

PROYECTO FINAL



Interrogantes del Grupo 5:

- Macchia Sebastian
- Paglino Guillermo
- Peña Matias
- Rodriguez Mikael

1. Título del Proyecto:

Recomendación Personalizada de Cervezas: Un Análisis Basado en Preferencias del Consumidor y realización de menu interactivo para usuario

2. Introducción

Contexto del Problema:

El crecimiento constante de la industria cervecera artesanal ha llevado a una diversificación significativa en la oferta de cervezas. La variedad de estilos, sabores y características ha generado un desafío para los consumidores al seleccionar nuevas cervezas que se ajusten a sus preferencias individuales. Este proyecto aborda la necesidad de proporcionar recomendaciones personalizadas de cervezas basadas en las preferencias del consumidor, utilizando técnicas de análisis de datos y machine learning.

Importancia y Relevancia:

La importancia radica en mejorar la experiencia del consumidor al facilitar la exploración y selección de cervezas que se alineen con sus gustos personales. Además, para las cervecerías, comprender las preferencias del mercado puede ayudar en la creación y comercialización de productos más ajustados a la demanda.

Objetivos del Proyecto:

Identificación de Patrones:

Aplicar técnicas de clustering para identificar patrones y agrupaciones en las preferencias de los consumidores a partir de revisiones de cervezas.

Desarrollo de Interfaz Interactiva:

Crear una interfaz interactiva que permita a los usuarios ajustar las puntuaciones de características y recibir recomendaciones de cervezas personalizadas.

Exploración de Graduación de Alcohol:

Analizar y presentar las cervezas con mayor graduación de alcohol, brindando opciones para aquellos que buscan experiencias más intensas.

Documentación y Entrega:

Documentar de manera clara y completa el proceso de análisis y desarrollo del proyecto.

3. Metodología

Datos Utilizados:

En este proyecto, se utilizaron dos conjuntos de datos de revisiones de cervezas obtenidos de las siguientes fuentes:

https://raw.githubusercontent.com/emmanueliarussi/DataScienceCapstone/master/3_MidtermProjects/ProjectBEE/data/beer_reviews_1.csv

https://raw.githubusercontent.com/emmanueliarussi/DataScienceCapstone/master/3_MidtermProjects/ProjectBEE/data/beer_reviews_2.csv

Estos datos contienen información detallada sobre revisiones de cervezas, incluyendo características como aroma, apariencia, sabor, graduación alcohólica, entre otros. La relevancia de estos datos radica en la posibilidad de identificar patrones y agrupaciones en las preferencias de los consumidores y en la calidad percibida de las cervezas.

Herramientas y Tecnologías:

Las herramientas y tecnologías utilizadas en este proyecto son:

Pandas: Librería para manipulación y análisis de datos.

Numpy: Librería para realizar operaciones matemáticas en arreglos y matrices.

Seaborn: Librería para visualización de datos basada en Matplotlib.

Matplotlib: Librería de trazado para crear visualizaciones estáticas, animadas e interactivas en Python.

Scikit-learn: Biblioteca de aprendizaje automático que incluye herramientas para preprocesamiento de datos, modelado y evaluación.

StandardScaler: Herramienta de preprocesamiento para estandarizar características eliminando la media y escalando a la varianza unitaria.

KMeans: Algoritmo de clustering K-Means para agrupar datos.

KElbowVisualizer: Visualizador para ayudar a encontrar el número óptimo de clusters en K-Means.

ipywidgets: Librería para crear widgets interactivos en el entorno de Jupyter (Colab).

IPython.display: Módulo para la gestión de la salida de las celdas en Jupyter (Colab).

SimpleImputer: Estrategias para imputar valores faltantes en datos.

Proceso de Análisis/Desarrollo:

Carga de Datos:

1. Se cargaron dos conjuntos de datos de revisiones de cervezas desde fuentes externas.

✓ Cargar los datos

```
# URL de los conjuntos de datos de revisiones de cerveza
url_beer_reviews_1 = "https://raw.githubusercontent.com/emmanueliarussi/DataScienceCapstone/master/3_MidtermProjects/ProjectBEE/data/beer_reviews_1.csv"
url_beer_reviews_2 = "https://raw.githubusercontent.com/emmanueliarussi/DataScienceCapstone/master/3_MidtermProjects/ProjectBEE/data/beer_reviews_2.csv"

# Utilizando Pandas, se cargan los datos desde las URL y se almacenan en dos DataFrames separados
beer_reviews_1 = pd.read_csv(url_beer_reviews_1)
beer_reviews_2 = pd.read_csv(url_beer_reviews_2)
```

2. Preprocesamiento de Datos:
3. Se combinaron los conjuntos de datos.

✓ Combinacion (concatenacion) de los dos conjuntos de datos

```
[ ] # Utilizando la función concat de Pandas, se combinan los dos conjuntos de datos de revisiones de cerveza en uno solo
# El parámetro ignore_index=True reinicia los índices del DataFrame resultante
beer_reviews = pd.concat([beer_reviews_1, beer_reviews_2], ignore_index=True)
```

4. Se eliminaron valores nulos y duplicados.

✓ Eliminacion de valores nulos y duplicados

```
# Se utiliza el método dropna() para eliminar filas que contienen valores nulos en el DataFrame beer_reviews
# Luego, se utiliza el método drop_duplicates() para eliminar filas duplicadas en el DataFrame resultante
beer_reviews = beer_reviews.dropna().drop_duplicates()
```

5. Se convirtió la columna 'review_time' a formato de fecha y hora.

Conversión de la columna 'review_time' a formato de fecha y hora

```
# Utilizando la función pd.to_datetime(), se convierte la columna 'review_time' del DataFrame beer_reviews  
# La unidad de tiempo se especifica como segundos (unit='s')  
beer_reviews['review_time'] = pd.to_datetime(beer_reviews['review_time'], unit='s')
```

Análisis Exploratorio de Datos:

- Se obtiene las características descriptivas del dataset concatenado.

Estadísticas descriptivas de las características principales

```
# Obtener estadísticas descriptivas de las características principales  
beer_reviews.describe()
```

	brewery_id	review_time	review_overall	review_aroma	review_appearance	review_palate	review_taste	beer_abv	beer_beerid
count	1.586614e+06	1.586614e+06	1.586614e+06	1.586614e+06	1.586614e+06	1.586614e+06	1.586614e+06	1.518829e+06	1.586614e+06
mean	3.130099e+03	1.224089e+09	3.815581e+00	3.735636e+00	3.841642e+00	3.743701e+00	3.792860e+00	7.042387e+00	2.171279e+04
std	5.578104e+03	7.654427e+07	7.206219e-01	6.976167e-01	6.160928e-01	6.822184e-01	7.319696e-01	2.322526e+00	2.181834e+04
min	1.000000e+00	8.406720e+08	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e-02	3.000000e+00
25%	1.430000e+02	1.173224e+09	3.500000e+00	3.500000e+00	3.500000e+00	3.500000e+00	3.500000e+00	5.200000e+00	1.717000e+03
50%	4.290000e+02	1.239203e+09	4.000000e+00	4.000000e+00	4.000000e+00	4.000000e+00	4.000000e+00	6.500000e+00	1.390600e+04
75%	2.372000e+03	1.288568e+09	4.500000e+00	4.000000e+00	4.000000e+00	4.000000e+00	4.500000e+00	8.500000e+00	3.944100e+04
max	2.800300e+04	1.326285e+09	5.000000e+00	5.000000e+00	5.000000e+00	5.000000e+00	5.000000e+00	5.770000e+01	7.731700e+04

- Se agruparon los datos por nombre de cerveza, cervecería y tipo de cerveza.

