"Analyzing Flight Delays: Identifying the Impact of Pilots and Destination Airports on Timely Arrivals"

This analysis seeks to uncover whether flight delays are more likely to be influenced by the assigned pilot or the destination airport, helping to pinpoint the primary cause of disruptions and improve future operational efficiency.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [5]:

```
df = pd.read_csv('/content/drive/MyDrive/busines case dsml/airline_data/Airlin
e Dataset Updated .csv')
```

In [8]:

```
df.head()
```

Out[8]:

	Passenger ID	First Name	Last Name	Gender	Age	Nationality	Airport Name	Airport Country Code	Country Name	Airpo Contine
0	ABVWIg	Edithe	Leggis	Female	62	Japan	Coldfoot Airport	US	United States	NA
1	jkXXAX	Elwood	Catt	Male	62	Nicaragua	Kugluktuk Airport	CA	Canada	NA
2	CdUz2g	Darby	Felgate	Male	67	Russia	Grenoble-Isère Airport	FR	France	E
3	BRS38V	Dominica	Pyle	Female	71	China	Ottawa / Gatineau Airport	CA	Canada	NA
4	9kvTLo	Bay	Pencost	Male	21	China	Gillespie Field	US	United States	NA

In [9]:

```
df.shape
```

Out[9]:

```
(98619, 15)
```

In [10]:

```
# for my analysis , i extract colums of (airport name, airport country code, airort contry name, departure date, arrival airport, pilot name, flight status)
```

In [11]:

```
df_an = df[['Airport Name', 'Airport Country Code', 'Country Name', 'Departure Date', 'Arrival Airport', 'Pilot Name', 'Flight Status' ]].reset_index(drop= True)
df_an.head()
```

Out[11]:

	Airport Name	Airport Country Code	Country Name	Departure Date	Arrival Airport	Pilot Name	Flight Status
0	Coldfoot Airport	US	United States	6/28/2022	CXF	Fransisco Hazeldine	On Time
1	Kugluktuk Airport	CA	Canada	12/26/2022	YCO	Marla Parsonage	On Time
2	Grenoble-Isère Airport	FR	France	1/18/2022	GNB	Rhonda Amber	On Time
3	Ottawa / Gatineau Airport	CA	Canada	9/16/2022	YND	Kacie Commucci	Delayed
4	Gillespie Field	US	United States	2/25/2022	SEE	Ebonee Tree	On Time

In [11]:

```
```

In [12]:

```
df_an.shape
```

Out[12]:

```
(98619, 7)
```

In [13]:

```
df_an.describe()
```

Out[13]:

	Airport Name	Airport Country Code	Country Name	Departure Date	Arrival Airport	Pilot Name	Flight Status
count	98619	98619	98619	98619	98619	98619	98619
unique	9062	235	235	364	9024	98605	3
top	San Pedro Airport	US	United States	7/22/2022	0	Kally Askell	Cancelled
freq	43	22104	22104	325	873	2	32942

In [14]:
```
df_an.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98619 entries, 0 to 98618
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Airport Name          98619 non-null  object
 1   Airport Country Code  98619 non-null  object
 2   Country Name          98619 non-null  object
 3   Departure Date        98619 non-null  object
 4   Arrival Airport       98619 non-null  object
 5   Pilot Name            98619 non-null  object
 6   Flight Status         98619 non-null  object
dtypes: object(7)
memory usage: 5.3+ MB
```

In [15]:
```
# converting Departure Date to day time
```

In [16]:
```
df_an['Departure Date'] = pd.to_datetime(df_an['Departure Date'], format="%d-%m-%Y", errors='coerce')
```

In [17]:
```
df_an.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98619 entries, 0 to 98618
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   Airport Name          98619 non-null  object        
 1   Airport Country Code  98619 non-null  object        
 2   Country Name          98619 non-null  object        
 3   Departure Date        38961 non-null  datetime64[ns]
 4   Arrival Airport       98619 non-null  object        
 5   Pilot Name            98619 non-null  object        
 6   Flight Status         98619 non-null  object        
dtypes: datetime64[ns](1), object(6)
memory usage: 5.3+ MB
```

In [18]:
```
# % percentage of departure date null value
df_an['Departure Date'].isnull().sum()/len(df_an)* 100
```

Out[18]:
```
60.493414048002926
```

when i converted date of object type to date time there are somany null value came to data the % of null value Increased from zero to 60, so the analysis with Departure Date of data type object is more suitable.

