FLIGHT DELAY REASON CLASSIFICATION

Deep Learning and ML for Accurate Delay Type Prediction

Abstract:

This project focuses on enhancing the accuracy of flight delay reason classification using deep learning and machine learning techniques. By analyzing a comprehensive dataset of flight delays and their contributing factors, we develop predictive models to identify the underlying reasons for delays. Utilizing neural networks and advanced ML algorithms, we aim to classify delay types into distinct categories, such as security, weather, and airline-related delays. Our approach integrates data preprocessing, feature engineering, and model optimization to provide actionable insights for improving flight management and operational efficiency. This work contributes to more precise delay predictions, ultimately aiding in better decision-making and resource allocation in the aviation industry.

Lyana Murad

Tables of Contents

Introduction About Data

Overview

This dataset appears to be related to flight information, with each row representing a flight and various columns providing details about the flight's schedule, delays, and other attributes. Here's a breakdown of the important aspects:

- **Total Records**: 1,048,575 non-null entries for most columns, indicating the dataset is large with over a million flights recorded.
- **Numerical Columns**:
    - There are columns representing **year, month, day**, **day of the week**, and various numerical details such as **flight numbers**, **scheduled times**, **actual times**, and **delays**.
    - Notably, columns like **departure time**, **departure delay**, **air time**, and **arrival time** have some missing values, indicating that not all flights had complete data.
    - Columns for various types of delays (e.g., **airline delay, security delay**) have significant missing data, possibly because not every flight experienced delays.
- **Categorical Columns**:
    - **Airline**, **tail number**, **origin airport**, and **destination airport** are stored as objects (strings), allowing identification of the flight and its route.
    - **Cancellation reason** has many missing values (with only 40,527 non-null entries), implying that most flights were not canceled.
- **Missing Data**: Several columns have missing values, especially delay-related columns and departure/arrival times, which might require cleaning or imputation for further analysis.
- **Memory Usage**: The dataset takes up around 248 MB in memory, which is typical for a dataset of this size.

This dataset could be useful for analyzing flight delays, cancellations, and general patterns in airline operations.

Objective

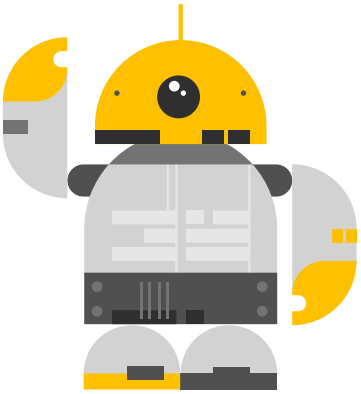
The objective of analyzing this flight dataset is to gain insights into various aspects of airline operations, including:

1. **Flight Delays and Cancellations**: To understand the patterns and causes of delays and cancellations, including the impact of different types of delays (e.g., airline delay, security delay, weather delay) on flight schedules.

2. **Operational Efficiency**: To evaluate the efficiency of flight operations by examining time metrics such as departure delays, taxi times, and elapsed times.
3. **Scheduling Patterns**: To identify trends in flight scheduling, including peak times for departures and arrivals, and how these affects overall flight performance.
4. **Flight Patterns and Routes**: To analyze flight routes, including common origins and destinations, and how these factors contribute to operational delays and efficiencies.
5. **Impact of External Factors**: To assess how external factors such as weather and security impact flight operations and contribute to delays or cancellations.
6. **Data Quality and Completeness**: To evaluate the completeness of the dataset and the presence of missing data, which may affect the accuracy of the analysis and require appropriate handling.
7. **Building a Predictive Model**: To develop a predictive model capable of forecasting delay reasons flights and identifying the factors influencing cancellations, in order to improve planning and make more accurate decisions in flight management.

By achieving these objectives, the analysis aims to enhance understanding of flight operations, improve airline performance, increase customer satisfaction, and provide predictive tools to support better flight management.

Data Analysis



Analysis

Data Column Types Overview

Discrete Numerical Columns:

- **YEAR:** The specific year of the flight.
- **MONTH:** The month of the flight (a number from 1 to 12).
- **DAY:** The day of the month (a number from 1 to 31).
- **DAY_OF_WEEK:** The day of the week (a number representing a specific day, e.g., 1 for Sunday).
- **FLIGHT_NUMBER:** A unique identifier for each flight.
- **DIVERTED:** Whether the flight was diverted (0 or 1).
- **CANCELLED:** Whether the flight was canceled (0 or 1).

Why discrete? These columns represent distinct categories or classifications and cannot take on any random value between two values.

Categorical Columns:

- **AIRLINE:** The name of the airline.
- **TAIL_NUMBER:** The tail number of the aircraft.
- **ORIGIN_AIRPORT:** The departure airport.
- **DESTINATION_AIRPORT:** The arrival airport.
- **CANCELLATION_REASON:** The reason for cancellation.

Why categorical? These columns represent different categories or groups and do not have a numerical value that can be used for mathematical operations.

Continuous Numerical Columns:

- **SCHEDULED_DEPARTURE:** The scheduled departure time.
- **DEPARTURE_TIME:** The actual departure time.
- **DEPARTURE_DELAY:** The departure delay in minutes.
- **TAXI_OUT:** The time it takes for the aircraft to taxi from the gate to the runway.
- **WHEELS_OFF:** The time when the aircraft leaves the ground.