✓ Agrupación de datos por nombre de cerveza, cervecería y tipo de cerveza

```
[ ] # Utilizando el método groupby() de Pandas, se agrupan los datos del DataFrame beer_reviews  
# La agrupación se realiza por las columnas "beer_name", "brewery_name" y "beer_style"  
beer = beer_reviews.groupby(["beer_name", "brewery_name", "beer_style"])
```

- Se creó un marco de datos con calificaciones promedio basadas en cervezas individuales.

```
[ ] # Se crea un DataFrame beer_df con los valores promedio de las reseñas para cada grupo de cerveza  
beer_df = pd.DataFrame(beer.mean())  
  
# Se restablece el índice del DataFrame beer_df para extraer las columnas "brewery_name" y "beer_style"  
ind_beer_rating_df = beer_df.reset_index(["brewery_name", "beer_style"])
```

beer_name	brewery_name	beer_style	brewery_id	review_overall	review_aroma	review_appearance	review_palate	review_taste	beer_abv	beer_beerid
! (Old Ale)	Närke Kulturbryggeri AB	Old Ale	10902.0	4.000000	4.5	4.00	4.000000	4.5	8.2	57645.0
"100"	The Covey Restaurant & Brewery	Belgian Dark Ale	13338.0	4.000000	4.0	4.00	4.000000	4.4	10.0	49236.0
"100" Pale Ale	Aviator Brewing Company	American IPA	18635.0	4.000000	4.0	3.50	4.000000	4.0	6.6	50696.0
"12" Belgian Golden Strong Ale	Flossmoor Station Restaurant & Brewery	Belgian Strong Pale Ale	612.0	4.000000	3.5	4.25	4.750000	4.5	9.0	44353.0
"33" Export	Baltika-Baku	Dortmunder / Export Lager	13640.0	3.000000	3.0	3.00	3.166667	3.0	4.8	30090.0
...
Žilavský Kanec Imperial Stout	Minipivovar Ožambersk	Russian Imperial Stout	18230.0	3.750000	4.5	4.50	4.250000	4.0	5.5	45214.0
Ω-naught (Omeganought)	Three Floyds Brewing Co. & Brewpub	American IPA	26.0	4.166667	4.0	4.00	3.500000	3.5	5.0	49427.0
"Chewy" Double IPA	The Bobcat Café & Brewery	American IPA	7302.0	2.500000	2.0	4.00	3.500000	2.5	6.9	71197.0
横濱ビール (Yokosuka Biaisake)	Kamakura Beer Brewing Company	English Bitter	4973.0	4.000000	3.5	3.00	3.500000	3.5	5.5	70370.0
豊山ビール (Hayama Beer)	Kamakura Beer Brewing Company	English Bitter	4973.0	4.000000	3.0	3.00	3.500000	4.0	5.5	70327.0

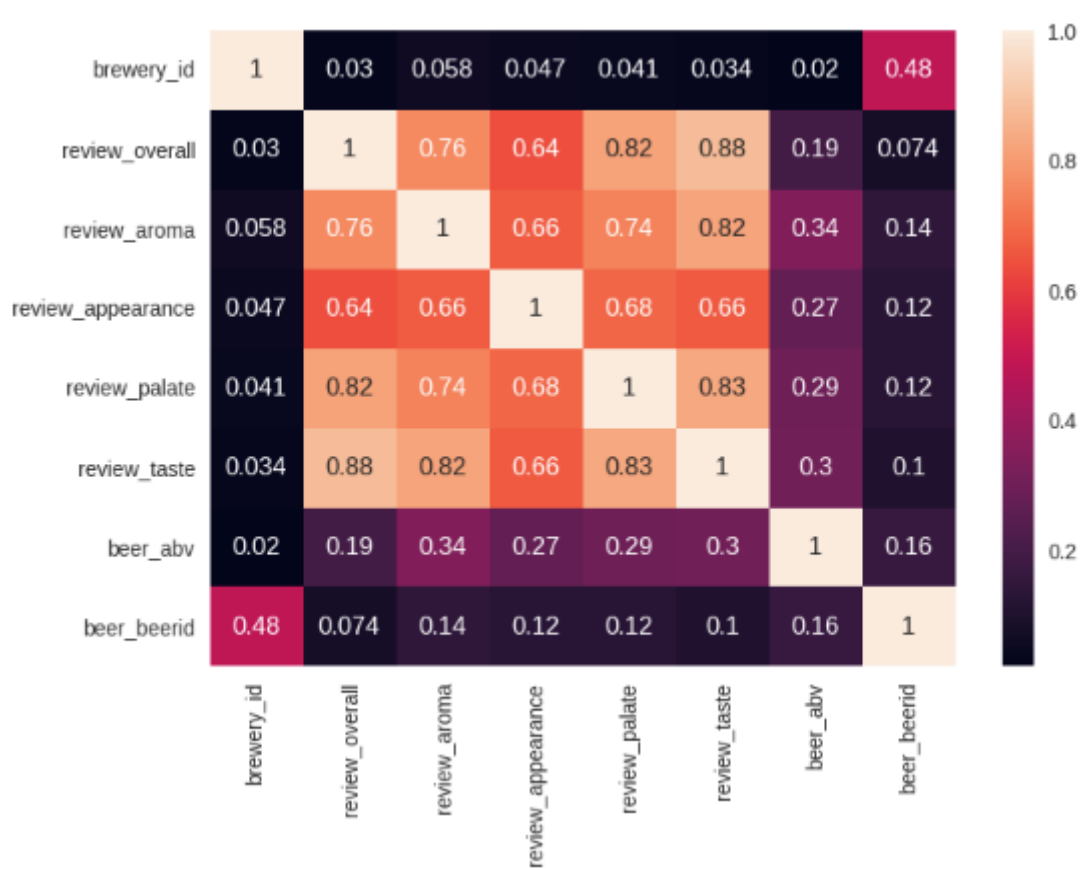
48832 rows x 10 columns

- Se visualizó la matriz de correlación para verificar las características relacionadas.

Creación de matriz de correlación para verificar las características relacionadas

```
# Se calcula la matriz de correlación para el DataFrame ind_beer_rating_df
corr_matrix = ind_beer_rating_df.corr()

# Se utiliza Seaborn para crear un mapa de calor con la matriz de correlación
# El parámetro annot=True agrega los valores de correlación en cada celda del mapa de calor
sns.heatmap(corr_matrix, annot=True)
```



Por ejemplo podemos decir que hay una fuerte relacion entre el nombre de la cerveza y la cerveceria. Como asi tambien el Paladar con la puntuacion final.

- Se seleccionaron características de interés.

