Business Understanding/Overview

The aviation industry is a vital component of the global economy, enabling the rapid movement of people and goods across the globe.

Aviation companies have been on an expansion spree in order to keep up with an ever increasing demand. Investing in this industry requires immense resources and therefore the need for an indepth analysis on what model of aircraft to invest in.

This analysis will utilize dataset from the National Transportation Safety Board that includes aviation accident data from 1962 to 2023 about civil aviation accidents.

Problem Statement

The company would like to diversify its fleet of aircrafts for commercial and private business and seeks the expertise in assessing the risks associated with different aircraft models, operational conditions, and their suitability for such a venture.

Objectives

The main objective is to identify low-risk aircraft models and operational strategies to guide the expansion in the aviation industry.

Research Questions

1. Which aircraft models have the lowest accident and fatality rates?
2. Are there differences in risk levels based on the purpose of flight?
3. During which phase of flight do most fatal injuries occur?
4. Which aircraft engine types have the lowest accident rates?
5. Are there specific parts of the world with a higher frequency of fatal injuries?

Methodology

For this analysis,CRISP-DM framework has been used.

Success Criteria

1. Total fatalities linked to each aircraft model.
2. Number of recorded incidents for each model.
3. The proportion of fatalities to total incidents for each model.

Limitations

1. Missing and incomplete Data
2. Lack of operational hours for each aircraft
3. Investment budget for expansion not provided

Data Understanding

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
data = pd.read_csv('/content/AviationData.csv', encoding='latin1')
data.head()
```

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	A
0	20001218X45444	Accident	SEA87LA080	24/10/1948	MOOSE CREEK, ID	United States	NaN	NaN	NaN	
1	20001218X45447	Accident	LAX94LA336	19/07/1962	BRIDGEPORT, CA	United States	NaN	NaN	NaN	
2	20061025X01555	Accident	NYC07LA005	30/08/1974	Saltville, VA	United States	36.922223	-81.878056	NaN	
3	20001218X45448	Accident	LAX96LA321	19/06/1977	EUREKA, CA	United States	NaN	NaN	NaN	
4	20041105X01764	Accident	CHI79FA064	02/08/1979	Canton, OH	United States	NaN	NaN	NaN	

5 rows × 31 columns

```
data.tail()
```

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	Airpo
88884	2.02212E+13	Accident	ERA23LA093	26/12/2022	Annapolis, MD	United States	NaN	NaN	NaN	
88885	2.02212E+13	Accident	ERA23LA095	26/12/2022	Hampton, NH	United States	NaN	NaN	NaN	
88886	2.02212E+13	Accident	WPR23LA075	26/12/2022	Payson, AZ	United States	341525N	1112021W	PAN	
88887	2.02212E+13	Accident	WPR23LA076	26/12/2022	Morgan, UT	United States	NaN	NaN	NaN	
88888	2.02212E+13	Accident	ERA23LA097	29/12/2022	Athens, GA	United States	NaN	NaN	NaN	

5 rows × 31 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Event.Id                88889 non-null  object
 1   Investigation.Type      88889 non-null  object
 2   Accident.Number         88889 non-null  object
 3   Event.Date              88889 non-null  object
 4   Location                88837 non-null  object
 5   Country                 88663 non-null  object
 6   Latitude                34382 non-null  object
 7   Longitude               34373 non-null  object
 8   Airport.Code            50132 non-null  object
 9   Airport.Name            52704 non-null  object
 10  Injury.Severity         87889 non-null  object
 11  Aircraft.damage         85695 non-null  object
 12  Aircraft.Category       32287 non-null  object
```

