK] integer	economy double precision	social_support double precision	health double precision	happiness_test double precision	happiness_prediction double precision
1	0.95148438215255	1.13785350322723	0.54145205020904	5.27899980545044	5.372036470773236
2	0.642	1.236	0.828	5.86	5.3821339762225575
3	1.26637	1.28548	0.90943	6.867	6.277844917765483
4	0.56044	0.95434	0.55449	5.185	4.753976092358699
5	0.84731	0.66366	0.04991	3.866	4.321682255204369

Como se puede ver el hapiness prediction no es 100% preciso con el happiness score debido a el porcentaje del R cuadrado que como ya explique antes solo se ajusta un 70.3% a los datos. Sin embargo se puede notar que estas 3 variables son las que definen en gran parte la calificación de la variable objetivo.



Para ello realice un diagrama de dispersión haciendo la comparación de happines score test y el happiness score prediction, como se puede apreciar en la gráfica se nota la variabilidad entre ambos que a pesar de que no sea la misma su rango de variabilidad tampoco es demasiada, cabe recalcar que este modelo no dispone de las otras características que aportan al happiness score, si bien las otras no tenían tanta correlación en temas de ser precisos si se hubieran analizado todas hasta de pronto hubiera llega el entreno a ser el adecuado, sin embargo lo que buscábamos era mirar cuales era las que más influyen en el happiness score a su vez de que este es uno de los tantos métodos para predecir y la regresión linear tampoco es tan robusto como otros.

Métricas

Para las métricas las que se usaron fueron R^2, MSE (error cuadrático medio) y el RMSE (raíz del error cuadrático medio), estos fueron resultados.

```
R2: 0.7032730315721939
MSE: 0.40783012953633313
RMSE: 0.6386157917999938
```