Data Source: SKYTRAX

Airlines Analysis on travelers

A Comparative Study of SIA, EVA, ANA, Emirates, and Lufthansa

Presented by:
Data Dynamos
Ho Kwok Leong, Richard, Nabilah

Singapore Airlines (SIA) is Singapore's flag carrier, celebrated for its exceptional service and modern fleet. With a vast global network, including subsidiaries like SilkAir and Scoot, SIA offers premium travel experiences and global connectivity.

Emirates: Leading Dubai-based airline with a modern fleet, serving over 150 destinations globally with luxury amenities and award-winning entertainment.

Lufthansa: Leading German airline with a global network, modern fleet, and premium service to over 220 destinations worldwide.

EVA Air: Leading Taiwanese airline with a modern fleet, serving over 60 international destinations. Renowned for quality service and luxurious cabins, EVA Air offers passengers a premium travel experience.

All Nippon Airways (ANA): Leading Japanese airline with a modern fleet, serving over 100 destinations globally. Renowned for exceptional service and punctuality, ANA offers passengers a premium travel experience.

Driving Actionable Insights

Customer-Centric Focus

- Prioritize initiatives that directly address customer pain points
- Leverage customer feedback

Service Quality

- Identify areas of excellence and opportunities
- Targeted strategies to consistently deliver high-quality services across all aspects of the airline's

Personalization

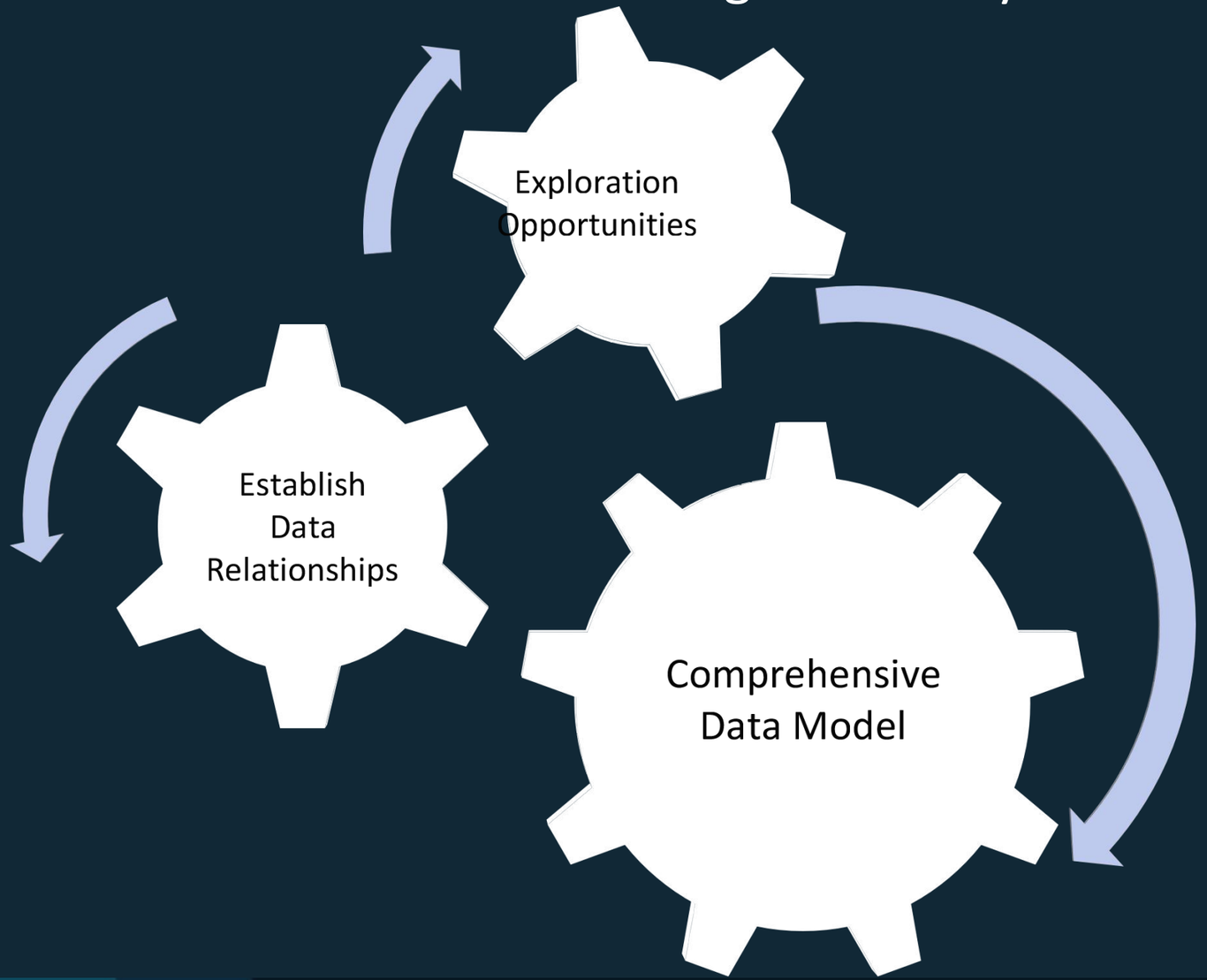
- Segmentation insights to personalize the travel experience
- Cater to the unique needs and preferences of different traveler types and classes.

Innovation

- Embrace a culture of continuous improvement
- Staying ahead of evolving customer expectations and industry trends

Exploring the Data Landscape

What do we aim to achieve through our analysis?



Content


Obtaining a dataset
Transforming the data
Loading into a DB
Market Analysis & Sentiment

After obtaining 5 airlines dataset, Merging all into 1 dataset

```
In [24]: import pandas as pd
         import re

         # Sample DataFrame

         def readData():
             colnames = ['Airline', 'Ratings', 'Snap Reviews', 'Country', 'Name', 'Date Travelled', 'Full Review', 'Aircraft Type',
             SIA =  EVA = pd.read_excel('/Users/nabs/Desktop/Interim Project/sia_excel.xlsx', names = colnames)
             EVA = pd.read_excel('/Users/nabs/Desktop/Interim Project/eva_data.xlsx', names = colnames)
             ANA = pd.read_excel('/Users/nabs/Desktop/Interim Project/ana_excel.xlsx', names = colnames)
             Emirates = pd.read_excel('/Users/nabs/Desktop/Interim Project/emirates_data.xlsx', names = colnames)
             Lufthansa = pd.read_excel('/Users/nabs/Desktop/Interim Project/luthansa_data.xlsx', names = colnames)
             df = pd.concat([SIA, EVA, ANA, Emirates, Lufthansa])

             return df
```

