Identifying A Low Risk Aircraft For A Strong Start In Aviation

This project aims to identify the lowest-risk aircraft for a new business venture into the aviation industry. It covers the following:

- Business Understanding
- Data Understanding
- Data Preparation
- Data Visualization

1. Business Undestanding ¶

Objective

In this project, a comprehensive risk assessment has been conducted to identify low-risk aircraft options for commercial operations.

Stakeholders

- The Executive: Responsible for the long-term benefits and financial risks associated with the Industry.
- The Head of Aviation: Responsible for operationalizing the expansion into the aviation industry and for the day-to-day logistics.
- The Finance Team: Responsible for ensuring financial sustainability.
- The Legal Team: They focus on the regulatory requirements and risk mitigation in the aviation sector.

Key Considerations

- The type of aircraft to be used for commercial and private enterprises.
- How the company can leverage on past data to make informed decisions about aircraft acquisition.

2. Data Understanding

Source of Data

The data used in this notebook is derived from **National Transportation Safety Board(NTSB) Aviation Accident Database** that includes aviation accident data from 1962-2023 about civil aviation accidents and selected incidents in the United States and international waters. The data can be accessed publicly from ([https://www.kaggle.com/datasets/khsamaha/aviation-accident-database-synopses (https://www.kaggle.com/datasets/khsamaha/aviation-accident-database-synopses)](https://www.kaggle.com/datasets/khsamaha/aviation-accident-database-synopses))

Data Description

This dataset contains attributes such as:

- Accident number
- Weather condition
- Investigation type
- Purpose of flight
- Engine type
- Country
- Event date

Load the Dataset

In [5]:

```
# Run the cell without changes
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

file_path = r"C:\Users\ADMIN\Downloads\extracted_files\AviationData.csv
aviation_data = pd.read_csv(file_path, encoding='ISO-8859-1', low_memor
aviation_data.head()
```

Out[5]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Cou
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	US
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	US
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	US
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	US
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	US

5 rows × 31 columns

3. Data Preparation

It involves the following:

- Handling missing values
- Filter for relevant data
- Converting dates to datetime format
- Injury severity score

In [6]:

```python
# Summary statistics
aviation_data.info()
aviation_data.describe()
aviation_data.shape
```

```
<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            50249 non-null  object 
 9   Airport.Name            52790 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             81812 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           82508 non-null  object 
 30  Publication.Date        75118 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

Out[6]:

```
(88889, 31)
```

In [7]:

```python
# Check for missing values
aviation_data.isna().sum()
```

Out[7]:

```
Event.Id                      0
Investigation.Type            0
Accident.Number               0
Event.Date                    0
Location                     52
Country                     226
Latitude                  54507
Longitude                 54516
Airport.Code              38640
Airport.Name              36099
Injury.Severity            1000
Aircraft.damage            3194
Aircraft.Category         56602
Registration.Number        1317
Make                         63
Model                        92
Amateur.Built               102
Number.of.Engines          6084
Engine.Type                7077
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              6381
Publication.Date          13771
dtype: int64
```

In [8]:

```
# Cleaning the dataset
aviation_data_cleaned = aviation_data.dropna(subset=['Weather.Condition
aviation_data_cleaned = aviation_data_cleaned.reset_index(drop=True)
print(aviation_data_cleaned)
```

```
             Event.Id Investigation.Type Accident.Number  Event.Date  
\
0      20001218X45444           Accident      SEA87LA080  1948-10-24
1      20001218X45447           Accident      LAX94LA336  1962-07-19
2      20061025X01555           Accident      NYC07LA005  1974-08-30
3      20001218X45448           Accident      LAX96LA321  1977-06-19
4      20041105X01764           Accident      CHI79FA064  1979-08-02
...               ...                ...             ...         ...
84392  20221212106443           Accident      WPR23LA064  2022-12-09
84393  20221212106444           Accident      ERA23LA085  2022-12-12
84394  20221215106463           Accident      ERA23LA090  2022-12-14
84395  20221219106470           Accident      ERA23LA091  2022-12-16
84396  20221227106497           Accident      WPR23LA075  2022-12-26