Selección de características de interés

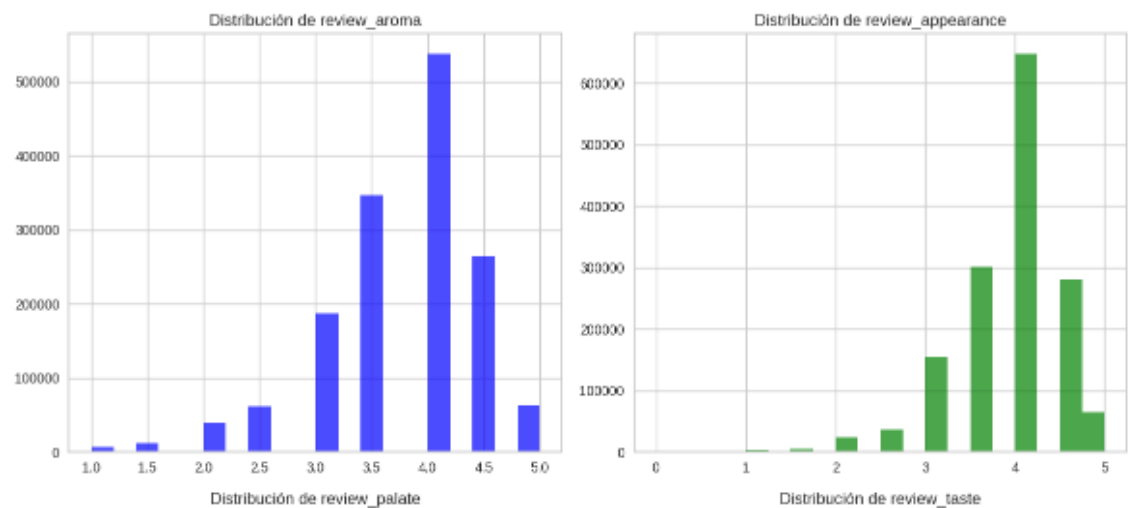
```
] # Se crea una lista llamada selected_features que contiene los nombres de las características seleccionadas  
selected_features = ["review_aroma", "review_appearance", "review_palate", "review_taste", "beer_abv"]
```

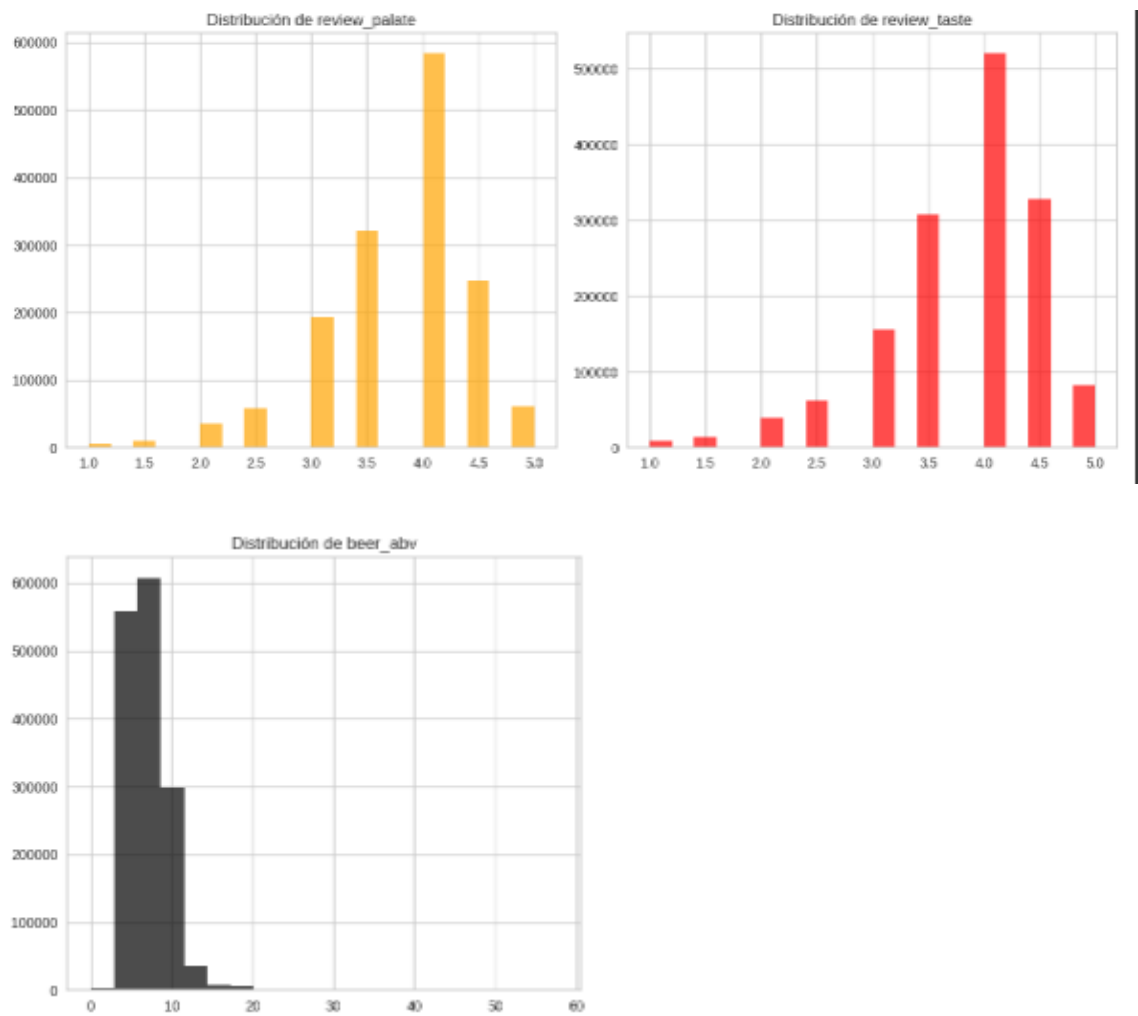
- Se realiza graficos de histogramas de las características seleccionadas

Histograma y definición colores para cada característica

```
# Define colores para cada característica  
colors = ['blue', 'green', 'orange', 'red', 'black']  
  
# Crear subgráficos  
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(12, 15))  
  
# Iterar sobre las características y colores  
for i, (col, color) in enumerate(zip(['review_aroma', 'review_appearance', 'review_palate', 'review_taste', 'beer_abv'], colors)):  
    beer_reviews[col].hist(bins=20, ax=axes[i // 2, i % 2], color=color, alpha=0.7)  
    axes[i // 2, i % 2].set_title(f'Distribución de {col}')  
  
# Ajustar el diseño y mostrar  
plt.tight_layout()  
plt.suptitle('Distribución de Variables', y=1.02) # Ajustar la posición del título  
plt.show()
```

Distribución de Variables





Clustering:

- Se determinó el número óptimo de clusters utilizando el método de codo.