Cleaning up the dataset

In [27]:
```
def remove_ordinal(date_str):
    if isinstance(date_str, str):  # Check if the value is a string
        return re.sub(r'\b(\d+)(st|nd|rd|th)\b', r'\1', date_str)
    else:
        return date_str  # Return the value unchanged if it's not a string

df = readData()

# Apply the function to the 'Date Travelled' column
df['Date Travelled'] = df['Date Travelled'].apply(remove_ordinal)

# Replace non-date values with NaN
df['Date Travelled'] = pd.to_datetime(df['Date Travelled'], errors='coerce')

# Print the DataFrame with parsed dates
print(df)
```

```
       Airline Ratings                                Snap Reviews        Country
0          SIA       9                    "Flight was amazing"          Indonesia
1          SIA       3  "seats on this aircraft are dreadful"              Spain
2          SIA      10          "Food was plentiful and tasty"          Singapore
3          SIA      10           "how much food was available"  United Kingdom
4          SIA      10         "service was consistently good"             Brunei
..         ...     ...                                     ...            ...
995  Lufthansa       8       "I would definitely fly LH again"  United States
996  Lufthansa       1        "forced me to pay extra baggage"  United States
997  Lufthansa       7           "Crew attentive and friendly"      Singapore
998  Lufthansa       9          "attendants extremely helpful"  United States
999        NaN       3                    "unacceptable flaws"  United States

                 Name Date Travelled  \
0    Alison Soetantyo     2024-03-01
1       Robert Watson     2024-02-21
2               S Han     2024-02-20
```

In [30]:
```
# Remove double quotes from the 'snap reviews' column
df['Snap Reviews'] = df['Snap Reviews'].str.replace('"', ' ')

# Display the updated DataFrame
df
```

Out[30]:

	Airline	Ratings	Snap Reviews	Country	Name	Date Travelled	Full Review	Aircraft Type	Traveller Type	Class of Travel	Travel Plans	Date Reviewed	Review Count
0	SIA	9	Flight was amazing	Indonesia	Alison Soetantyo	2024-03-01	Flight was amazing. The crew onboard this fl...	Boeing 777-300ER	Solo Leisure	Business Class	Jakarta to Singapore	2023-12-01 00:00:00	NaN
1	SIA	3	seats on this aircraft are dreadful	Spain	Robert Watson	2024-02-21	Booking an emergency exit seat still meant h...	Boeing 737 Max 8	Solo Leisure	Economy Class	Phuket to Singapore	2024-02-01 00:00:00	17 reviews

Cleaning up the dataset

```
In [31]: df['Aircraft Type'] = df['Aircraft Type'].str.replace('Boeing ', 'B')

         # Display the updated DataFrame
         df
```

Out[31]:

	Airline	Ratings	Snap Reviews	Country	Name	Date Travelled	Full Review	Aircraft Type	Traveller Type	Class of Travel	Travel Plans	Date Reviewed	Review Count
0	SIA	9	Flight was amazing	Indonesia	Alison Soetantyo	2024-03-01	Flight was amazing. The crew onboard this fl...	B777-300ER	Solo Leisure	Business Class	Jakarta to Singapore	2023-12-01 00:00:00	NaN

```
In [6]: import re
        def clean(text):
            text = re.sub(r'[\W]+', ' ', text.lower())
            text = text.replace("hadn't", "had not")\
                       .replace("wasn't", "was not")\
                       .replace("didn't", "did not")\
                       .replace("didn t", "did not")\
                       .replace("couldn't", "could not")\
                       .replace("shouldn't", "should not")\
                       .replace("wouldn't", "would not")\
                       .replace("doesn't", "does not")\
                       .replace("aren't", "are not")\
                       .replace("weren't", "were not")\
                       .replace("hasn't", "has not")\
                       .replace("haven't", "have not")\
                       .replace("won't", "will not")\
                       .replace("isn't", "is not")\
                       .replace("aren't", "are not")\
                       .replace("doesn't", "does not")\
                       .replace("haven't", "have not")\
                       .replace("mustn't", "must not")\
                       .replace("shan't", "shall not")\
                       .replace("mightn't", "might not")\
                       .replace("needn't", "need not")\
                       .replace("oughtn't", "ought not")\
                       .replace("ain't", "am not / is not / are not")
            return text
```

Cleaning up the dataset

In [11]:
```
# Print the data types of the columns in the concatenated DataFrame
print("Data types of columns in the concatenated DataFrame:")
print(concatenated_df.dtypes)
```

```
Data types of columns in the concatenated DataFrame:
airline            object
ratings           float64
snap_reviews       object
country            object
name               object
date_travelled     object
full_review        object
aircraft_type      object
traveller_type     object
class_travel       object
travel_plans       object
date_reviewed      object
review_count       object
Unnamed: 13        object
dtype: object
```

In [15]:
```
# Convert date_travelled and date_reviewed to datetime format
concatenated_df['date_travelled'] = pd.to_datetime(concatenated_df['date_travelled'], errors='coerce')
concatenated_df['date_reviewed'] = pd.to_datetime(concatenated_df['date_reviewed'], errors='coerce')


# Now check the data types again
print(concatenated_df.dtypes)
```

```
airline                   object
ratings                  float64
snap_reviews              object
country                   object
name                      object
date_travelled    datetime64[ns]
full_review               object
aircraft_type             object
traveller_type            object
class_travel              object
travel_plans              object
date_reviewed     datetime64[ns]
review_count              object
Unnamed: 13               object
dtype: object
```

Loading the clean data into a database

- Using Python
- Importing relevant libraries
- Creating a table with chosen columns
- Populating data into the created table

In [16]:
```
!pip install sqlalchemy
!pip install psycopg2
```

```
Requirement already satisfied: sqlalchemy in c:\users\keiot\anaconda3\lib\site-packages (2.0.25)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\keiot\anaconda3\lib\site-packages (from sqlalch
emy) (4.9.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\keiot\anaconda3\lib\site-packages (from sqlalchemy) (3.
0.1)
Requirement already satisfied: psycopg2 in c:\users\keiot\anaconda3\lib\site-packages (2.9.9)
```

In [17]:
```
# Import Libraries
import pandas as pd
import numpy as np
import sqlalchemy as db
```