- **SCHEDULED_TIME:** The scheduled flight time.
- **ELAPSED_TIME:** The actual flight time.
- **AIR_TIME:** The time the aircraft is in the air.
- **DISTANCE:** The distance traveled.
- **WHEELS_ON:** The time when the wheels touch down.
- **TAXI_IN:** The time it takes for the aircraft to taxi from the runway to the gate.
- **SCHEDULED_ARRIVAL:** The scheduled arrival time.
- **ARRIVAL_TIME:** The actual arrival time.
- **ARRIVAL_DELAY:** The arrival delay in minutes.
- **AIR_SYSTEM_DELAY:** Delay due to the air traffic control system.
- **SECURITY_DELAY:** Delay due to security.
- **AIRLINE_DELAY:** Delay due to the airline.
- **LATE_AIRCRAFT_DELAY:** Delay due to a late arriving aircraft.
- **WEATHER_DELAY:** Delay due to weather conditions.

Why continuous? These columns represent numerical values that can take on any value within a range and can be used for mathematical operations like mean and standard deviation.

Discrete Numerical Columns:

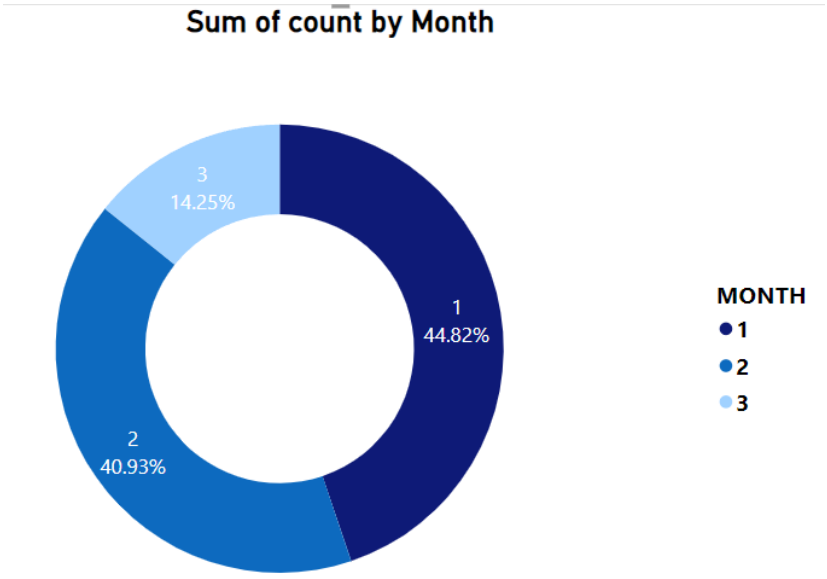
- **Column(1): YEAR**

YEAR	count
2015	1048575

In this context, it seems that every row in the dataset corresponds to a flight that took place in 2015, and the dataset includes over one million flights for that year. The count (1,048,575) indicates the total number of flights or records in the dataset for the given year.

- **Column(2):MONTH**

MONTH	count
1	469968
2	429191
3	149416



In analyzing travel data, we observe that January (the first month) experiences the highest travel rates compared to other months, followed by February (the second month) in the second position. Conversely, March (the third month) shows the lowest travel rates. These results could be attributed to various factors, such as seasonal holidays or special events that influence travel preferences during these periods.

Reasons for Increased Travel in January and February Compared to March:

Several factors can explain the variation in flight numbers between January, February, and March:

1. **Holidays and Events:**
    - **School Holidays:** January and February often coincide with school holidays in many countries, prompting families to plan trips during this period.
    - **Christmas and New Year:** Many cultures celebrate Christmas and New Year at the end of December and the beginning of January, encouraging people to travel for celebrations or short vacations.
    - **Holiday Timing**: A significant portion of annual holidays occurs in the early months of the year, contributing to increased travel.
2. **Weather Conditions:**
    - **Escaping the Cold:** Many people seek to escape the harsh winter weather in their home countries and travel to warmer destinations during January and February.
    - **Tourism Seasons:** Winter is a peak tourist season in many warm destinations, such as the Caribbean and Southeast Asia.
3. **Ticket Prices:**
    - **Promotional Offers:** Airlines often release special promotions during this period to attract travelers, making ticket prices more appealing.
    - **Demand and Supply:** Ticket prices may be less inflated at the beginning of the year compared to the period leading up to summer holidays.
4. **Travel Planning:**
    - **Advance Booking**: People tend to plan their trips in January and February in advance to take advantage of promotions and secure flight reservations.