              Location        Country   Latitude   Longitude Airport.C
ode  \
0      MOOSE CREEK, ID  United States        NaN         NaN          
NaN
1       BRIDGEPORT, CA  United States        NaN         NaN          
NaN
2        Saltville, VA  United States  36.922223  -81.878056          
NaN
3           EUREKA, CA  United States        NaN         NaN          
NaN
4           Canton, OH  United States        NaN         NaN          
NaN
...                ...            ...        ...         ...          
...
84392  Casa Grande, AZ  United States    325736N    1114536W          
CGZ
84393    Knoxville, TN  United States    355745N    0835218W          
DKX
84394     San Juan, PR  United States    182724N    0066554W          
SIG
84395  Brooksville, FL  United States    282825N    0822719W          
BKV
84396       Payson, AZ  United States    341525N    1112021W          
PAN

                         Airport.Name  ... Purpose.of.flight  \
0                                 NaN  ...          Personal   
1                                 NaN  ...          Personal   
2                                 NaN  ...          Personal   
3                                 NaN  ...          Personal   
4                                 NaN  ...          Personal   
...                               ...  ...               ...   
84392  Casa Grande Municipal Airport  ...          Personal   
84393      KNOXVILLE DOWNTOWN ISLAND  ...     Instructional   
84394  FERNANDO LUIS RIBAS DOMINICCI  ...          Personal   
84395     BROOKSVILLE-TAMPA BAY RGNL  ...          Personal   
84396                         PAYSON  ...          Personal   

                           Air.carrier Total.Fatal.Injuries  \
0                                  NaN                  2.0   
1                                  NaN                  4.0   
2                                  NaN                  3.0   
3                                  NaN                  2.0   
4                                  NaN                  1.0   
...                                ...                  ...   
84392                              NaN                  0.0   
84393  Knoxville Flight Training Academy                  0.0   
```

```
84394      SKY WEST AVIATION INC TRUSTEE                  0.0
84395                  GERBER RICHARD E                  0.0
84396                               NaN                  0.0

      Total.Serious.Injuries  Total.Minor.Injuries  Total.Uninjured  \
0                        0.0                   0.0              0.0
1                        0.0                   0.0              0.0
2                        NaN                   NaN              NaN
3                        0.0                   0.0              0.0
4                        2.0                   NaN              0.0
...                      ...                   ...              ...
84392                    0.0                   0.0              1.0
84393                    0.0                   0.0              1.0
84394                    0.0                   0.0              1.0
84395                    1.0                   0.0              0.0
84396                    0.0                   0.0              1.0

      Weather.Condition  Broad.phase.of.flight   Report.Status  \
0                   UNK                 Cruise  Probable Cause
1                   UNK                Unknown  Probable Cause
2                   IMC                 Cruise  Probable Cause
3                   IMC                 Cruise  Probable Cause
4                   VMC               Approach  Probable Cause
...                 ...                    ...             ...
84392               VMC                    NaN             NaN
84393               VMC                    NaN             NaN
84394               VMC                    NaN             NaN
84395               VMC                    NaN             NaN
84396               VMC                    NaN             NaN

      Publication.Date
0                  NaN
1           19-09-1996
2           26-02-2007
3           12-09-2000
4           16-04-1980
...                ...
84392       13-12-2022
84393       15-12-2022
84394       27-12-2022
84395       23-12-2022
84396       27-12-2022

[84397 rows x 31 columns]
```

In [9]:

```python
# Convert dates to datetime format
aviation_data['Event.Date'] = pd.to_datetime(aviation_data['Event.Date'
```

In [10]:

```python
# Injury severity score
aviation_data['Injury severity'] = (
    aviation_data['Total.Fatal.Injuries'] * 3 +
    aviation_data['Total.Serious.Injuries'] * 2 +
    aviation_data['Total.Minor.Injuries'] * 1
)
```

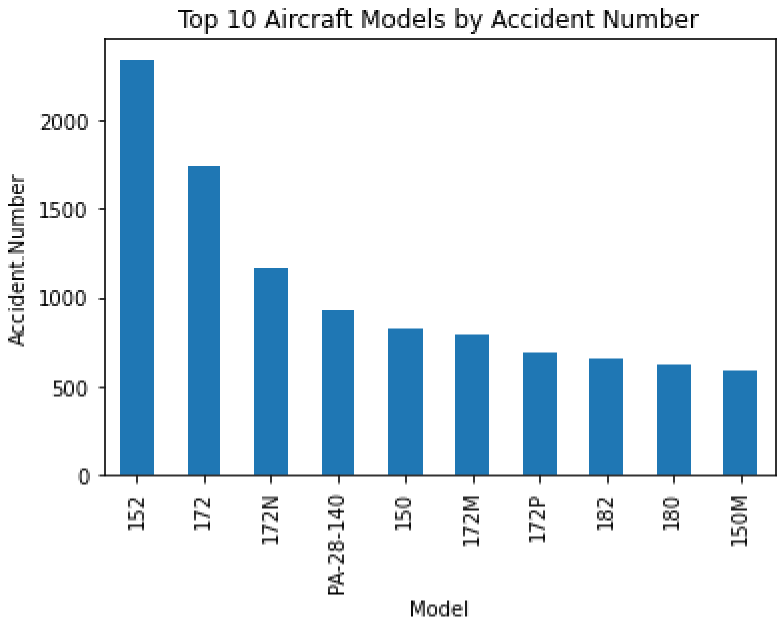
In [11]:
```python
# Filter for relevant data
aviation_data = aviation_data[aviation_data['Investigation.Type'] == 'A
```

4. Data Visualization

Key Business Questions

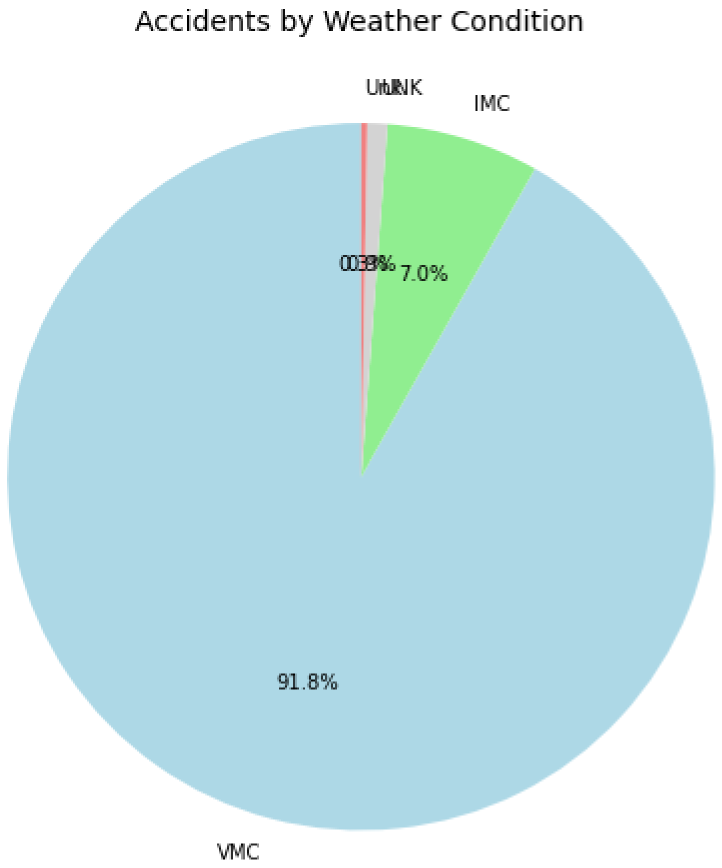
- Which aircraft has the lowest risk based on the accident history?
- What weather conditions correlate with higher risks?
- How the number of engines in an aircraft translates to the degree of injuries in case of an accident.

In [12]:
```python
# Lowest risk aircraft
model_accidents = aviation_data['Model'].value_counts().head(10)
model_accidents.plot(kind='bar', title='Top 10 Aircraft Models by Accid
plt.show()
```



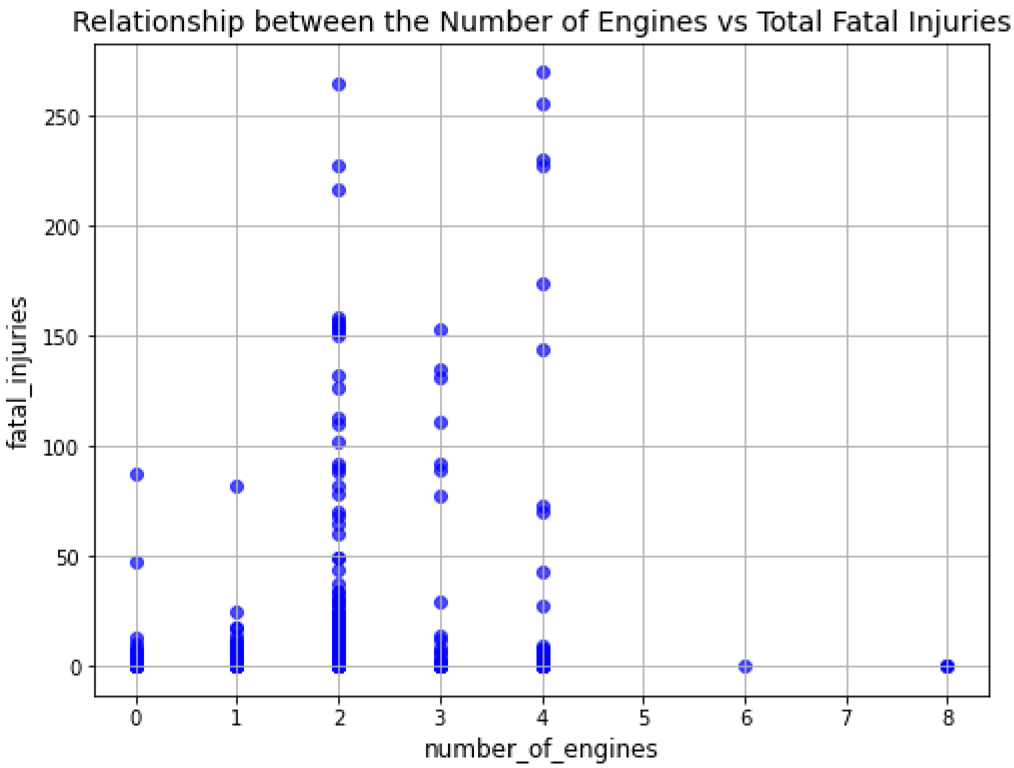
In [13]:

```
# Number of accidents by weather condition
weather_counts = aviation_data['Weather.Condition'].value_counts()

# Create a pie chart
plt.figure(figsize=(7, 8))
plt.pie(weather_counts, labels=weather_counts.index, autopct='%1.1f%%',
plt.title('Accidents by Weather Condition', fontsize=14)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a

# Show the plot
plt.show()
```



In [14]:

```python
# Scatter plot
number_of_engines= aviation_data['Number.of.Engines']
fatal_injuries= aviation_data['Total.Fatal.Injuries']

plt.figure(figsize=(8, 6))
plt.scatter(number_of_engines, fatal_injuries, color='blue', alpha=0.7)
plt.title('Relationship between the Number of Engines vs Total Fatal In
plt.xlabel('number_of_engines', fontsize=12)
plt.ylabel('fatal_injuries', fontsize=12)
plt.grid(True)
plt.show()
```



5. Conclusion

Key Findings:

- Aircraft models with the lowest accident rates.
- Weather conditions associated with high risks.
- The impact the number of engines has on the total fatal injuries.

Summary

- Aircraft models with fewer engines are associated with higher risks in adverse weather.
- Aircraft models with single engines should be avoided in places with harsh weather conditions.

Recommendations:

- The company should focus on acquiring aircrafts with low acciddent numbers.
- More training should be offered with regards to adverse weather conditions.

- Aircraft models with single engines should be avoided in places with adverse weather