In [18]:
```
# Create Table in PostgreSQL
# Create connection engine

#user postgres, password admin63, database airline01
engine = db.create_engine('postgresql://postgres:admin63@localhost:5432/airline88')
conn = engine.raw_connection()
```

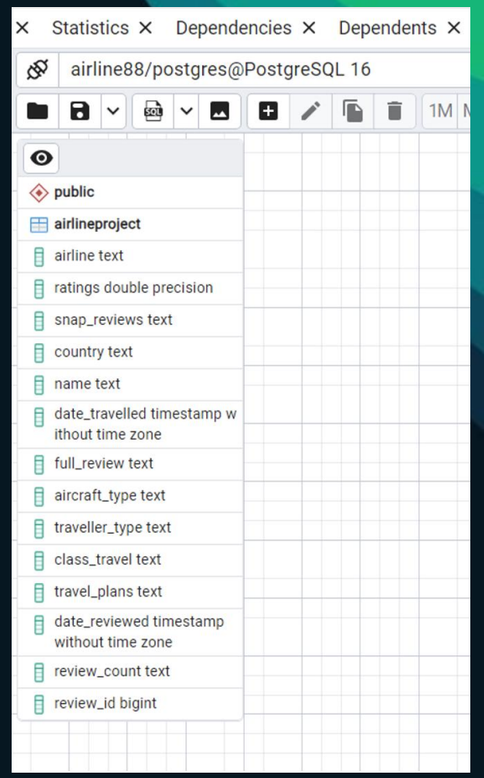
In [19]:
```
# Create new tables in PostgreSQL

commands = (# TABLE :
            ''' CREATE TABLE IF NOT EXISTS airlineproject (
    airline VARCHAR(255),
    ratings FLOAT,
    snap_reviews TEXT,
    country VARCHAR(255),
    name VARCHAR(255),
    date_travelled DATE,
    full_review TEXT,
    aircraft_type VARCHAR(255),
    traveller_type VARCHAR(255),
    class_travel VARCHAR(255),
    travel_plans TEXT,
    date_reviewed DATE,
    review_count INTEGER,
    review_id SERIAL PRIMARY KEY
);''')

# Initialize connection to PostgreSQL
cur = conn.cursor()

# Create cursor to execute SQL commands
#for command in commands:
cur.execute(commands)

# Commit changes
conn.commit()

# Close communication with server
cur.close()
conn.close()
```

In [20]:
```
# Read data from the Excel file into a DataFrame
airlineproject = pd.read_excel('airlineproject.xlsx')

# Now the DataFrame airlineproject is defined and contains the data from the Excel file
%who
```

```
airlineproject   alter_query     commands        conn     create_engine   cur     db      engine  np
pd
```

Loading the clean data into a database

- Establishing our Entity-Relationship Model (ERD)
- Extraction of data from Postgresql




```
[3]: # Extract data from Postgresql
engine = db.create_engine('postgresql://postgres:admin63@localhost:5432/airline88')
conn = engine.raw_connection()

[4]: pd.read_sql('SELECT * FROM airlineproject', conn)
```

```
C:\Users\keiot\AppData\Local\Temp\ipykernel_58692\2592107901.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  pd.read_sql('SELECT * FROM airlineproject', conn)
```

[4]:

	airline	ratings	snap_reviews	country	name	date_travelled	full_review	aircraft_type	traveller_type	class_travel	travel_plans	date_reviewed	review_cc
0	ANA	10.0	"the service cannot be faulted"	(India)	Rajat Garg	2019-06-14	✅ | Delhi to Seattle via Tokyo. Awesome airl...	None	Family Leisure	Economy Class	Delhi to Seattle via Tokyo	2019-06-01	1 rev
1	ANA	10.0	"Simply the best airline"	(France)	W Jewell	2018-12-10	✅ | Frankfurt to Tokyo. Simply the best airli...	Boeing 777	Couple Leisure	Economy Class	Frankfurt to Tokyo	2018-11-01	1 rev

Sentiment Analysis: Uncovering the Voice of the Customer

Sentiment Scoring

- To gauge the overall sentiment expressed in customer reviews
- Ranging from positive to negative
- Identifies areas of strength and opportunities for improvement from the customer's perspective

Thematic Analysis

- Using Natural Language Processing (NLP) we can identify common themes and topics discussed in the review text
- Provides valuable insights which guides targeted improvement efforts