Why March Might Be Different:

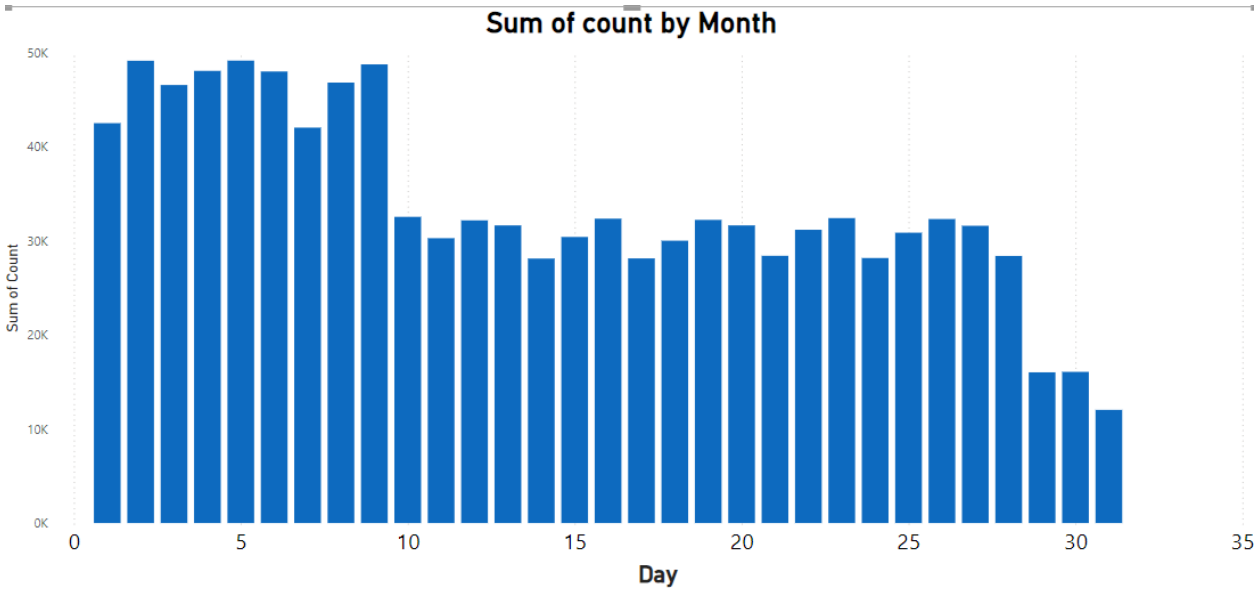
- **End of Peak Season:** With the end of school holidays and year-end celebrations, travel demand begins to decline gradually.
- **Spring Beginning:** In some regions, spring begins in March, which may attract travelers to other destinations.
- **Return to Routine:** After the holiday period, many people return to their daily routines, reducing their desire to travel.

Other Influencing Factors:

- **Global Events:** Global events such as economic crises or natural disasters may affect travel rates.
- **Traveler Preferences:** Traveler preferences vary; some prefer to travel during quieter periods, while others prefer to travel during peak seasons.

These hypotheses combine seasonal patterns, traveler behavior, and operational factors to explain why there are more flights in January and February compared to March.

- **Column(3):DAY**



The flight data for different days of the month shows considerable variation in the number of flights on each day. Here's a summary of the observations and possible reasons behind this variation:

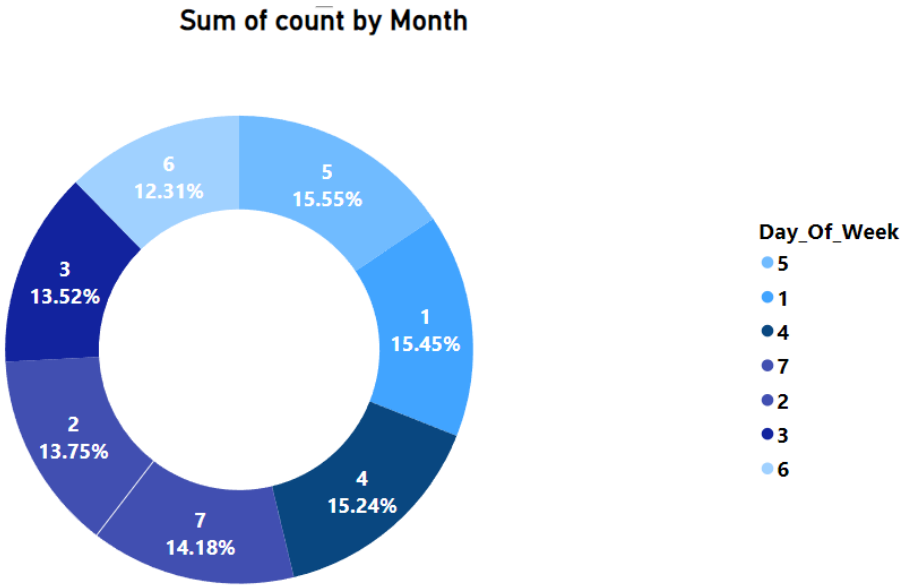
1. **Higher Counts on Specific Days:**
    - Days 5, 2, 9, 4, 6, and 8 have the highest flight counts, with Day 5 having the most at 49,180 flights.
    - These days are relatively evenly distributed throughout the month, suggesting a general pattern of higher flight activity spread across different days.
2. **Lower Counts Towards the End of the Month:**
    - Days 30 and 31 have the lowest flight counts, with Day 30 and Day 31 recording significantly fewer flights compared to earlier days.
    - This trend could be due to fewer days available for flights towards the end of the month, with Day 31 only occurring in months with 31 days.
3. **Possible Reasons for Variation:**

- **Month-Length Variation:** The number of days in a month can affect flight scheduling. Months with 30 or 31 days naturally have more flights available compared to months with 28 or 29 days.
- **Business vs. Leisure Travel:** The demand for flights might be influenced by weekdays versus weekends. For instance, weekdays might show higher flight numbers due to business travel, while weekends might have varied patterns.
- **Operational Factors:** Airlines might schedule more flights on certain days of the week or dates based on anticipated demand, such as holiday seasons or peak travel periods.
- **Flight Scheduling:** Some airlines might have fixed schedules that impact flight numbers on specific days, creating patterns where certain days consistently have more or fewer flights.