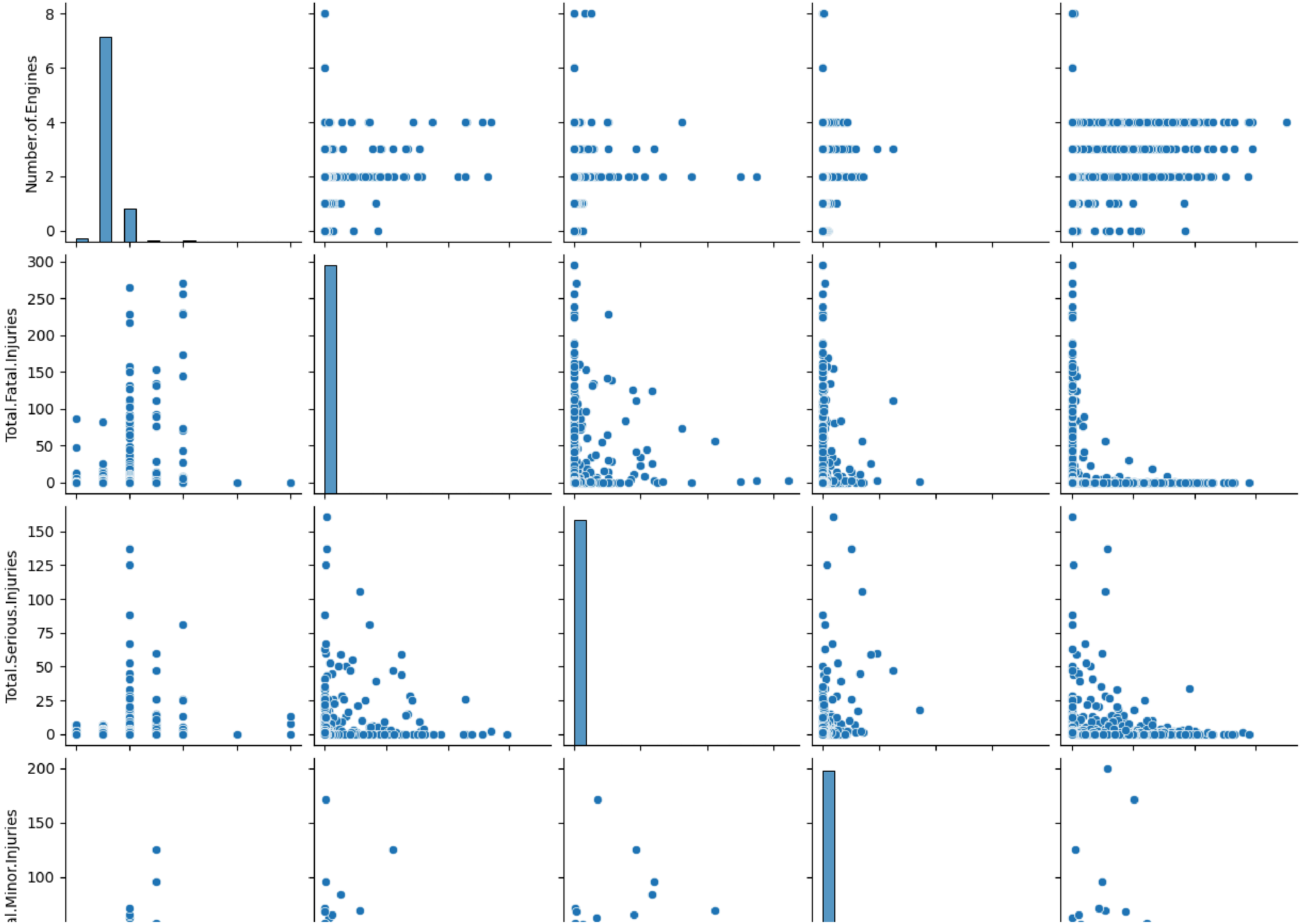
```
 13  Registration.Number     87507 non-null  object
 14  Make                    88826 non-null  object
 15  Model                   88797 non-null  object
 16  Amateur.Built           88787 non-null  object
 17  Number.of.Engines       82805 non-null  float64
 18  Engine.Type             81793 non-null  object
 19  FAR.Description         32023 non-null  object
 20  Schedule                12582 non-null  object
 21  Purpose.of.flight       82697 non-null  object
 22  Air.carrier             16648 non-null  object
 23  Total.Fatal.Injuries    77488 non-null  float64
 24  Total.Serious.Injuries  76379 non-null  float64
 25  Total.Minor.Injuries    76956 non-null  float64
 26  Total.Uninjured         82977 non-null  float64
 27  Weather.Condition       84397 non-null  object
 28  Broad.phase.of.flight   61724 non-null  object
 29  Report.Status           82505 non-null  object
 30  Publication.Date        75118 non-null  object
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

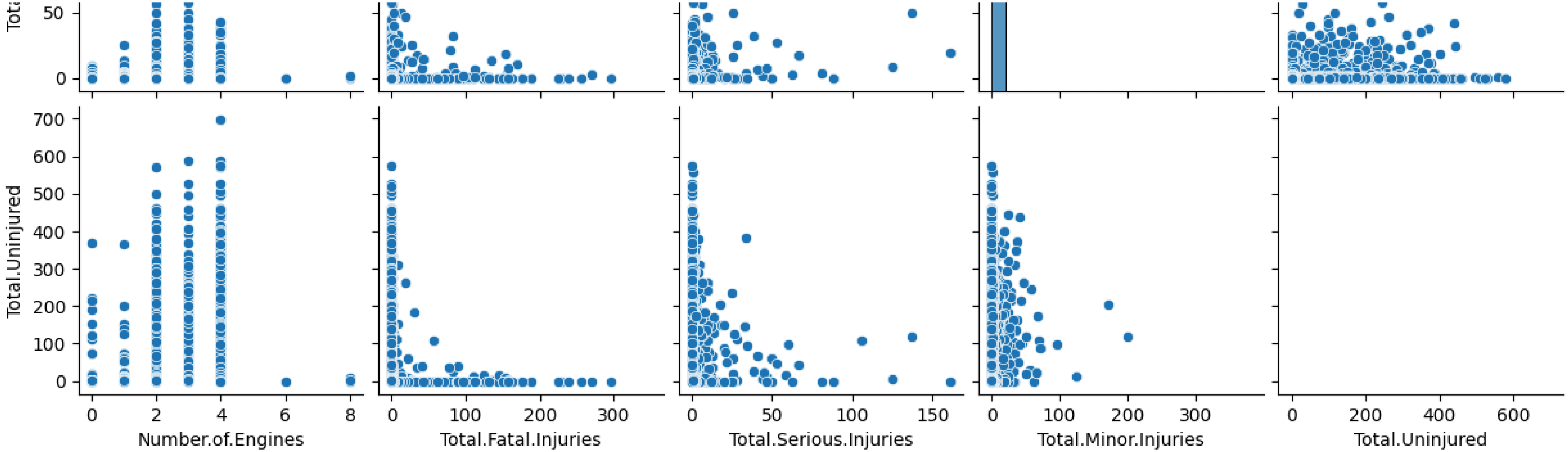
```
# Columns
data.columns
```