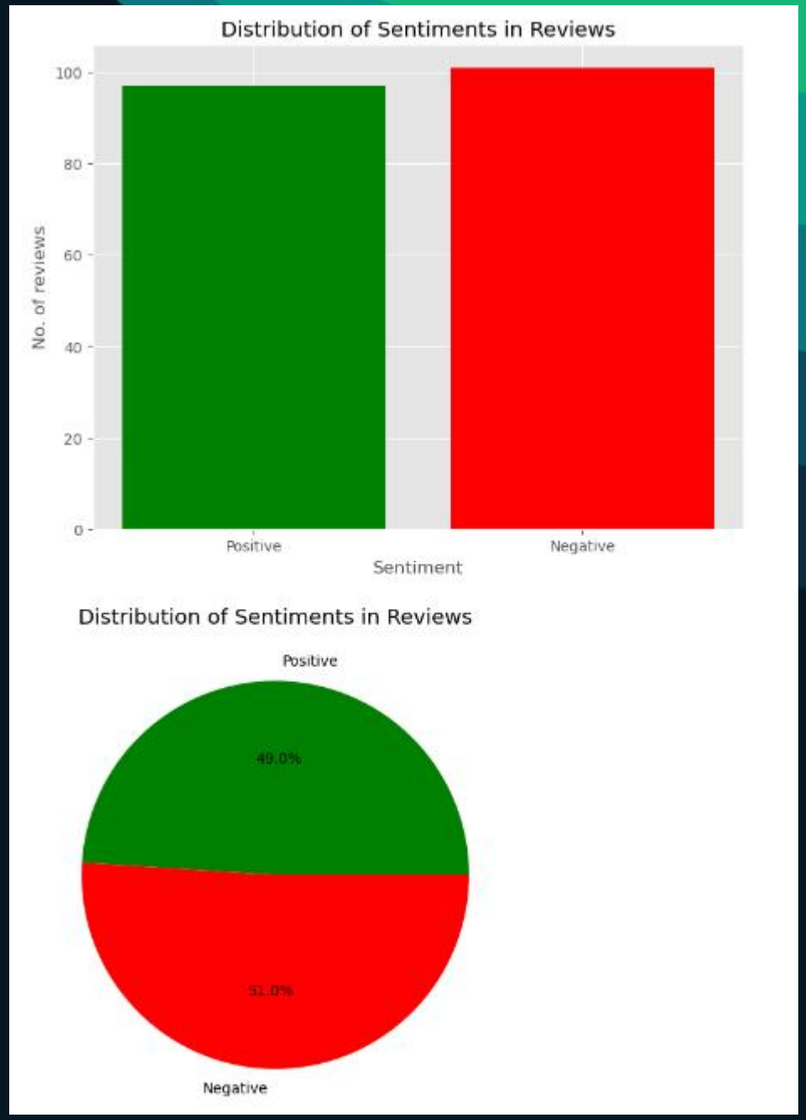
Temporal Trends

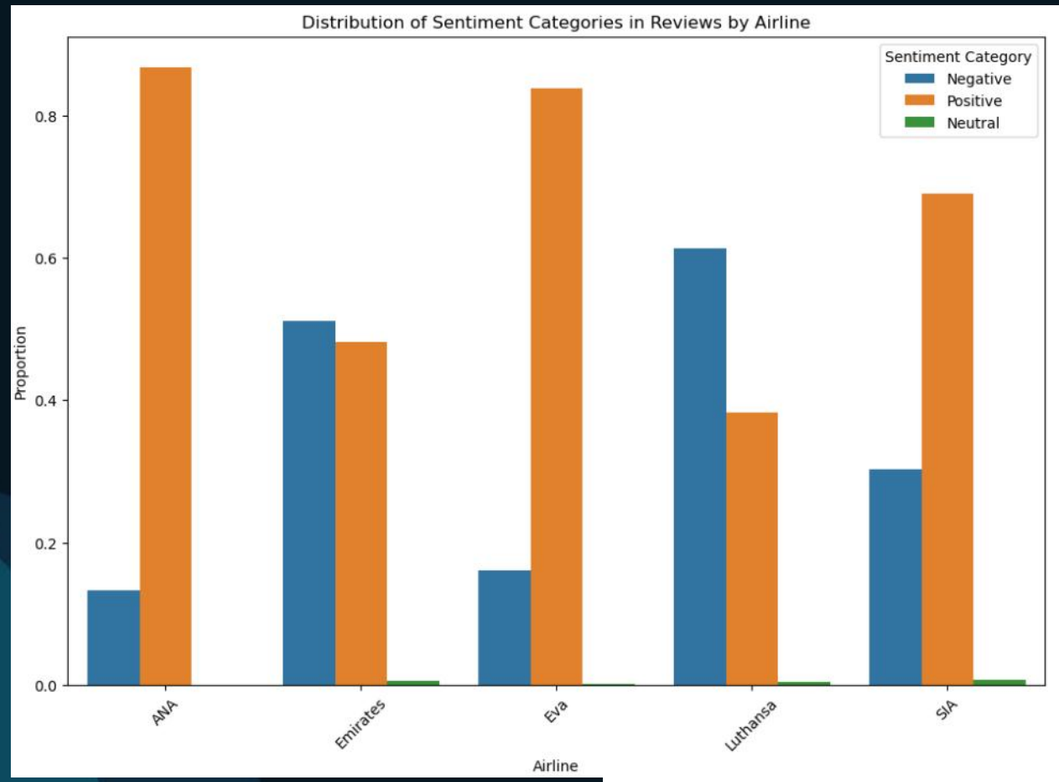
- Analyzing sentiment and thematic trends over time can reveal how customer perceptions evolve
- Improves ability to address emerging concerns
- This dynamic understanding can inform strategic planning and decision-making.

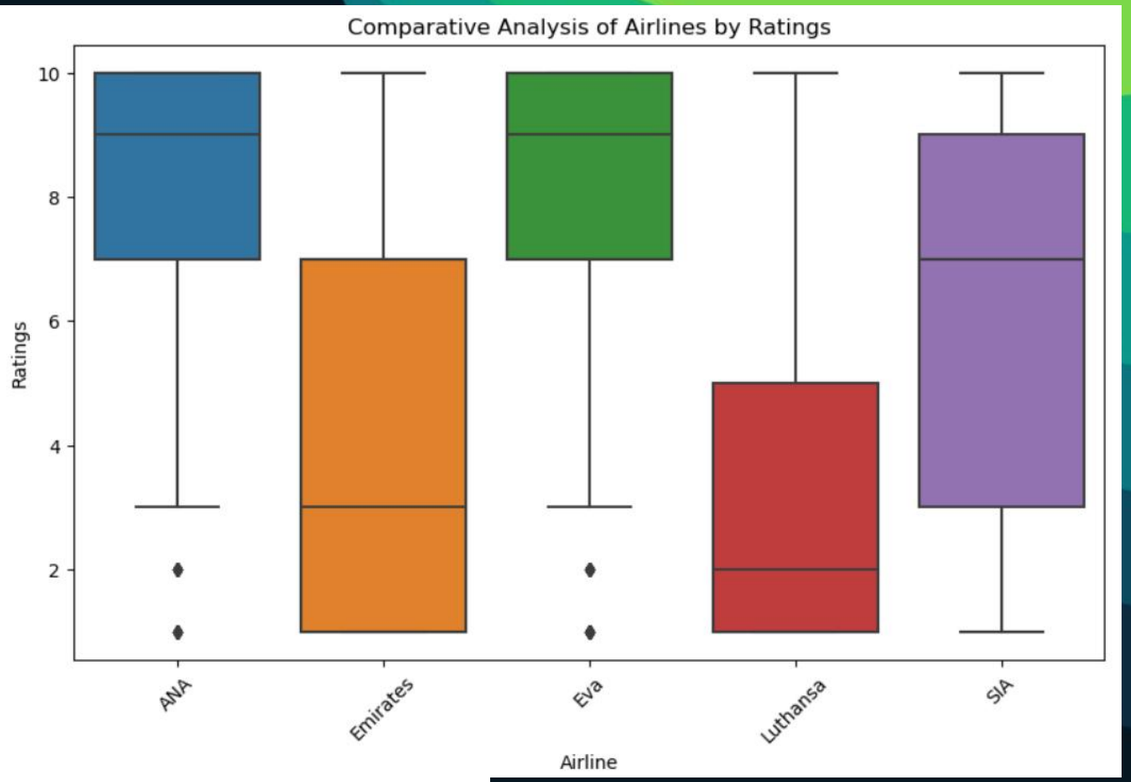
Out[7]:

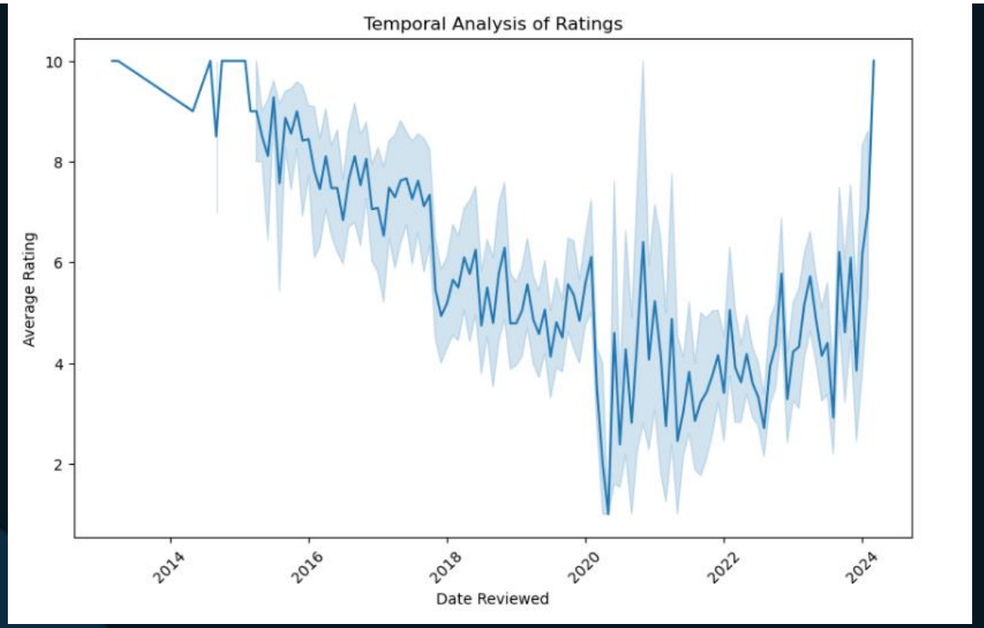
	Review	Sentiment
0	my experience with the ground staff cabin facilities and the cabin crew has been pathetic ground staff at delhi was rude and bossy cabin crew did not take the requests of the passengers choice req...	0.0
1	flight was amazing the crew onboard this flight were very welcoming and gave a good atmosphere the crew serving my aisle goes by the initial g she was very kind helpful gave my mom a bday cake for...	1.0
2	booking an emergency exit seat still meant huge discomfort in a seat far too narrow and poor padding meaning back ache in 90 minute flight the seats on this aircraft are dreadful the headphones an...	0.0
3	excellent performance on all fronts i would definitely choose to use this airline again the aircraft is well maintained and the staff well trained on hospitality food was plentiful and tasty	1.0
4	pretty comfortable flight considering i was flying in economy class in one of the older aircraft in their fleet which is still kept in relatively good condition the economy class cabin was in a 3 ...	1.0
5	the service was consistently good from start to finish the cabin crew showcased the friendly singaporean culture at its finest it seemed like they really cared for the passengers which is what i l...	1.0
6	this flight was over six hours long on a b737 max8 i booked an emergency exit row seat by the window the seat is absolutely dreadful lacks padding and within a few hours developed back ache that l...	0.0
7	boarding process went smoothly and plane left on time i had a window seat in the first part of economy section the plane has 3 4 3 layout we were 3 adults on this row and it was very narrow when t...	0.0
8	pleasant flight which operated on time check in at changi was smooth and assistance given to me since i did not know how to use the check in kiosk and bag drop lunch was provided onboard and advan...	1.0
9	i embarked on a journey with high hopes and anticipation having secured a connecting flight ticket from penang to singapore and then onwards to jakarta opting for singapore airlines renowned for ...	0.0

Sentiment Analysis on Reviews

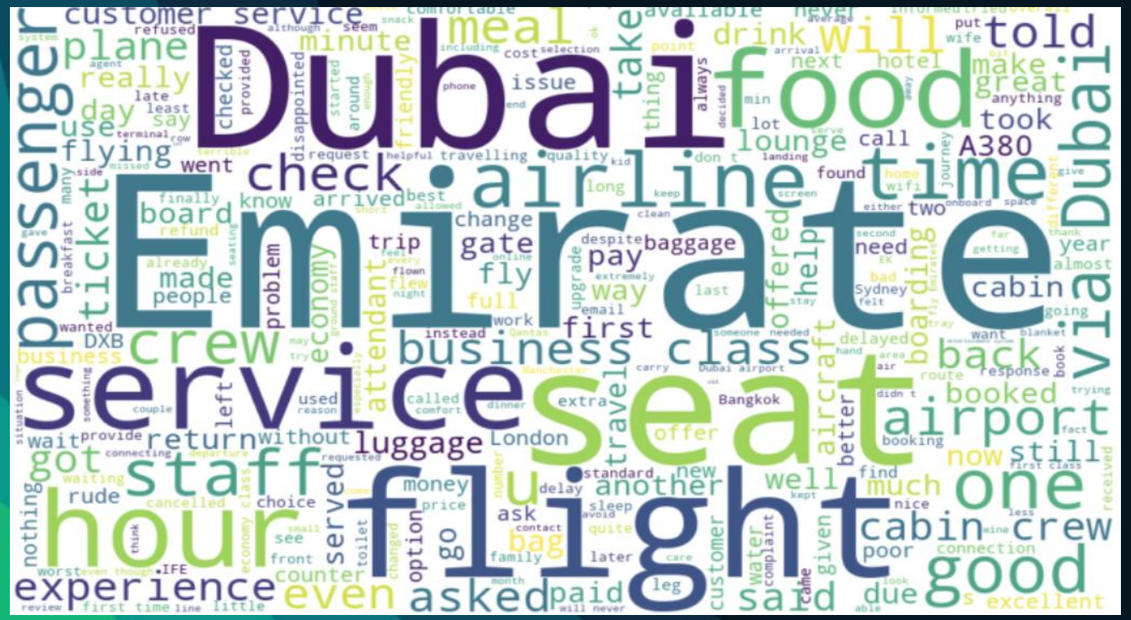



Distribution of Sentiment Categories in Reviews by Airline
Sentiment Category
Negative
Positive
Neutral
Proportion
0.0
0.2
0.4
0.6
0.8
ANA
Emirates
Eva
Lufthansa
SIA
Airline