In summary, the variation in flight counts across different days of the month can be attributed to a combination of monthly length, demand patterns related to business and leisure travel, and airline scheduling practices. The lower flight counts towards the end of the month, especially on Days 30 and 31, may be influenced by the number of days available for flights in the month and operational scheduling constraints.

- **Column(4):DAY_OF_WEEK**

DAY_OF_WEEK	
5	163070
1	162041
4	159800
7	148678
2	144193
3	141753
6	129040



Here's the analysis with the days of the week named as you specified:

Possible Reasons for Variation:

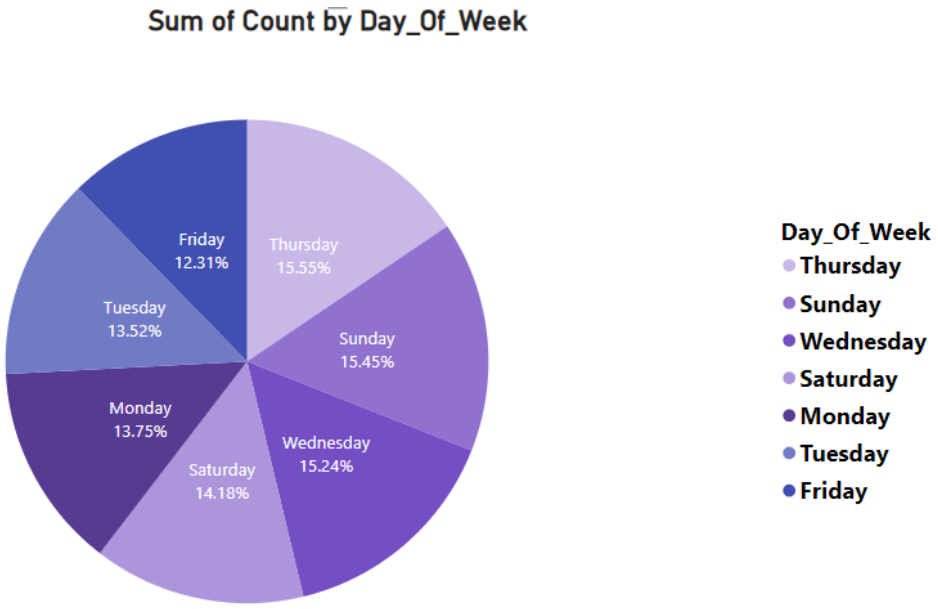
1. **Increased Traffic on Specific Days:**

- High Counts on Days 1, 2, 4, 5, and 7: These days show significantly higher flight counts, with Day 5 (Thursday) having the highest count of 163,070 flights.
    - **Possible Explanation:**
        - **Beginning of the Month:** Day 1 (Sunday) could see a high number of flights as people start new trips or return from end-of-month holidays**.**
        - **Weekend Patterns:** If Days 1 (Sunday) or Day 2 (Monday) fall near weekends, travel demand might increase as people take advantage of longer weekends for trips.

2. **Lower Counts on Day 6:**
    - **Day 6 (Friday):** Shows a significantly lower flight count of 12,904 flights compared to other days.
        - **Possible Explanation:**
            - **Weekday Influence**: Day 6 (Friday) could be a weekday with lower travel demand compared to weekends or holidays, leading to fewer flights.
            - **Operational Factors:** It's possible that fewer flights are scheduled on this day due to operational constraints or lower expected demand.
3. **General Trends and Patterns:**
    - **Variation in Demand:** Flight counts can vary widely depending on numerous factors, including:
        - **Travel Trends:** Specific days might align with peak travel periods, such as holidays, school vacation times, or significant events, leading to higher flight volumes.
        - **Airline Scheduling:** Airlines may schedule more flights on certain days to match anticipated demand, while other days may have fewer flights scheduled based on operational needs or lower expected passenger numbers.
4. **Other Influencing Factors:**
    - **Seasonal Variations:** Depending on the time of year, certain days may naturally have higher travel volumes due to seasonal trends.
    - **Regional Factors:** Local events or regional holidays might impact flight counts on specific days

Note: I changed the day of the week from numerical to categorical to make it easier to read and understand

DAY_OF_WEEK	count
Thursday	163070
Sunday	162041
Wednesday	159800
Saturday	148678
Monday	144193
Tuesday	141753



The flight data shows a notable variation in flight counts across different days of the week, likely influenced by patterns of business and leisure travel. Business travelers typically fly on weekdays (Monday to Friday), which may explain the relatively higher counts on Thursdays, as travelers might be returning home or starting their weekend early. Conversely, weekends (Saturday and Sunday) often see increased leisure travel, leading to higher flight counts on these days. Additionally, airlines may adjust their flight schedules to meet demand, with potentially more flights on weekends to cater to leisure travelers and on Thursdays to accommodate business travelers. Other factors, such as holidays, special events, or seasonal trends, could also contribute to fluctuations in flight counts throughout the week.