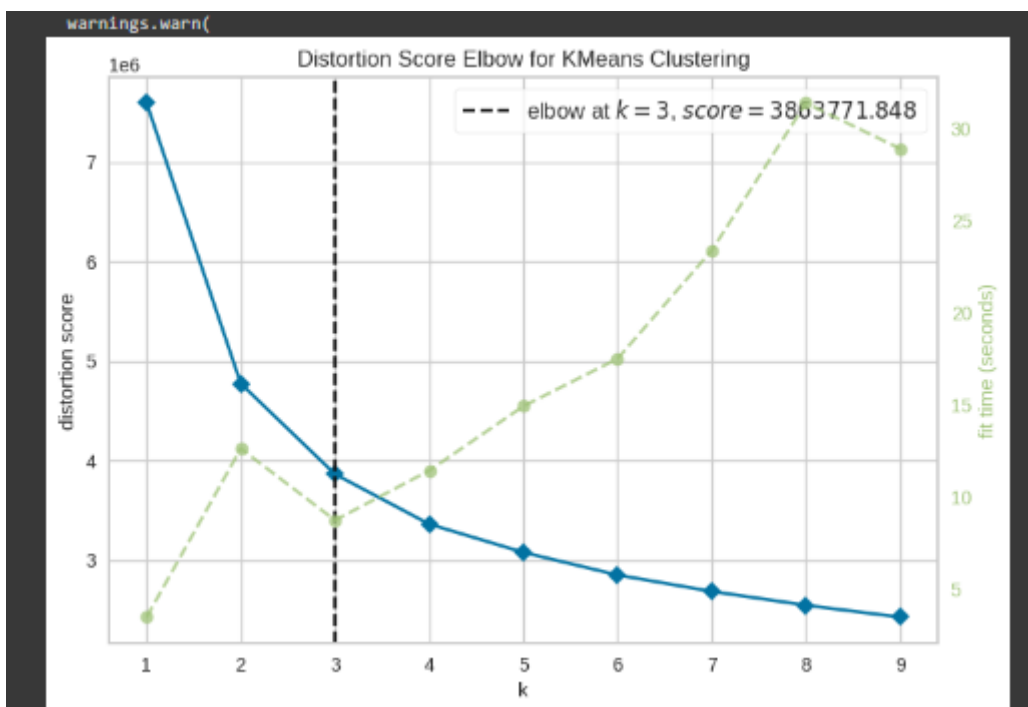
Determinación del número óptimo de clusters (k)

```
] # Selecciona las características de interés del DataFrame beer_reviews
selected_data = beer_reviews[selected_features]

# Estandariza los datos utilizando StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(beer_reviews[selected_features])

# Utiliza el visualizador KElbowVisualizer para determinar el número óptimo de clusters (k)
# El rango de k se establece de 1 a 10, pero puedes ajustarlo según sea necesario
visualizer = KElbowVisualizer(KMeans(), k=(1, 10))
visualizer.fit(scaled_data)

# Muestra el gráfico que ayuda a determinar el número óptimo de clusters
visualizer.show()
```



✓ Selección del número óptimo de clusters (k)

```
[ ] # Se obtiene el número óptimo de clusters determinado por el visualizador KElbowVisualizer
num_clusters_optimal = visualizer.elbow_value_

# Muestra el número óptimo de clusters
print("el numero optimo de cluster, es el numero:", num_clusters_optimal)
```

- Se aplicó el algoritmo K-Means con el número óptimo de clusters.

```
# Copiamos el DataFrame original
data_beer_cluster = beer_reviews.copy()

# Aplica K-means con el número óptimo de clusters
kmeans = KMeans(n_clusters=num_clusters_optimal, random_state=42)
data_beer_cluster["cluster"] = kmeans.fit_predict(scaled_data)
```

- Se realizó un análisis de resultados, generando boxplots para cada cluster y cada característica seleccionada.