In [19]:
```
df_an[['Airport Name', 'Pilot Name']].nunique()
```

Out[19]:

	0
Airport Name	9062
Pilot Name	98605

dtype: int64

In [20]:
```
percentage_unique_pilots  = df_an['Pilot Name'].nunique() / len(df_an) * 100
print(round(percentage_unique_pilots,2))
```

```
99.99
```

Based on the analysis that **99.99%** of pilots in the dataset are unique, it becomes clear that the current data may not provide meaningful insights into whether specific pilots are responsible for flight delays. This high percentage suggests that the airline engages many different pilots across various flights, making it difficult to track patterns or identify frequent contributors to delays.

Recommendation to the airline company: To gain deeper insights into pilot performance and flight delays, it is essential to collect and analyze data on pilots who are repeatedly assigned to flights. Focus on gathering comprehensive data that tracks pilot frequency, workload, and their association with delays. By identifying patterns related to pilots with more frequent assignments, the airline can optimize scheduling, reduce delays, and improve overall operational efficiency. Additionally, incentivizing pilots with good performance records or exploring pilot fatigue management strategies can enhance reliability and reduce delays.

In [20]:
```

```

In [21]:
```
# to analysis with flight delay of airport, i used the column of df_an['Flight Status'] to 'o' for delay and '1' for on time
```

In [22]:
```
df_an["Flight Status"].unique()
```

Out[22]:
```
array(['On Time', 'Delayed', 'Cancelled'], dtype=object)
```

In [23]:

```
# Mapping is used to convert column of "Flight_Status" to numerical value
''' Let " On Time" = 0,
        " Delayed" = 1,
        " Cancelled" = 2
'''
# Mapping flight Status to numerical Value

status_Mapping ={
    'On Time': 0,
    'Delayed': 1,
    'Cancelled':2
}
# Applaying Mapping to Create Numerical Data
df_an['Flight Status_new'] = df_an['Flight Status'].map(status_Mapping)
```

In [24]:

```
df_an.head()
```

Out[24]:

	Airport Name	Airport Country Code	Country Name	Departure Date	Arrival Airport	Pilot Name	Flight Status	Flight Status_new
0	Coldfoot Airport	US	United States	NaT	CXF	Fransisco Hazeldine	On Time	0
1	Kugluktuk Airport	CA	Canada	NaT	YCO	Marla Parsonage	On Time	0
2	Grenoble-Isère Airport	FR	France	NaT	GNB	Rhonda Amber	On Time	0
3	Ottawa / Gatineau Airport	CA	Canada	NaT	YND	Kacie Commucci	Delayed	1
4	Gillespie Field	US	United States	NaT	SEE	Ebonee Tree	On Time	0

In [25]:

```
df_an[['Flight Status_new']].value_counts()
```

Out[25]:

	count
Flight Status_new	
2	32942
0	32846
1	32831

dtype: int64

In [29]:

```
df_new=pd.DataFrame( df_an.groupby(['Airport Name','Country Name'        ])['Flight Status_new'].value_counts().reset_index(name='count'))
```

In [30]:
```
df_new
```

Out[30]:

	Airport Name	Country Name	Flight Status_new	count
0	28 de Noviembre Airport	Argentina	0	7
1	28 de Noviembre Airport	Argentina	1	5
2	28 de Noviembre Airport	Argentina	2	4
3	9 de Maio - Teixeira de Freitas Airport	Brazil	1	4
4	9 de Maio - Teixeira de Freitas Airport	Brazil	0	4
...	...	...	...	...
26569	Žabljak Airport	Montenegro	2	3
26570	Žabljak Airport	Montenegro	1	2
26571	Žilina Airport	Slovakia	2	4
26572	Žilina Airport	Slovakia	0	3
26573	Žilina Airport	Slovakia	1	2