_____

- **Column(5): FLIGHT_NUMBER**

FLIGHT_NUMBER is a unique identifier used to specify a particular flight. It typically consists of an airline code (usually two letters) followed by a flight number (usually 1 to 4 digits). For example, "DL1234" is a flight number for Delta Airlines.

Importance of FLIGHT_NUMBER:

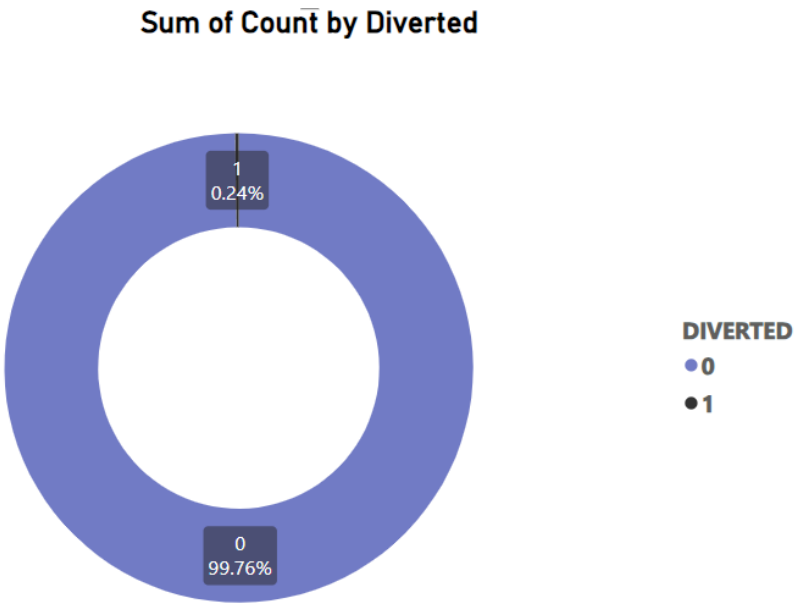
- **Flight Identification:** The flight number is used to pinpoint a specific flight among all the flights operated by an airline.
- **Ticket Booking:** When booking a flight, you specify the flight number you wish to travel on.
- **Flight Tracking:** The flight number is used to track the status of a flight, such as delays or cancellations.
- **Airport Operations:** The flight number is used in various airport operations, like check-in and boarding.

Example:

If you want to travel from Riyadh to Dubai on an Emirates flight, you would search for the appropriate flight number on the airline's website or app. You might find a flight like "EK507.

- **Column(6): DIVERTED**

DIVERTED	count
0.000000	93224
1.000000	213



Out of the listed flights, 99.77% (93,224 flights) reached their original destination without being diverted, indicating that these flights did not encounter any exceptional circumstances requiring a route change. On the other hand, 0.23% (213 flights) were diverted to another airport or alternative destination due to reasons that may include bad weather, technical issues, or other emergencies.

DIVERTED is a term used when an aircraft is forced to land at an airport other than its planned DESTINATION_AIRPORT due to unexpected circumstances. These circumstances can include:

- **Weather:** Severe weather conditions like storms, fog, or snow can make it unsafe for an aircraft to continue to its intended destination.
- **Mechanical Issues**: Technical problems with the aircraft can necessitate an emergency landing at the nearest suitable airport.
- **Medical Emergencies**: If a passenger or crew member requires immediate medical attention, the flight may be diverted to an airport with appropriate medical facilities.
- **Security Threats:** In the event of security concerns, a flight may be diverted for safety reasons.

Key points about DIVERTED flights:

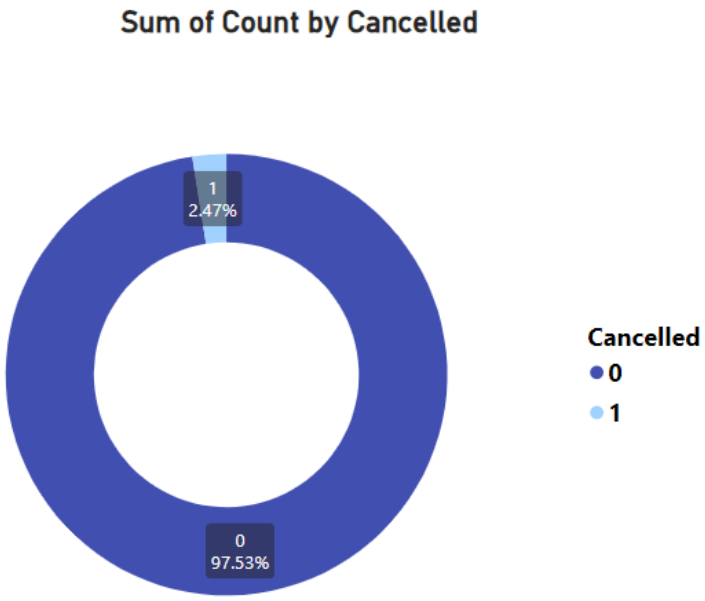
- **Unexpected Landing**: It's an unplanned landing at an airport other than the originally scheduled one.
- **Reasons:** Diversions are typically due to weather, mechanical issues, medical emergencies, or security threats.
- **Passenger Disruptions:** Diversions can cause significant inconvenience and delays for passengers.
- **Fuel Shortage:** Sometimes, a flight may be diverted to a nearby airport due to fuel shortages resulting from long delays or unforeseen circumstances.
- **Airport Problems**: Issues at the destination airport, such as a closed runway or heavy air traffic, may necessitate a diversion.
- **Airline Responsibilities:** Airlines are generally responsible for arranging transportation and accommodations for passengers affected by diversions.