Análisis de Resultados: Genera boxplots para cada cluster y cada característica seleccionada.

```
# Análisis de Resultados: Generación de boxplots

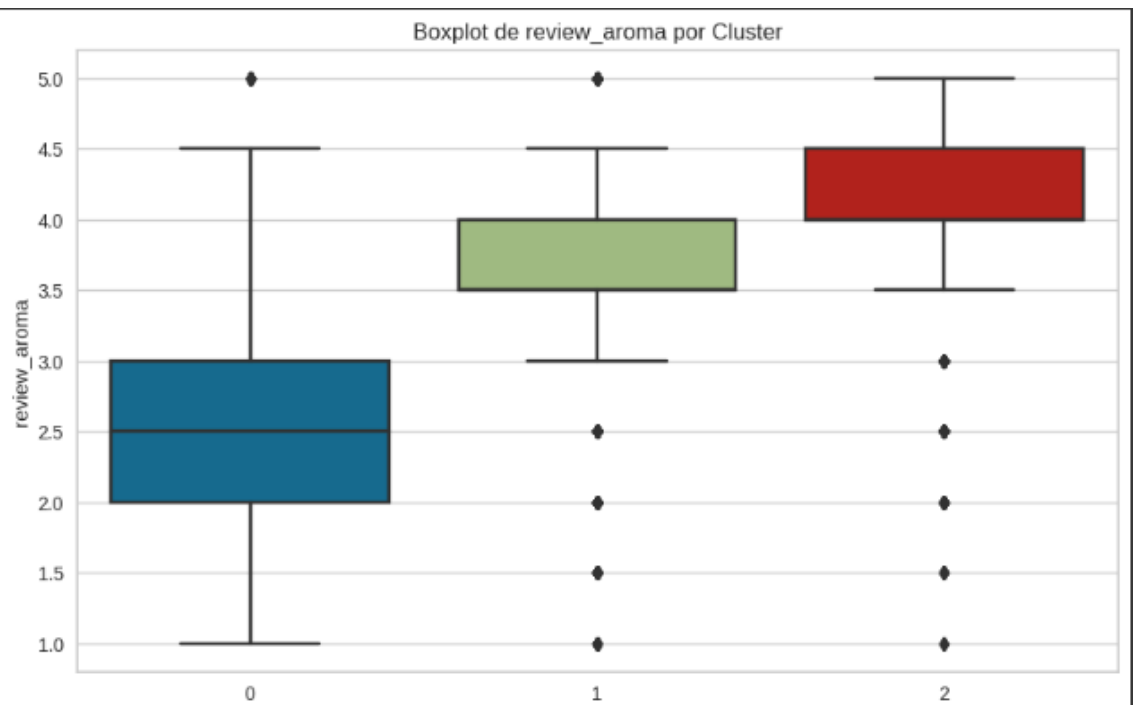
import seaborn as sns
import matplotlib.pyplot as plt

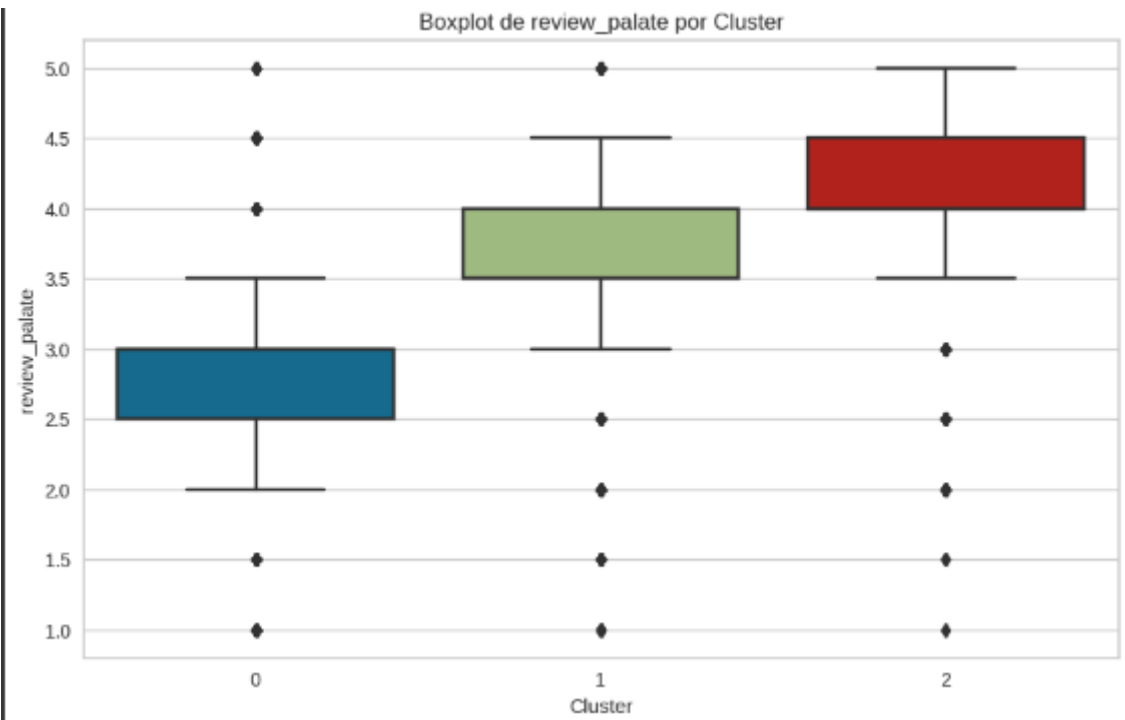
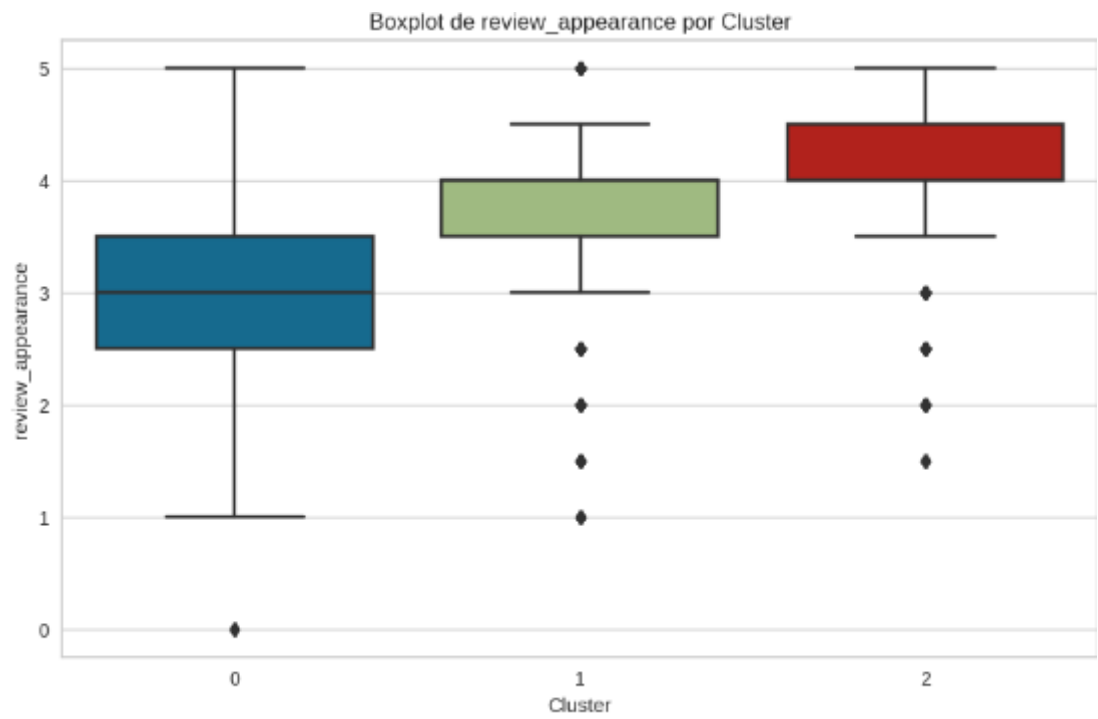
# Iterar sobre las características seleccionadas
for feature in selected_features:
    plt.figure(figsize=(10, 6))

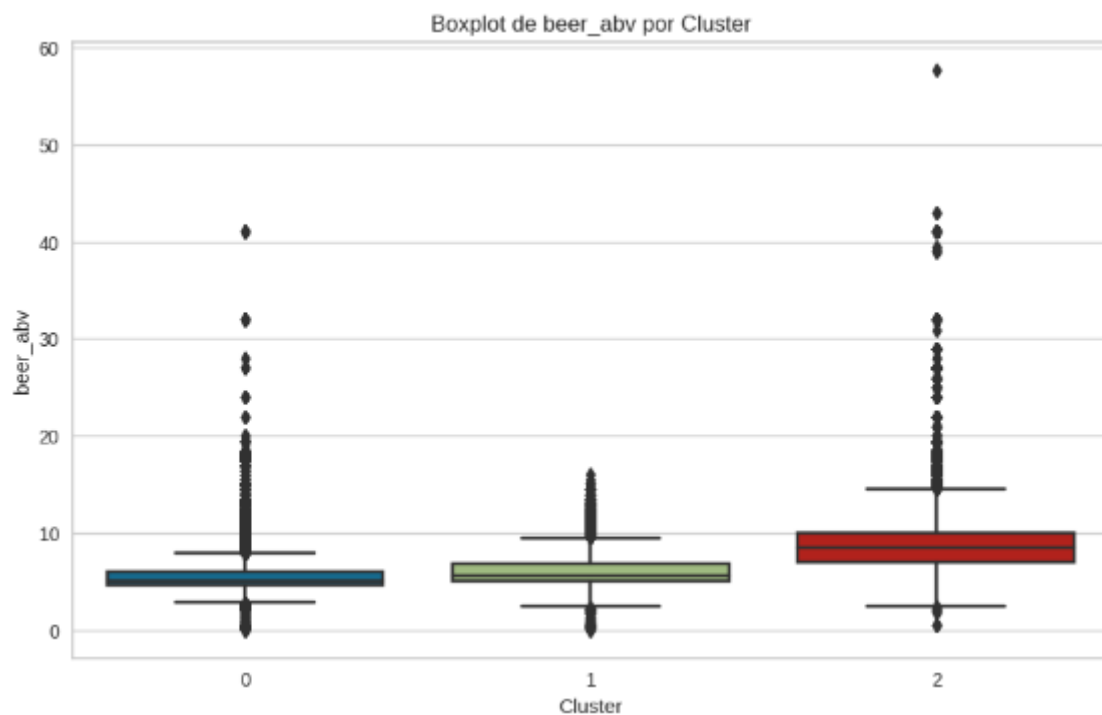
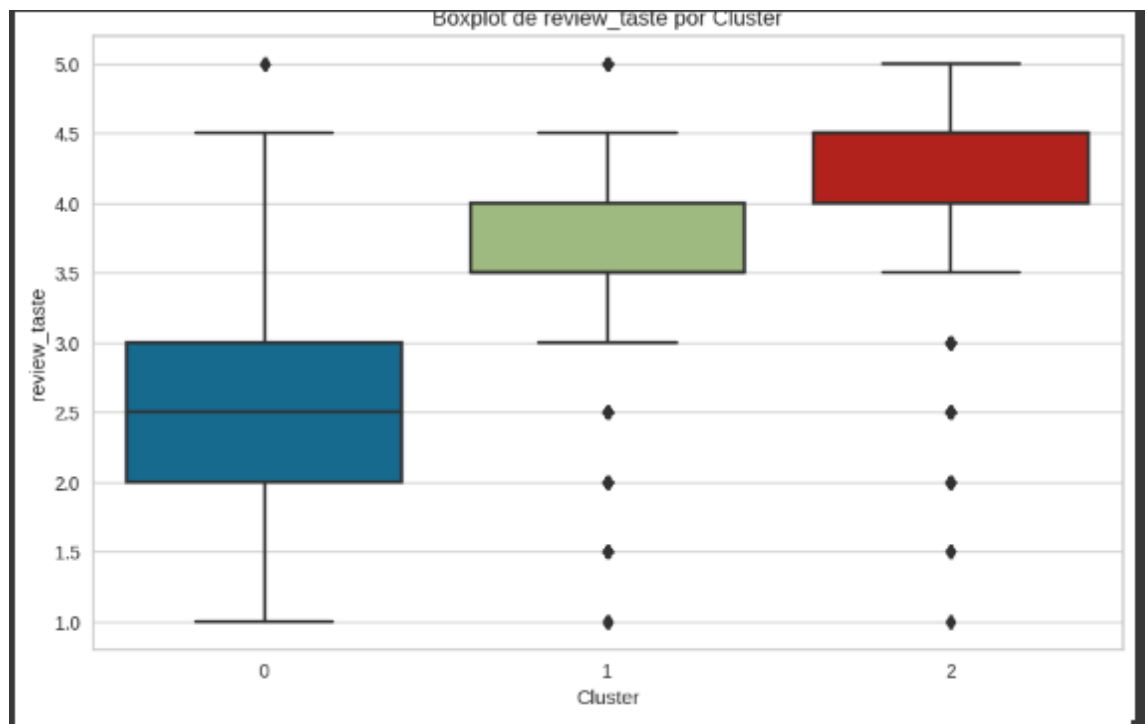
    # Utilizar Seaborn para crear un boxplot para cada cluster
    sns.boxplot(x="cluster", y=feature, data=data_beer_cluster)

    # Configuración de etiquetas y título
    plt.title(f"Boxplot de {feature} por Cluster")
    plt.xlabel("Cluster")
    plt.ylabel(feature)

    # Mostrar el boxplot
    plt.show()
```