```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Number.of.Engines', 'Engine.Type', 'FAR.Description',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f40f5a7f040>
```




Tota
50
0
700
600
500
400
300
200
100
0
Total.Uninjured
0 2 4 6 8
Number.of.Engines
0 100 200 300
Total.Fatal.Injuries
0 50 100 150
Total.Serious.Injuries
0 100 200 300
Total.Minor.Injuries
0 200 400 600
Total.Uninjured

```
data.describe()
```

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Uninjured
count	82805.000000	77488.000000	76379.000000	76956.000000	82977.000000
mean	1.146585	0.647855	0.279881	0.357061	5.325440
std	0.446510	5.485960	1.544084	2.235625	27.913634
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000	0.000000	1.000000
75%	1.000000	0.000000	0.000000	0.000000	2.000000
max	8.000000	349.000000	161.000000	380.000000	699.000000

Data Cleaning

```
## Missing Values
data.isna().sum()
```

	0
Event.Id	0
Investigation.Type	0
Accident.Number	0
Event.Date	0
Location	52
Country	226
Latitude	54507
Longitude	54516
Airport.Code	38757
Airport.Name	36185
Injury.Severity	1000
Aircraft.damage	3194
Aircraft.Category	56602
Registration.Number	1382
Make	63
Model	92
Amateur.Built	102
Number.of.Engines	6084
Engine.Type	7096
FAR.Description	56866
Schedule	76307
Purpose.of.flight	6192

Air.carrier	72241
Total.Fatal.Injuries	11401
Total.Serious.Injuries	12510
Total.Minor.Injuries	11933
Total.Uninjured	5912
Weather.Condition	4492
Broad.phase.of.flight	27165
Report.Status	6384
Publication.Date	13771

dtype: int64