26574 rows × 4 columns

In [32]:
```
df_new=pd.DataFrame( df_an.groupby(['Airport Name','Country Name'        ])['F
light Status_new'].value_counts().unstack(fill_value=0).reset_index())
```

In [33]:
```
df_new
```

Out[33]:

Flight Status_new	Airport Name	Country Name	0	1	2
0	28 de Noviembre Airport	Argentina	7	5	4
1	9 de Maio - Teixeira de Freitas Airport	Brazil	4	4	2
2	A Coruña Airport	Spain	4	2	5
3	A L Mangham Jr. Regional Airport	United States	5	3	3
4	A P Hill AAF (Fort A P Hill) Airport	United States	5	3	4
...	...	...	...	...	...
9102	Şanlıurfa GAP Airport	Turkey	1	3	4
9103	Şırnak Şerafettin Elçi Airport	Turkey	4	3	4
9104	Šiauliai International Airport	Lithuania	4	3	4
9105	Žabljak Airport	Montenegro	7	2	3
9106	Žilina Airport	Slovakia	3	2	4

9107 rows × 5 columns

In [44]:
```
df_new_melted = df_new.melt(id_vars=['Airport Name','Country Name'],  var_name
="Flight Status_new")
```

In [45]:

```
df_new_melted
```

Out[45]:

	Airport Name	Country Name	Flight Status_new	value
0	28 de Noviembre Airport	Argentina	0	7
1	9 de Maio - Teixeira de Freitas Airport	Brazil	0	4
2	A Coruña Airport	Spain	0	4
3	A L Mangham Jr. Regional Airport	United States	0	5
4	A P Hill AAF (Fort A P Hill) Airport	United States	0	5
...	...	...	...	...
27316	Şanlıurfa GAP Airport	Turkey	2	4
27317	Şırnak Şerafettin Elçi Airport	Turkey	2	4
27318	Šiauliai International Airport	Lithuania	2	4
27319	Žabljak Airport	Montenegro	2	3
27320	Žilina Airport	Slovakia	2	4

27321 rows × 4 columns

In [41]:

```
# Create a pivot table to aggregate flight status counts by Airport and Country
pivot_table = df_new_melted.pivot_table(index=[ 'Country Name'], columns='Flight Status_new', aggfunc='size', fill_value=0)
pivot_table
```

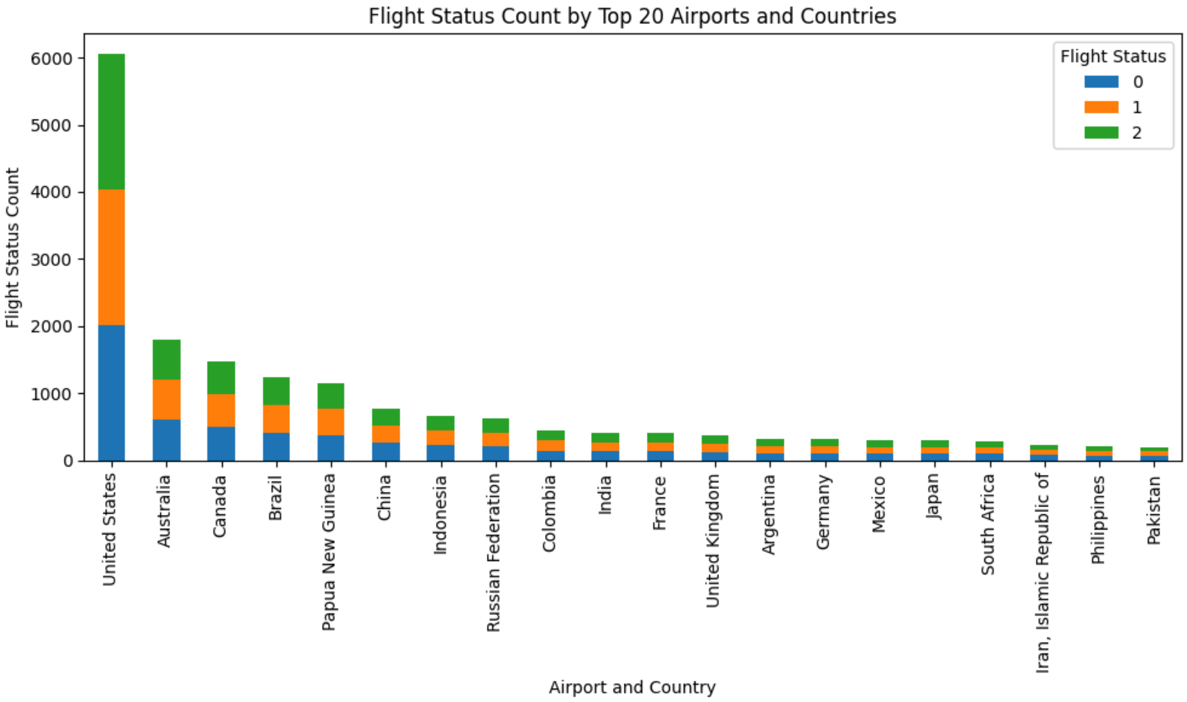
Out[41]:

Flight Status_new	**0**	**1**	**2**
Country Name			
Afghanistan	32	32	32
Albania	1	1	1
Algeria	43	43	43
American Samoa	4	4	4
Andorra	1	1	1
...	...	...	...
Wallis and Futuna	2	2	2
Western Sahara	4	4	4
Yemen	19	19	19
Zambia	21	21	21
Zimbabwe	13	13	13

235 rows × 3 columns

In [40]:

```python
# Create a pivot table to aggregate flight status counts by Airport and Country
pivot_table = df_new_melted.pivot_table(index=[ 'Country Name'], columns='Flight Status_new', aggfunc='size', fill_value=0)

# Calculate total counts for each airport and sort them to get the top 20
pivot_table['Total Flights'] = pivot_table.sum(axis=1)
top_20_airports = pivot_table.sort_values('Total Flights', ascending=False).head(20)

# Drop the 'Total Flights' column for plotting
top_20_airports = top_20_airports.drop(columns='Total Flights')

# Plotting
top_20_airports.plot(kind='bar', stacked=True, figsize=(10, 6))

plt.title('Flight Status Count by Top 20 Airports and Countries')
plt.ylabel('Flight Status Count')
plt.xlabel('Airport and Country')
plt.xticks(rotation=90)
plt.legend(title='Flight Status')
plt.tight_layout()

plt.show()
```



Conclusion:

In countries like the US, the volume of flight services is significantly higher compared to other countries. As a result, both the number of delays and cancellations are proportionally higher. The data reveals a correlation between the frequency of delays and the occurrence of cancellations. This leads to substantial revenue losses for airlines operating in such regions, and in severe cases, could even result in airlines discontinuing their services in those countries.

Suggestions:

Operational Improvements: Airlines should focus on improving their operations, particularly in regions with high flight volumes like the US. This includes optimizing scheduling, improving maintenance procedures, and enhancing crew management to reduce delays and cancellations.

Real-Time Monitoring and Predictive Analytics: Airlines can adopt predictive analytics and real-time flight monitoring systems to detect and address potential issues early, reducing the likelihood of delays and cancellations.

Customer-Centric Policies: Implement policies such as compensation or rebooking for delayed or canceled flights. This will help mitigate revenue loss by retaining customer trust and loyalty.

Government Collaboration: Airlines should work with local authorities to address any infrastructural or regulatory challenges that may contribute to flight delays and cancellations.

Alternative Routes and Flexibility: Offering flexible route options and collaborating with other airlines for shared services could minimize the impact of flight disruptions in highly affected areas.

By focusing on these areas, airlines can reduce their financial losses and improve their overall service quality in high-traffic regions like the US.