Basándonos en la interpretación de los boxplots generados para cada cluster y cada característica seleccionada, podríamos hacer observaciones tentativas sobre la posible interpretación de los clusters. Sin embargo, la interpretación exacta puede depender de la distribución específica de los datos y de la naturaleza de las características seleccionadas.

En general, si observas que el Cluster 0 tiene, por ejemplo, valores más bajos en las características de puntuación en comparación con los Clusters 1 y 2, podrías interpretar que el Cluster 0 representa cervezas con puntajes más bajos. Del mismo modo, si el Cluster 2 tiene valores más altos, podrías interpretar que representa cervezas con puntajes más altos. El Cluster 1, al tener valores intermedios, podría representar cervezas con puntajes medianos.

Hemos definido condiciones basadas en los valores de la columna 'cluster' y has asignado etiquetas específicas ('Menor Puntuación', 'Puntuación Mediana', 'Mejor Puntuación') según el cluster al que pertenece cada observación. Estas etiquetas se han agregado como una nueva columna llamada 'Puntuación'.

```
# Definir las condiciones y los valores a asignar
conditions = [
    data_beer_cluster['cluster'] == 0,
    data_beer_cluster['cluster'] == 1,
    data_beer_cluster['cluster'] == 2,
]

values = ["Menor Puntuación", "Puntuación Mediana", "Mejor Puntuación"]

# Agregar la columna 'Puntuación' basándose en las condiciones
data_beer_cluster['Puntuación'] = np.select(conditions, values, default="Otra Puntuación")
```

Interfaz de Usuario Interactiva:

- Se creó una interfaz interactiva con widgets para permitir a los usuarios ajustar las puntuaciones de características.

```
Definir características seleccionadas para la interfaz

# Definir características seleccionadas para la interfaz
selected_features = ['review_aroma', 'review_appearance', 'review_palate', 'review_taste', 'beer_abv']

# Crear diccionario de deslizadores para cada característica
feature_sliders = {feature: widgets.FloatSlider(
    value=3.0,
    min=1.0,
    max=5.0,
    step=0.1,
    description=f'{feature}: ',
    style={'description_width': 'initial'} # Ajustar el ancho de la descripción
    for feature in selected_features}
```

Crear Boton para mostrar recomendaciones

```
# Crear botón para mostrar recomendaciones
button = widgets.Button(description="Mostrar Recomendaciones")

# Función que se ejecuta al hacer clic en el botón
def on_button_click(b):
    # Obtener las puntuaciones seleccionadas por el usuario
    selected_scores = {feature: slider.value for feature, slider in feature_sliders.items()}

    # Asignar puntajes directamente a los clusters
    data_beer_cluster["score"] = data_beer_cluster["cluster"].map({
        i: selected_scores for i in range(num_clusters_optimal)
    })

    # Filtrar las cervezas según las puntuaciones seleccionadas por el usuario
    selected_beers = data_beer_cluster.copy()
    for feature, score in selected_scores.items():
        selected_beers = selected_beers[selected_beers[feature] >= score]

    # Mostrar el listado de cervezas recomendadas
    recommended_beers = selected_beers[['beer_name', 'brewery_name', 'score']]
    print("\nListado de cervezas recomendadas:")
    print(recommended_beers)

    # Visualización de las puntuaciones seleccionadas
    plt.figure(figsize=(10, 6))
    sns.barplot(x=list(selected_scores.keys()), y=list(selected_scores.values()), color='blue')
    plt.title('Puntuaciones Seleccionadas')
    plt.show()

# Asociar la función on_button_click al evento de hacer clic en el botón
button.on_click(on_button_click)

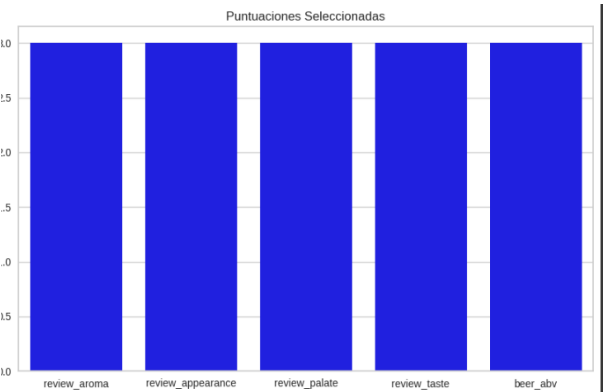
# Crear una lista de widgets para mostrar
widget_list = list(feature_sliders.values()) + [button]
```