Example:

- **A flight from New York to Los Angeles may be DIVERTED to Denver due to severe weather conditions in California.**

- **Column(7):Cancelled**

CANCELLED	count
0.000000	91133
1.000000	2304



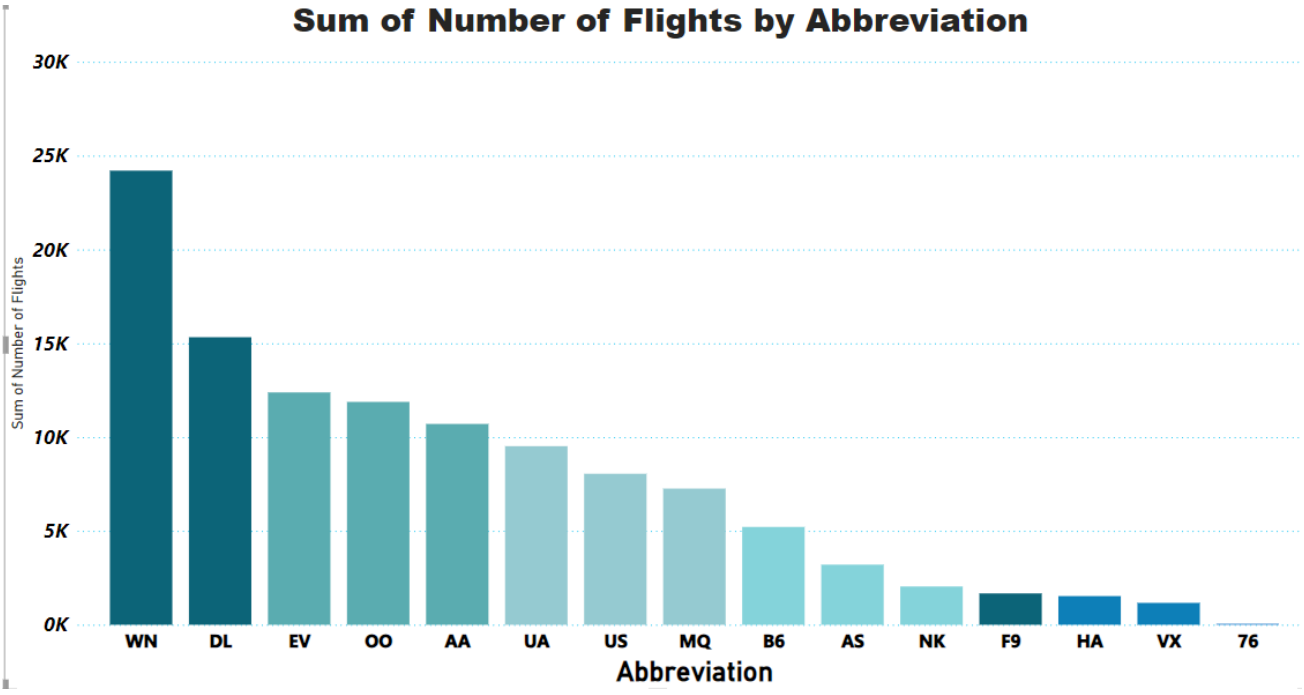
Out of a total of 93,438 flights, 97.53% (91,133 flights) were not cancelled, meaning the vast majority operated as scheduled. However, 2.47% (2,304 flights) were cancelled.

Flight cancellations can happen for various reasons, including severe weather conditions, mechanical issues, operational constraints like crew shortages, or logistical problems at the airport. Airlines prioritize safety and operational efficiency, so cancellations, though inconvenient, are often necessary to avoid risks or ensure smooth future operations.

Categorical Columns:

- **Column(1): AIRLINE**

Abbreviation	Airline Name	Number of Flights
WN	Southwest Airlines	24203
DL	Delta Air Lines	15325
EV	ExpressJet Airlines	12383
OO	SkyWest Airlines	11878
AA	American Airlines	10703
UA	United Airlines	9518
US	US Airways (now part of American Airlines)	8048
MQ	Envoy Air (subsidiary of American Airlines)	7257
B6	JetBlue Airways	5213
AS	Alaska Airlines	3204
NK	Spirit Airlines	2040
F9	Frontier Airlines	1671
HA	Hawaiian Airlines	1528
VX	Virgin America (merged with Alaska Airlines)	1166
76	(This may be an error in the data, as "76" is not recognized as an airline abbreviation)	1



These hypotheses suggest a mix of major airlines dominating the market and regional carriers fulfilling essential connectivity roles. The variation in percentages reflects differences in airline sizes, operational scopes, and market strategies.