```
data.isna().mean()
```

	0
Event.Id	0.000000
Investigation.Type	0.000000
Accident.Number	0.000000
Event.Date	0.000000
Location	0.000585
Country	0.002542
Latitude	0.613203
Longitude	0.613304
Airport.Code	0.436016
Airport.Name	0.407081
Injury.Severity	0.011250
Aircraft.damage	0.035932
Aircraft.Category	0.636772
Registration.Number	0.015547
Make	0.000709
Model	0.001035
Amateur.Built	0.001147
Number.of.Engines	0.068445
Engine.Type	0.079830
FAR.Description	0.639742
Schedule	0.858453
Purpose.of.flight	0.069660

Air.carrier	0.812710
Total.Fatal.Injuries	0.128261
Total.Serious.Injuries	0.140737
Total.Minor.Injuries	0.134246
Total.Uninjured	0.066510
Weather.Condition	0.050535
Broad.phase.of.flight	0.305606
Report.Status	0.071820
Publication.Date	0.154924

dtype: float64

Handling missing values

```
## Drop columns missing values
data.dropna(axis = 1, inplace = True)
```

```
data.isna().sum()
```

	0
Event.Id	0
Investigation.Type	0
Accident.Number	0
Event.Date	0

dtype: int64

Handling duplicates

```
data.duplicated().sum()
```

```
26
```

```
data.drop_duplicates()
```

	Event.Id	Investigation.Type	Accident.Number	Event.Date
0	20001218X45444	Accident	SEA87LA080	24/10/1948
1	20001218X45447	Accident	LAX94LA336	19/07/1962
2	20061025X01555	Accident	NYC07LA005	30/08/1974
3	20001218X45448	Accident	LAX96LA321	19/06/1977
4	20041105X01764	Accident	CHI79FA064	02/08/1979
...	...	...	...	...
88884	2.02212E+13	Accident	ERA23LA093	26/12/2022
88885	2.02212E+13	Accident	ERA23LA095	26/12/2022
88886	2.02212E+13	Accident	WPR23LA075	26/12/2022
88887	2.02212E+13	Accident	WPR23LA076	26/12/2022
88888	2.02212E+13	Accident	ERA23LA097	29/12/2022

88863 rows × 4 columns

There were 26 duplicates, we proceed with data cleaning

Visualization of data

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Reloading the dataset
file_path = '/content/AviationData.csv'
aviation_data = pd.read_csv(file_path, encoding='latin1')
```

```
# Selecting relevant columns for analysis
relevant_data = aviation_data[
    ['Aircraft.Category', 'Make', 'Model', 'Purpose.of.flight', 'Total.Fatal.Injuries',
     'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured', 'Weather.Condition',
     'Broad.phase.of.flight']
]

# Converting injury columns to numeric for calculations
for col in ['Total.Fatal.Injuries', 'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured']:
    relevant_data[col] = pd.to_numeric(relevant_data[col], errors='coerce')

# Grouping data by Make and Model to calculate average incidents and injuries
risk_summary_avg = relevant_data.groupby(['Make', 'Model']).agg({
    'Total.Fatal.Injuries': 'mean',
    'Total.Serious.Injuries': 'mean',
    'Total.Minor.Injuries': 'mean',
    'Total.Uninjured': 'mean'
}).reset_index()

# Adding a column for average total incidents
risk_summary_avg['Avg.Incidents'] = (
    risk_summary_avg['Total.Fatal.Injuries'] +
    risk_summary_avg['Total.Serious.Injuries'] +
    risk_summary_avg['Total.Minor.Injuries'] +
    risk_summary_avg['Total.Uninjured']
)

# Sorting by Average Fatal Injuries for the plot
risk_summary_avg_sorted = risk_summary_avg.sort_values(by='Total.Fatal.Injuries', ascending=False)

# Filter the data for aircraft models with recorded average fatal injuries
fatal_injuries_avg_data = risk_summary_avg_sorted[risk_summary_avg_sorted['Total.Fatal.Injuries'] > 0]

# Plot the average number of fatal injuries versus aircraft model
plt.figure(figsize=(12, 6))
```