review_aroma: 3.00
review_appearance: 3.00
review_palate: 3.00
review_taste: 3.00
beer_abv: 3.00
Mostrar Recomend...

```
Listado de cervezas recomendadas:
beer_name      brewery_name \
4      Cauldron DIPA      Caldera Brewing Company
5      Caldera Ginger Beer      Caldera Brewing Company
6      Caldera Ginger Beer      Caldera Brewing Company
8      Caldera Ginger Beer      Caldera Brewing Company
9      Caldera Ginger Beer      Caldera Brewing Company
...
1586608 The Horseman's Ale      The Defiant Brewing Company
1586609 The Horseman's Ale      The Defiant Brewing Company
1586611 The Horseman's Ale      The Defiant Brewing Company
1586612 The Horseman's Ale      The Defiant Brewing Company
1586613 The Horseman's Ale      The Defiant Brewing Company

score
4      {'review_aroma': 3.0, 'review_appearance': 3.0...
5      {'review_aroma': 3.0, 'review_appearance': 3.0...
6      {'review_aroma': 3.0, 'review_appearance': 3.0...
8      {'review_aroma': 3.0, 'review_appearance': 3.0...
9      {'review_aroma': 3.0, 'review_appearance': 3.0...
...
1586608 {'review_aroma': 3.0, 'review_appearance': 3.0...
1586609 {'review_aroma': 3.0, 'review_appearance': 3.0...
1586611 {'review_aroma': 3.0, 'review_appearance': 3.0...
1586612 {'review_aroma': 3.0, 'review_appearance': 3.0...
1586613 {'review_aroma': 3.0, 'review_appearance': 3.0...
```



- Se implementó la función para recomendar cervezas según las puntuaciones seleccionadas.

Creamos una función para recomendar cervezas

```
# Función para recomendar cervezas
def recomendar_cervezas(cluster):
    """
    Recomienda las mejores cervezas dentro de un cluster específico.

    Parámetros:
    - cluster: Número de cluster para el cual se quieren recomendar cervezas.

    Devuelve:
    - DataFrame con las mejores cervezas del cluster especificado.
    """
    # Filtrar datos para el cluster especificado
    cluster_data = data_beer_cluster[data_beer_cluster["cluster"] == cluster]

    # Ordenar cervezas por review_overall de forma descendente y tomar las primeras 5
    top_beers = cluster_data.sort_values("review_overall", ascending=False).head(5)

    # Seleccionar columnas relevantes para la recomendación
    recommendation = top_beers[["beer_name", "brewery_name", "review_overall", "beer_style"]]

    # Devolver el DataFrame con las mejores cervezas del cluster
    return recommendation
```

Top 5 de Cervezas con más graduación de alcohol:

Se obtuvo y mostró el top 5 de cervezas con mayor graduación de alcohol y su puntuación asociada.

Top 5 Cervezas con Mayor Graduación de Alcohol y su Puntuación:

	beer_name	brewery_name	beer_style	beer_abv	Puntuación
12919	Schorschbräu Schorschbock 57%	Schorschbräu	Eisbock	57.7	Mejor Puntuación
12939	Schorschbräu Schorschbock 43%	Schorschbräu	Eisbock	43.0	Mejor Puntuación
12940	Schorschbräu Schorschbock 43%	Schorschbräu	Eisbock	43.0	Mejor Puntuación
746310	Sink The Bismarck!	BrewDog	American Double / Imperial IPA	41.0	Mejor Puntuación
746311	Sink The Bismarck!	BrewDog	American Double / Imperial IPA	41.0	Mejor Puntuación

Nos resulta interesante estas cervezas, porque salen de los sabores y de la graduación alcohólica que estamos acostumbrados. También tenemos conocimientos, sobre otras cervezas con graduaciones más altas, pero no se ven en nuestros dataset.

A continuación dejamos links por si desea consultar

<https://www.verema.com/blog/cervezas/1399874-diez-cervezas-mas-graduacion-mundo>

<https://labuenacheve.com/blog/las-6-cervezas-con-mas-alcohol-del-mundo/>

4. Resultados

Presentación de Resultados:

A continuación, se presentan los principales resultados obtenidos:

Número Óptimo de Clusters (K):

Utilizando la técnica de codo (Elbow Method), se determinó que el número óptimo de clusters para el conjunto de datos es 3.

Análisis de Clusters:

Se aplicó el algoritmo K-Means con 3 clusters.

Se generaron boxplots para cada cluster y cada característica seleccionada, permitiendo visualizar las diferencias en las preferencias de los usuarios.

Interfaz de Usuario Interactiva:

Se desarrolló dos interfaces interactivas que permiten a los usuarios ajustar las puntuaciones de características y recibir recomendaciones de cervezas personalizadas.

La primera es por si no sabemos un nombre de una cerveza en particular, y la otra por si lo conocemos.

Top 5 de Cervezas con Mayor Graduación de Alcohol:

Se identificaron y presentaron las 5 cervezas con la graduación de alcohol más alta, junto con sus respectivas puntuaciones.

También existen cervezas, cuya graduación de alcohol es muy superior a la media.
reemplazar abajo les mostrare las 5 con con mayor graduacion y su puntuacion

```
26] # Obtener el top 5 de cervezas con mayor graduación de alcohol y su puntuación
top_5_high_abv_beers = data_beer_cluster.nlargest(5, 'beer_abv')

# Mostrar el DataFrame resultante
print("Top 5 Cervezas con Mayor Graduación de Alcohol y su Puntuación:")
print("*****")
top_5_high_abv_beers[['beer_name', 'brewery_name', 'beer_style', 'beer_abv', 'Puntuación']]
```

Top 5 Cervezas con Mayor Graduación de Alcohol y su Puntuación:

	beer_name	brewery_name	beer_style	beer_abv	Puntuación
12919	Schorschbräu Schorschbock 57%	Schorschbräu	Eisbock	57.7	Mejor Puntuación
12939	Schorschbräu Schorschbock 43%	Schorschbräu	Eisbock	43.0	Mejor Puntuación
12940	Schorschbräu Schorschbock 43%	Schorschbräu	Eisbock	43.0	Mejor Puntuación
746310	Sink The Bismarck!	BrewDog	American Double / Imperial IPA	41.0	Mejor Puntuación
746311	Sink The Bismarck!	BrewDog	American Double / Imperial IPA	41.0	Mejor Puntuación

Top 5 de Cervezas IPA con mayor puntuacion:

Se identificaron y presentaron las 5 cervezas IPA con la puntuacion más alta

Las 5 cerveceras y las 5 cervezas IPA que recomendaríamos por obtener mayor puntuación en general, son las siguientes:

```
# Filtrar las cervezas tipo IPA
ipa_beers = data_beer_cluster[data_beer_cluster['beer_style'].str.lower().str.contains('ipa')]

# Agrupar por nombre de cerveza y seleccionar la cerveza con la puntuación más alta en cada grupo
top_ipa_beers = ipa_beers.groupby('beer_name').apply(lambda group: group.nlargest(1, 'review_overall'))

# Restablecer el índice para evitar duplicados
top_ipa_beers = top_ipa_beers.reset_index(drop=True)

# Ordenar por la puntuación 'review_overall' de mayor a menor
top_ipa_beers = top_ipa_beers.sort_values(by='review_overall', ascending=False)

# Mostrar el resultado
top_ipa_beers[['beer_name', 'brewery_name', 'review_overall']].head()
```

	beer_name	brewery_name	review_overall
1421	Furious	Surly Brewing Company	5.0
3382	Size 7	Steel Toe Brewing	5.0
1589	Hardway IPA	Chicago Brewing Company	5.0
1588	Hardcore IPA (2nd Ed. 9.2%)	BrewDog	5.0
1587	Hardcore IPA	BrewDog	5.0

Interpretación:

Los resultados de clustering proporcionan información valiosa sobre las preferencias de los consumidores de cerveza y permiten personalizar las recomendaciones en función de las características deseadas.