In the dataset, Southwest Airlines (WN) leads with the highest flight count, accounting for approximately 21.14% of all flights. Delta Air Lines (DL) follows with a significant share of 14.08%. Envoy Air (EV) and SkyWest Airlines (OO) are also prominent, representing 10.60% and 10.22% of the flights, respectively. American Airlines (AA) holds a notable 9.32% of the total flights, while United Airlines (UA) contributes 8.37%. US Airways (US) covers 7.05% of the flights, and American Eagle (MQ) accounts for 6.25%. JetBlue Airways (B6) has a 4.60% share, with Alaska Airlines (AS) and Spirit Airlines (NK) contributing 2.83% and 1.87%, respectively. Frontier Airlines (F9) and Hawaiian Airlines (HA) have smaller shares at 1.40% and 1.35%, respectively, with Virgin America (VX) having the smallest share at 0.99%.

1. **Market Dominance:**
    - **Southwest Airlines (WN):** The highest percentage of flights suggests that Southwest Airlines is a major player in the market, possibly with a high volume of routes or flights.
    - **Delta Air Lines (DL):** The second-highest percentage indicates Delta's significant presence and operational scale, likely offering a broad range of domestic and international flights.
2. **Regional Operations:**
    - **Envoy Air (EV) and SkyWest Airlines (OO):** These regional carriers have a substantial share, indicating their importance in connecting smaller markets and regional airports to major hubs.
3. **Hub and Spoke Model:**
    - **American Airlines (AA) and United Airlines (UA):** Their large percentages might reflect extensive hub-and-spoke networks, serving as key airlines for major travel routes and hub connections.
4. **Budget Airlines:**
    - **JetBlue Airways (B6), Spirit Airlines (NK), Frontier Airlines (F9**): The presence of budget airlines in the data could indicate a growing trend towards low-cost travel options, catering to price-sensitive travelers.
5. **Specialized Services:**
    - **Alaska Airlines (AS) and Hawaiian Airlines (HA):** These airlines may serve specific regional markets (e.g., the West Coast and Hawaii), which could explain their lower, but still significant, percentages.
6. **Historical Context:**

- **Virgin America (VX**): The smaller percentage might be influenced by its historical operational status and recent merger with Alaska Airlines, affecting its overall flight volume.

Features of Southwest Airlines:

- **Low Prices:** Southwest Airlines is known for offering relatively competitive prices, making it a popular choice for travelers seeking cost-effective travel options.
- **Baggage Policy**: Southwest offers a distinctive baggage policy allowing passengers to check two bags for free, which differs from many other airlines that charge for checked baggage.
- **No Change Fees:** Southwest has a flexible policy regarding booking changes, with no additional fees for modifying or canceling reservations.
- **Customer Service**: Southwest is known for its good customer service, focusing on providing a positive travel experience.

Conclusion:

Southwest Airlines is generally not considered an expensive airline. On the contrary, it is known for offering travel options at lower prices compared to traditional airlines.

- **Column(2): TAIL_NUMBER**

TAIL_NUMBER is a specific sequence of letters and numbers used to identify a particular aircraft. It's like a license plate for a car, but for airplanes.

Key points about TAIL_NUMBER:

- **Uniqueness**: Each aircraft has its own unique TAIL_NUMBER.
- **Identification:** It's used to distinguish one aircraft from another, especially in air traffic control and aviation records.
- **Structure:** The exact format of a TAIL_NUMBER can vary depending on the country or region, but it typically includes a combination of letters and numbers.
- **Registration:** The TAIL_NUMBER is linked to the aircraft's registration information, which includes the owner, operator, and other relevant details.

Example:

- **N12345:** A common format for US-registered aircraft.

Why is it important?

- **Safety**: It helps prevent accidents by ensuring that air traffic controllers can accurately identify and track each aircraft.

- **Regulation:** It's essential for regulatory purposes, such as tracking aircraft maintenance and ownership.
- **Identification:** It's a crucial tool for identifying aircraft in case of emergencies or investigations.

- **Column(3): ORIGIN_AIRPORT**

ORIGIN_AIRPORT is the airport from which an aircraft departs on a specific flight. It's the starting point of the journey.

Key points about ORIGIN_AIRPORT:

- **Departure:** It's the airport where the flight begins its journey.
- **Location:** It's the physical location where the aircraft takes off.
- **Identification**: It's identified by its unique airport code (e.g., JFK for John F. Kennedy International Airport).
- **Data:** It's a common data point in flight information, travel itineraries, and aviation databases.

Example:

- **JFK: If the ORIGIN_AIRPORT for a flight is JFK, it means the flight departed from John F. Kennedy International Airport.**

Why is it important?

- **Travel Planning**: It's crucial for travelers to know the ORIGIN_AIRPORT to plan their trip, including transportation to the airport and check-in procedures.
- **Flight Information:** It's a fundamental piece of information in flight schedules, itineraries, and real-time flight tracking.
- **Aviation Data: It's used in various aviation databases and analytics for studying flight patterns, passenger traffic, and airport operations.**

- **Column(4): DESTINATION_AIRPORT**

DESTINATION_AIRPORT is the airport where an aircraft is scheduled to arrive at the end of a flight. It's the final destination of the journey.

