Final Project Submission

Please fill out:

- Student name: Joackim Kisenya
- Student pace: part time
- Scheduled project review date/time: 29/04/2025
- Instructor name: Maryann Mwikali
- Blog post URL:Interactive Dashboard

Business understanding

Introduction

The company I am working with is interested in purchasing and operating airplanes for commercial and private enterprises.I seek to determine which aircraft are the lowest risk for the company to start this new business venture. This will assist the head of the new aviation division make a decision on which aircraft to purchase.

Data understanding

The data Aviation_Data.csv is sourced from kaggle and it is from the National Transport safety Board.It contains information from 1962 to 2023 about civil aviation accidents and selected incidents within the United States, its territories and possessions, and international waters.

Setup

- Import pandas and Matplotlib libraries for analysis and plotting respectively.

```
import pandas as pd #import pandas library to manipulate our dataset
import matplotlib.pyplot as plt #import matplotlib library for visualizations
```

Data inspection

Load **Aviation_Data.csv** file as **Aviation** DataFrame

```
Aviation = pd.read_csv("data/Aviation_Data.csv", low_memory = False) #Reading Aviation
Aviation.head() #Displaying the first few records
```

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longit
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	NaN	N
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	NaN	N
2	20061025001555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	36.922223	-81.878
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	NaN	N
4	20041105001764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	NaN	N

5 rows × 31 columns

```
#Display information about our dataset.
Aviation.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90348 entries, 0 to 90347
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Event.Id                88889 non-null  object
 1   Investigation.Type      90348 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            50704 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     87572 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                12584 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        73659 non-null  object
dtypes: float64(5), object(26)
memory usage: 21.4+ MB
```

```
#print number of rows and columns
num_rows, num_cols = Aviation.shape
print(f'Number of rows: {num_rows}')
print(f'Number of columns: {num_cols}')
```

```
Number of rows: 90348
Number of columns: 31
```

```
#checking for missing values in our dataset
Aviation.isnull().sum()
```

```
Event.Id                  1459
Investigation.Type           0
Accident.Number           1459
Event.Date                1459
Location                  1511
Country                   1685
Latitude                 55966
Longitude                55975
Airport.Code             40216
Airport.Name             37644
Injury.Severity           2459
Aircraft.damage           4653
Aircraft.Category        58061
Registration.Number       1776
Make                      1522
Model                     1551
Amateur.Built             1561
Number.of.Engines         7543
Engine.Type               8555
FAR.Description          58325
Schedule                 77766
Purpose.of.flight         7651
Air.carrier              73700
Total.Fatal.Injuries     12860
Total.Serious.Injuries   13969
Total.Minor.Injuries     13392
Total.Uninjured           7371
Weather.Condition         5951
Broad.phase.of.flight    28624
Report.Status             7843
Publication.Date         16689
dtype: int64
```

```
#checking for percentage of missing values in our dataset
((Aviation.isnull().sum()/Aviation.isnull().sum().sum()) *100).round(2)
```

```
Event.Id                  1.61
Investigation.Type        0.00
Accident.Number           1.61
Event.Date                1.61
Location                  1.67
Country                   1.87
Latitude                 61.96
Longitude                61.98
Airport.Code             44.51
Airport.Name             41.67
Injury.Severity           2.72
Aircraft.damage           5.15
Aircraft.Category        64.26
Registration.Number       1.97
Make                      1.68
Model                     1.72
Number.of.Engines         8.35
Engine.Type               9.47
FAR.Description          64.56
Schedule                 86.07
Purpose.of.flight         8.47
Air.carrier              81.58
Total.Fatal.Injuries     14.23
Total.Serious.Injuries   15.46
Total.Minor.Injuries     14.82
Total.Uninjured           8.16
Weather.Condition         6.59
Broad.phase.of.flight    31.68
Report.Status             8.68
Publication.Date         18.47
dtype: float64
```

Data preparation

- Drop some columns with over 40% missing values.

```
# Drop selected columns with more than 40% missing values
Aviation = Aviation.drop(['Latitude', 'Longitude','Airport.Code','Airport.Name','FAR.Description',
Aviation = Aviation.drop(columns=drop_cols)
Aviation.isnull().sum()
Aviation.shape
```

```
(90344, 24)
```

```
# Drop rows with missing values in key columns.
Aviation.dropna(subset=['Make', 'Model', 'Location', 'Total.Fatal.Injuries', 'Aircraft
                        'Purpose.of.flight','Aircraft.damage'], inplace=True)
```

```
# Strip whitespace
Aviation = Aviation.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

```
# Convert 'Total.Fatal.Injuries' to numeric
Aviation['Total.Fatal.Injuries'] = pd.to_numeric(Aviation['Total.Fatal.Injuries'], er
```

```
# Filter out rows with invalid or negative injury values
Aviation = Aviation[Aviation['Total.Fatal.Injuries'] >= 0]
Aviation.shape
```

```
(23194, 24)
```

Categorize data

- Filter data to required private and commercial airplanes
- Filter data by Airplane category

```
# Define the custom order for sorting
custom_order = ["Personal", "Business", "Ferry", "Executive/Corporate"]

# Sort the dataframe based on the custom order
Aviation = Aviation[Aviation['Purpose.of.flight'].isin(custom_order)].copy()
Aviation_sorted['Purpose.of.flight'] = pd.Categorical(Aviation_sorted['Purpose.of.fli
Aviation_sorted = Aviation_sorted.sort_values('Purpose.of.flight')
```