La interfaz interactiva brinda a los usuarios la capacidad de explorar y descubrir nuevas cervezas según sus preferencias individuales.

El análisis del top 5 de cervezas con mayor graduación de alcohol muestra opciones fuertes para aquellos que buscan experiencias más intensas.

5. Discusión

Comparación con Objetivos:

Objetivo Cumplido:

Se logró identificar patrones y agrupaciones en las preferencias de los consumidores a través del clustering.

La interfaz interactiva proporciona una herramienta útil para la recomendación personalizada de cervezas.

Desafíos y Limitaciones:

Desafíos:

La variabilidad en las preferencias de los usuarios puede hacer que sea difícil definir clusters claramente distintos.

La calidad de las recomendaciones depende en gran medida de la calidad y cantidad de datos disponibles.

Limitaciones:

La calidad de las recomendaciones está vinculada a la calidad de las revisiones y la disponibilidad de datos.

La interfaz interactiva es una representación simplificada y puede no abordar todas las preferencias posibles.

6. Conclusiones

Reflexiones Finales:

El proyecto demuestra la aplicación efectiva de técnicas de clustering y visualización de datos para entender las preferencias de los consumidores en el contexto de las revisiones de cervezas.

La interfaz interactiva agrega un componente de usabilidad, permitiendo a los usuarios explorar y descubrir cervezas de manera personalizada.

Aplicación de Conocimientos:

Se aplicaron conceptos clave de big data y machine learning, como preprocesamiento de datos, clustering y visualización, para abordar el problema y proporcionar soluciones útiles.

Sugerencias para Futuras Investigaciones:

Explorar enfoques más avanzados de recomendación, como sistemas de recomendación basados en contenido o filtrado colaborativo.

Incorporar más datos demográficos y de contexto para refinar aún más las recomendaciones.

7. Entrega de Código y Documentación

Enlace a Google Colab (Proporcionar el enlace al Google Colab donde se realizó el trabajo). (REEMPLAZAR POR REPOSITORIO GITHUB)

Documentación del Código:

El código está documentado y comentado para facilitar la comprensión y el mantenimiento.

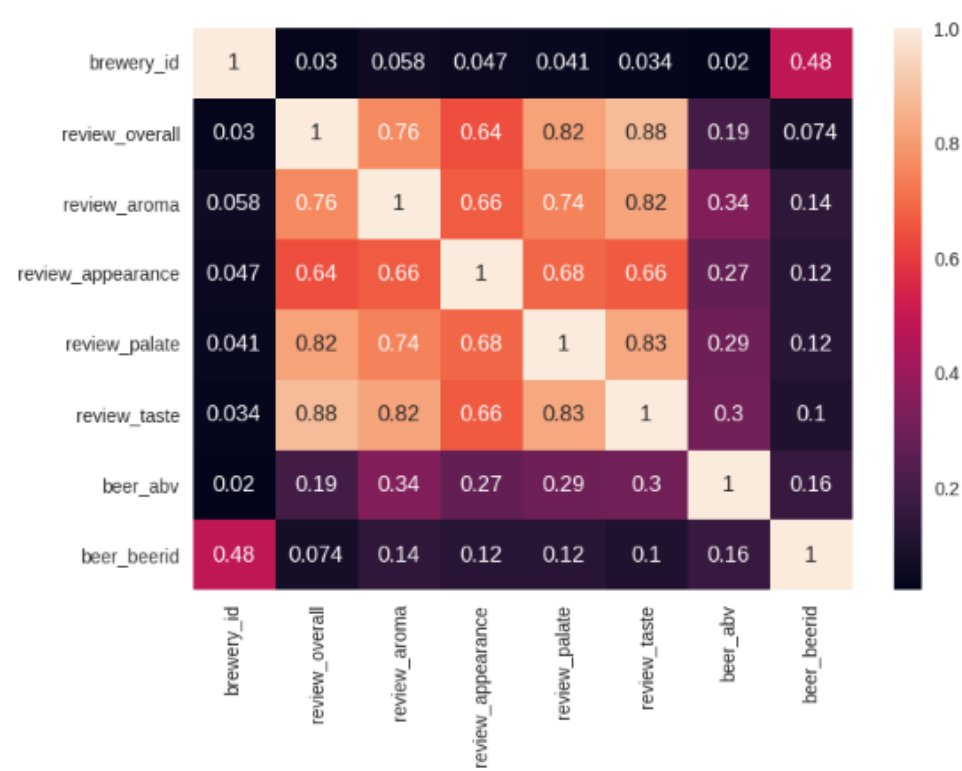
Si tuviera que elegir 5 cervezas para recomendar, ¿cuál elegiría? ¿Por qué?

```
Listado de cervezas recomendadas:
```

	beer_name
443	Caldera IPA
1729	T.J.'s Best Bitter
1768	Wobbly Bob APA
2878	Hop Dam Triple IPA
3146	Mean Manalishi Double I.P.A.

Este listado se obtuvo asignando las máximas puntuaciones de las características seleccionadas, al programa creado.

¿Cuáles de los factores (sabor, aroma, apariencia, paladar) son más importantes para determinar la calidad general de una cerveza?



Con este grafico de matriz de confusionn, podemos ver que paladar, aroma y sabor son los mas influyente en cuanto a factores importantes para determinar la la calidad general de una cerveza

Si por lo general disfruto de las IPA, ¿qué cerveza debo probar?

Las 5 cervcerzas IPA que recomendaríamos por obtener mayor puntuacion en general, son las siguientes:

```
# Filtrar las cervezas tipo IPA
ipa_beers = data_beer_cluster[data_beer_cluster['beer_style'].str.lower().str.contains('ipa')]

# Agrupar por nombre de cerveza y seleccionar la cerveza con la puntuación más alta en cada grupo
top_ipa_beers = ipa_beers.groupby('beer_name').apply(lambda group: group.nlargest(1, 'review_overall'))

# Restablecer el índice para evitar duplicados
top_ipa_beers = top_ipa_beers.reset_index(drop=True)

# Ordenar por la puntuación 'review_overall' de mayor a menor
top_ipa_beers = top_ipa_beers.sort_values(by='review_overall', ascending=False)

# Mostrar el resultado
top_ipa_beers[['beer_name', 'brewery_name', 'review_overall']].head()
```

MATERIA: Elementos de aprendizaje de máquina y Big Data

PROFESOR: Barcos Valentin

GRUPO 5

	beer_name	brewery_name	review_overall
1421	Furious	Surly Brewing Company	5.0
3382	Size 7	Steel Toe Brewing	5.0
1589	Hardway IPA	Chicago Brewing Company	5.0
1588	Hardcore IPA (2nd Ed. 9.2%)	BrewDog	5.0
1587	Hardcore IPA	BrewDog	5.0