Comparative Analysis of Airlines by Ratings
Ratings
2
4
6
8
10
ANA
Emirates
Eva
Lufthansa
SIA
Airline


Temporal Analysis of Ratings
Average Rating
2
4
6
8
10
2014
2016
2018
2020
2022
2024
Date Reviewed

```
In [14]: rom wordcloud import WordCloud
         mport matplotlib.pyplot as plt

         topSIA = ['Singapore', 'Airlines', 'SIA', 'flight', 'Flight']
         f_SIA = df[df['airline'] == 'SIA'].copy()  # Make a copy of the DataFrame to avoid modifying the original
         f_SIA.loc[:, 'full_review'] = df_SIA['full_review'].apply(lambda x: " ".join(word for word in x.split() if word not

         Concatenate the text from all rows into a single string
         ext = ' '.join(df_SIA['full_review'])

         Generate word cloud
         c = WordCloud(background_color="white", max_words=300, width=1000, height=600)
         c.generate(text)

         Display the word cloud
         lt.figure(figsize=(12, 6))
         lt.imshow(wc, interpolation='bilinear')
         lt.axis("off")
         lt.show()
```



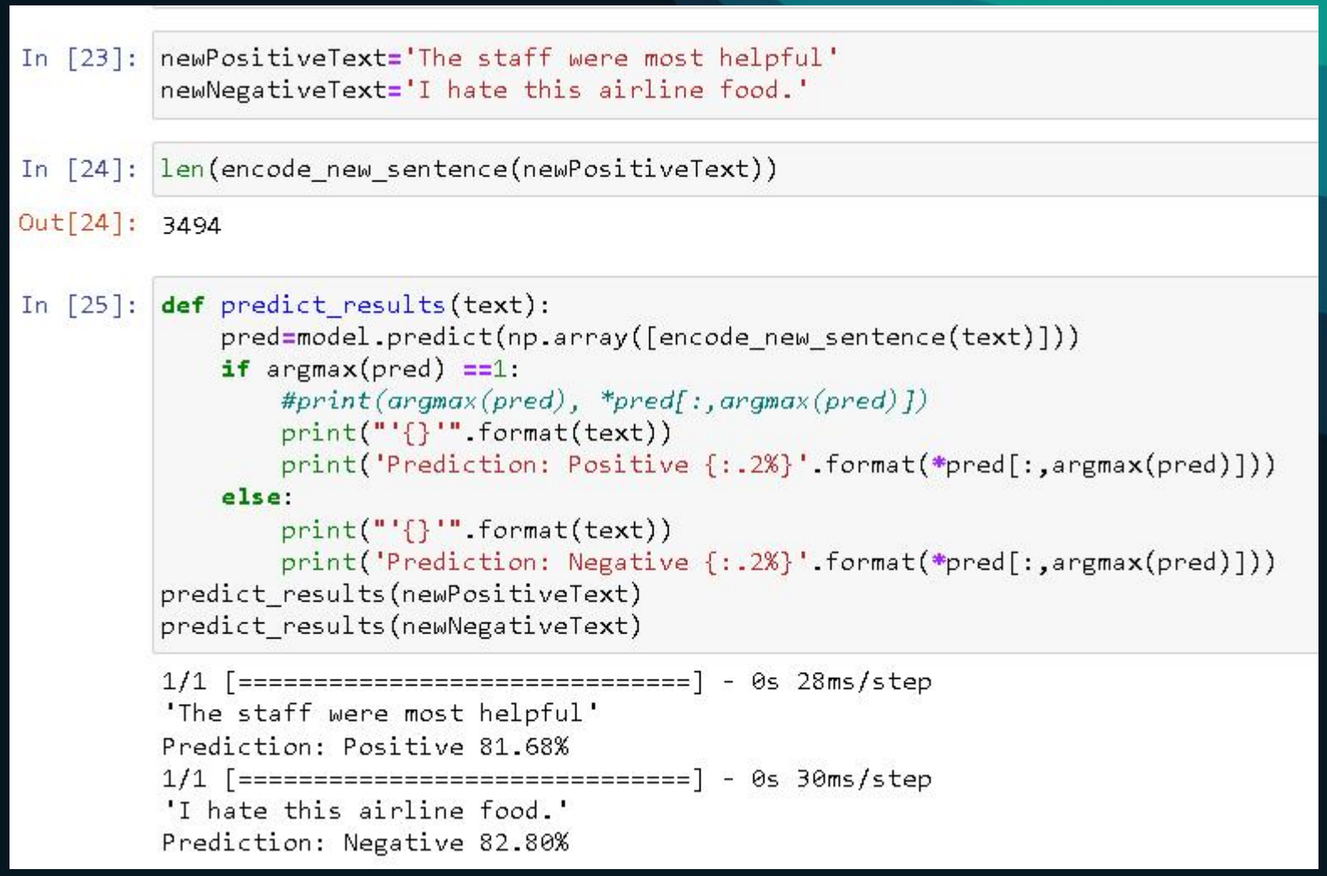
Word Cloud to understand customer trends

Key points

1. Service
2. Seats
3. Food
4. Flight
5. Business Class

Training a model to auto-detect positive or negative feedback from travelers benefits airlines by:

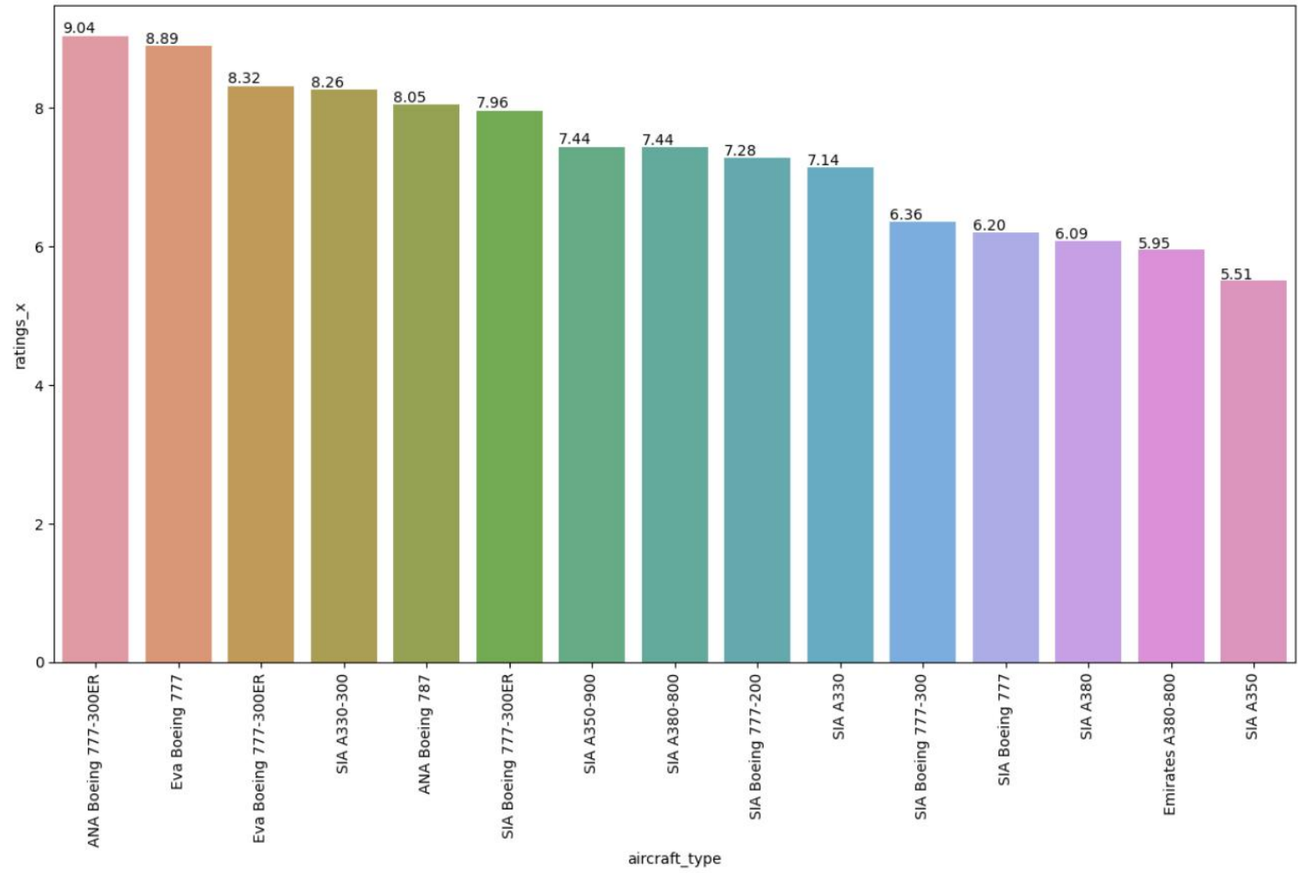
- Improving efficiency and handling large volumes of feedback.
- Providing real-time insights for prompt responses and continuous improvement.
- Enhancing customer experience and satisfaction.
- Supporting data-driven decision-making and personalization.
- Offering a competitive edge and brand reputation management.



```
In [23]: newPositiveText='The staff were most helpful'
         newNegativeText='I hate this airline food.'

In [24]: len(encode_new_sentence(newPositiveText))

Out[24]: 3494

In [25]: def predict_results(text):
             pred=model.predict(np.array([encode_new_sentence(text)]))
             if argmax(pred) ==1:
                 #print(argmax(pred), *pred[:,argmax(pred)])
                 print("'{}'".format(text))
                 print('Prediction: Positive {:.2%}'.format(*pred[:,argmax(pred)]))
             else:
                 print("'{}'".format(text))
                 print('Prediction: Negative {:.2%}'.format(*pred[:,argmax(pred)]))
         predict_results(newPositiveText)
         predict_results(newNegativeText)

         1/1 [==============================] - 0s 28ms/step
         'The staff were most helpful'
         Prediction: Positive 81.68%
         1/1 [==============================] - 0s 30ms/step
         'I hate this airline food.'
         Prediction: Negative 82.80%
```

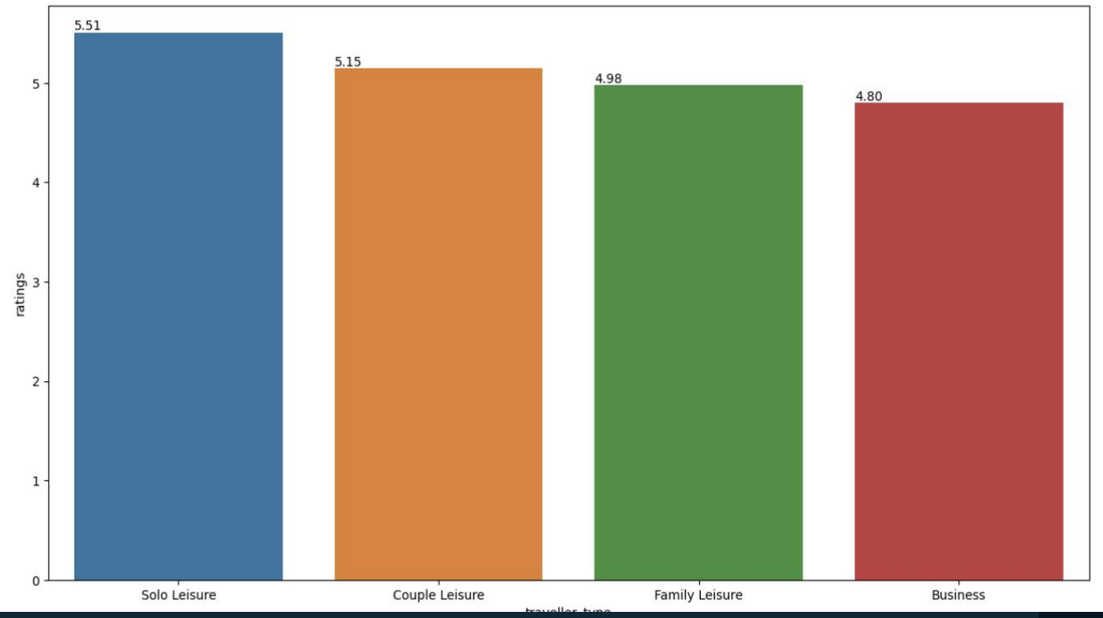
Comparison of the aircraft type vs overall ratings