```
Aviation_sorted.shape
```

```
(13994, 24)
```

```
#Filter data by Airplane category
Airplanes = Aviation_sorted[Aviation_sorted['Aircraft.Category'] == 'Airplane']
Airplanes.to_csv('airplanes_data.csv', index=False)
```

Analyze Risk by Airplane Make and Model

- Fatality rate by Make
- Fatality rate by Model
- Popular low risk airplanes

```
K = 50
```

```
# Filter makes with at least K accidents
makes_with_min_accidents = Airplanes.groupby('Make').filter(lambda x: len(x) >= K)
```

```
# Calculate fatality rate for each make
fatality_rates_by_make = makes_with_min_accidents.groupby('Make').agg(
    Total_Accidents=('Event.Id', 'count'),
    Total_Fatalities=('Total.Fatal.Injuries', 'sum')
)
fatality_rates_by_make['fatality_rate'] = fatality_rates_by_make['total_fatalities']
```

```
# Sort by fatality rate in ascending order
lowest_fatality_rates_by_make = fatality_rates_by_make.sort_values('fatality_rate', a
```

```
K = 50
```

```
# Filter models with at least K accidents
models_with_min_accidents = Airplanes.groupby('Model').filter(lambda x: len(x) >= K)
```

```
# Calculate fatality rate for each model
fatality_rates = models_with_min_accidents.groupby('Model').agg(
    Total_Accidents=('Event.Id', 'count'),
    Total_Fatalities=('Total.Fatal.Injuries', 'sum')
)
fatality_rates['fatality_rate'] = fatality_rates['total_fatalities'] / fatality_rates
```

```
# Sort by fatality rate in ascending order
lowest_fatality_rates = fatality_rates.sort_values('fatality_rate', ascending=True)
```

```
# Group by Make and Model
risk_analysis = Airplanes.groupby(["Make", "Model"]).agg(
    Total_Accidents=('Event.Id', 'count'),
    Total_Fatalities=('Total.Fatal.Injuries', 'sum'),
    Avg_Fatalities_Per_Accident=('Total.Fatal.Injuries', 'mean')
).reset_index()
```

```
# Sort by total fatalities to identify low-risk aircraft
low_risk_aircraft = risk_analysis.sort_values(by='Total_Fatalities', ascending=True)
```

```
# Group by Make and Model for popular airplanes
popular_airplanes = Airplanes.groupby(["Make", "Model"]).agg(
    Total_Accidents=('Event.Id', 'count'),
    Total_Fatalities=('Total.Fatal.Injuries', 'sum'),
    Avg_Fatalities_Per_Accident=('Total.Fatal.Injuries', 'mean')
).reset_index()
```

```
# Define a threshold for popularity (e.g., aircraft with more than 50 accidents)
popular_airplanes = popular_airplanes[popular_airplanes["Total_Accidents"] > 50]
```

```
# Sort by Total_Accidents in descending order
popular_airplanes = popular_airplanes.sort_values(by="Total_Accidents", ascending=False
popular_airplanes.head(10)
```

	Make	Model	Total Accidents	Total.Fatalities	Avg_Fatalities_Per_Accident
938	CESSNA	172	228	56.0	0.245614
1394	Cessna	172	175	34.0	0.194286
1235	CIRRUS DESIGN CORP	SR22	115	7.0	0.060870
971	CESSNA	180	114	11.0	0.096491
983	CESSNA	182	110	30.0	0.272727
435	BEECH	A36	108	79.0	0.731481
3685	PIPER	PA-18-150	107	5.0	0.046729
1432	Cessna	182	98	37.0	0.377551
3706	PIPER	PA28	97	40.0	0.412371
953	CESSNA	210N	92	32.0	0.347826

```
popular_airplanes.to_csv('popular_airplanes', index=False) #Dump the dataset to a csv
```

Visualizations

- Fatalities by Make
- Fatalities by Model
- Low-risk airplanes

```
top_10_lowest_fatality_rates = lowest_fatality_rates_by_make.head(10)
```

```
# Plot the data
plt.figure(figsize=(10, 6))
plt.bar(lowest_fatality_rates_by_make.index, top_10_lowest_fatality_rates['fatality_ra
plt.xlabel('Make', fontsize=12)
plt.ylabel('Fatality Rate', fontsize=12)
plt.title('Top 10 Makes with Lowest Fatality Rates', fontsize=14)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



```
# Select the top 10 low-risk popular aircraft
top_10_low_risk_popular_aircraft = popular_airplanes.head(10)

# Plot the data
plt.figure(figsize=(10, 6))
plt.bar(top_10_low_risk_popular_aircraft['Make'] + " " + top_10_low_risk_popular_airc
        top_10_low_risk_popular_aircraft['Total_Fatalities'], color='orange')
plt.xlabel('Aircraft (Make and Model)', fontsize=12)
plt.ylabel('Total Fatalities', fontsize=12)
plt.title('Top 10 Low-Risk Aircraft by Total Fatalities', fontsize=14)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(10, 6))
plt.bar(lowest_fatality_rates.index[:10], lowest_fatality_rates['fatality_rate'][:10],
plt.xlabel('Model', fontsize=12)
plt.ylabel('Fatality Rate', fontsize=12)
plt.title('Top 10 Models with Lowest Fatality Rates', fontsize=14)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```


RECOMMENDATION

From my analysis of Aviation_Data.csv; here are my findings:

1. Purchase CESSNA 172 model which has a low fatality rate.
2. CESSNA airplane make has low fatality rates and should be considered as the best manufacturer among the ones analyzed.
3. Cessna 180 and Piper Pa 18-150 models have the lowest fatalities; Recommend purchase of these models.