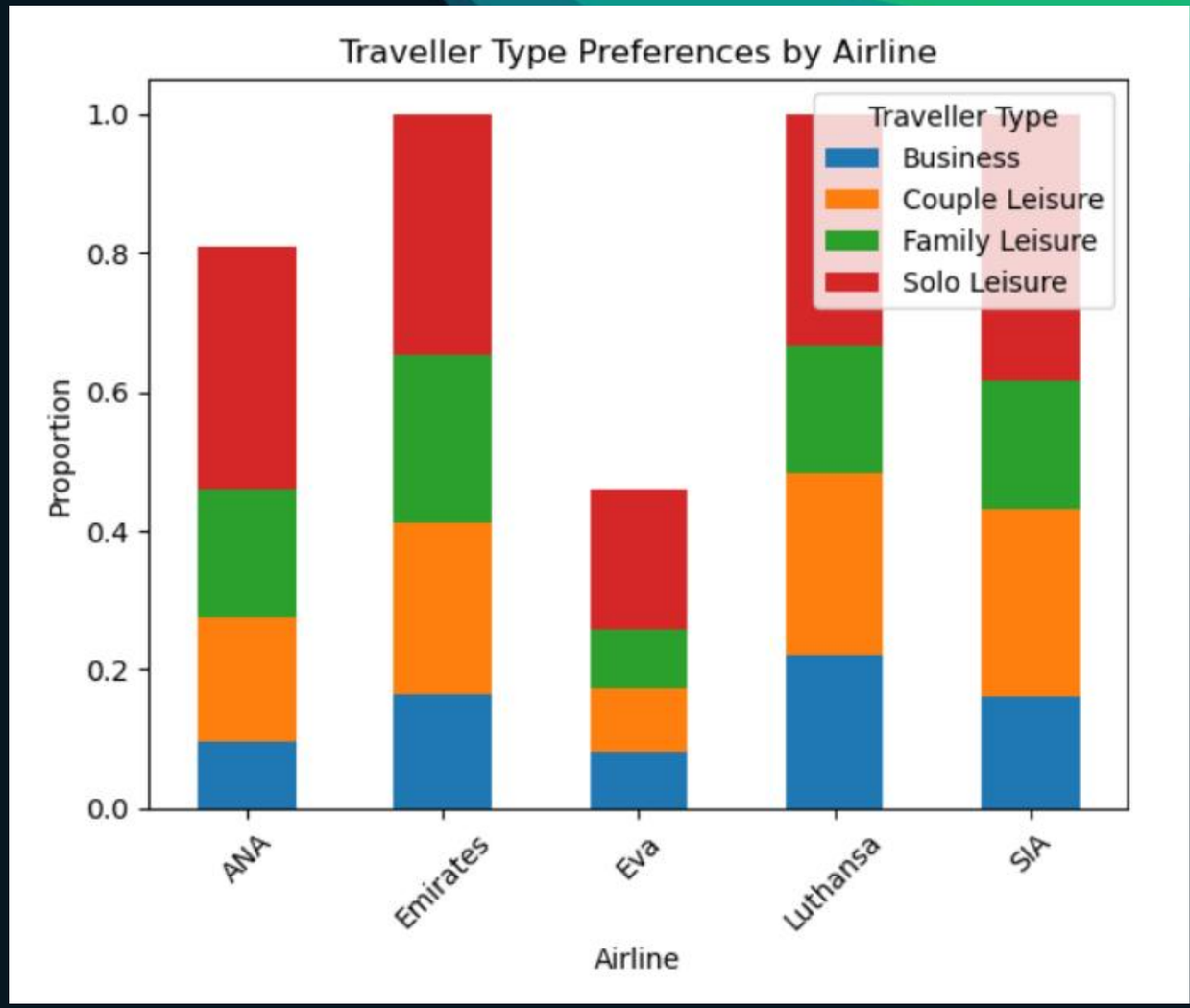
In [7]:
```
dff = df.groupby(['airline','aircraft_type'])['ratings'].mean().reset_index()#.sort_values(by = 'OverallScore', asce
dff1 = df.groupby(['airline','aircraft_type'])['ratings'].count().reset_index()
xx = pd.merge(dff,dff1, on = ['airline','aircraft_type'])
xx = xx[xx['ratings_y'] >=25].sort_values(by = 'ratings_x', ascending = False)
xx['aircraft_type'] = xx['airline'] + [' '] +xx['aircraft_type']
xx = xx.head(15)

import seaborn as sns
fig, ax = plt.subplots(figsize=(15,8))
sns.barplot(x="aircraft_type", y="ratings_x", data=xx, ax=ax)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)
for p in ax.patches:
    ax.annotate("%0.2f"%(p.get_height()), (p.get_x(), p.get_height() * 1.005))
plt.show()
```



Traveler type

In [6]:
```
# Remove non-numeric values from the 'Ratings' column
df['ratings'] = pd.to_numeric(df['ratings'], errors='coerce')

# Drop rows with NaN values in the 'Ratings' column
df = df.dropna(subset=['ratings'])

# Calculate the mean of 'Ratings' grouped by 'Traveller Type'
Seat = df.groupby('traveller_type')['ratings'].mean().reset_index().sort_values(by='ratings', ascending=False)

# Plotting
import seaborn as sns
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(15, 8))
sns.barplot(x="traveller_type", y="ratings", data=Seat, ax=ax)

for p in ax.patches:
    ax.annotate("%0.2f" % (p.get_height()), (p.get_x(), p.get_height() * 1.005))

plt.show()
```





Limitations:

- Data Availability and Accuracy: Getting accurate and up-to-date data can be challenging as airlines often release this information periodically and may not provide detailed breakdowns.

- Customer Sentiment and Perception: Assessing customer satisfaction and perception involves analyzing qualitative data from various sources, such as reviews and surveys. Interpreting this data accurately to understand customers preferences can be challenging.

- Global Events and Crises: The airline industry is vulnerable to global events and crises such as pandemics, natural disasters, terrorist attacks, and economic downturns. And these can affect the accuracy of our analysis.

Conclusion: Unlocking the Power of Customer Feedback

This interim assessment analysis has showcased the immense value that can be extracted from customer feedback data to drive meaningful improvements in the airline industry. By leveraging the power of Python data analysis techniques, we can guide airlines in enhancing service quality, boosting customer satisfaction, and making informed, data-driven decisions.

From exploring the data landscape and conducting analysis to applying sentiment analysis and generating actionable recommendations, this assessment has demonstrated the holistic approach required to harness the true potential of customer feedback. By fostering a data-driven culture and embracing a continuous improvement mindset, airlines can translate these insights into tangible business results and cement their position as industry leaders in customer experience.

As airlines strive to navigate the evolving industry landscape and meet the changing expectations of their customers, this analysis project serves as a blueprint for unlocking the transformative power of data-driven insights. By empowering airlines to listen to the voice of the customer and make strategic, evidence-based decisions, we can collectively elevate the customer experience and drive the industry forward.