Key points about DESTINATION_AIRPORT:

- **Arrival:** It's the airport where the flight will land.
- **Location**: It's the physical location where the aircraft will come to a stop.

- **Identification:** It's identified by its unique airport code (e.g., LAX for Los Angeles International Airport).
- **Data:** It's a common data point in flight information, travel itineraries, and aviation databases.

Example:

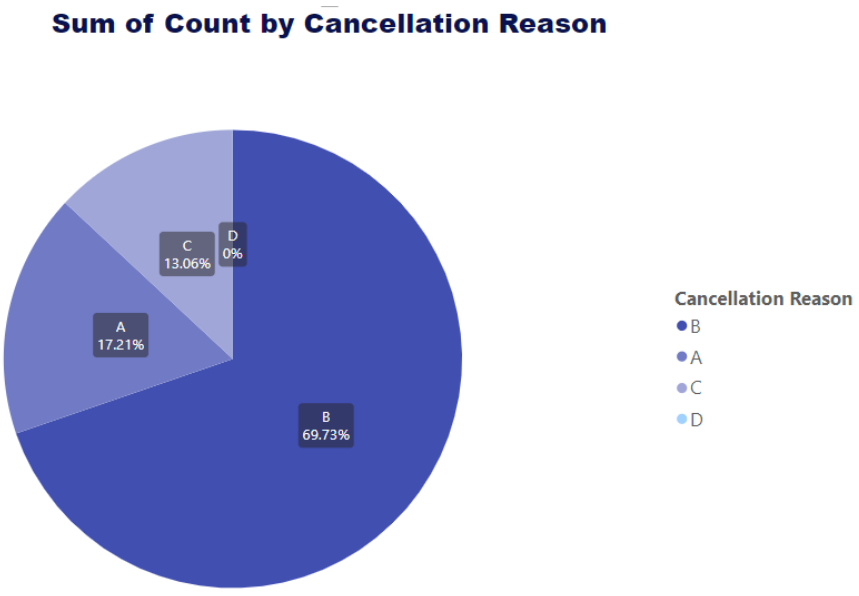
- **LAX:** If the DESTINATION_AIRPORT for a flight is LAX, it means the flight will arrive at Los Angeles International Airport.

Why is it important?

- **Travel Planning:** It's crucial for travelers to know the DESTINATION_AIRPORT to plan their onward travel, accommodation, and activities.
- **Flight Information:** It's a fundamental piece of information in flight schedules, itineraries, and real-time flight tracking.
- **Aviation Data:** It's used in various aviation databases and analytics for studying flight patterns, passenger traffic, and airport operations.

- **Column(5):Cancellation Reason**

CANCELLATION_REASON	
B	28260
A	6974
C	5291
D	2



1.A (Weather-related):

- **Description:** These cancellations were due to weather-related issues. Severe weather conditions such as thunderstorms, snow, fog, or hurricanes can prevent safe flight operations, leading to flight cancellations for the safety of passengers and crew.

2. **B (Airline-related):**

- **Description:** The largest proportion of cancellations were due to airline-related issues. These could include mechanical failures, staffing problems, or other operational challenges within the airline, such as overbooking or logistical errors.

3. **C (Security):**

- **Description:** Security-related cancellations, though rare, occur due to safety threats or security concerns at airports, such as suspicious activities, bomb threats, or government-mandated security measures that prevent flights from departing.

4. **D (Other/Unspecified):**

- **Description:** These cancellations are due to other or unspecified reasons, which may include unexpected operational disruptions not covered by weather, airline issues, or security concerns. This category is extremely rare and represents just 2 cases in the dataset.

▪ Continuous Numerical Columns:

Explanation of Flight Data Points:

1. **SCHEDULED_DEPARTURE:**
   - **Description:** The planned time for an aircraft to depart from the origin airport.
   - **Importance**: Indicates the scheduled start time of the journey according to the flight timetable.
2. **DEPARTURE_TIME:**
   - **Description:** The actual time when the aircraft departs.
   - **Importance:** Shows how closely the flight adhered to its scheduled departure time.
3. **DEPARTURE_DELAY:**
   - **Description:** The delay in departure, measured in minutes.
   - **Importance**: Reflects the extent of delay from the scheduled departure time.
4. **TAXI_OUT:**
   - **Description:** The time taken for the aircraft to taxi from the gate to the runway.
   - **Importance:** Indicates the efficiency of airport operations in managing aircraft movements.
5. **WHEELS_OFF:**
   - **Description**: The time when the aircraft leaves the runway and becomes airborne.

- **Importance:** Marks the actual start of the flight phase**.**

6. **SCHEDULED_TIME:**
   - **Description:** The scheduled duration of the flight from takeoff to landing.
   - **Importance:** Used to estimate the expected flight duration according to the schedule**.**
7. **ELAPSED_TIME:**
   - **Description:** The actual time taken for the flight from takeoff to landing.
   - **Importance:** Shows the difference between the scheduled flight time and the actual flight duration.
8. **AIR_TIME:**
   - **Description**: The time the aircraft spends in the air, excluding taxiing.
   - **Importance:** Indicates the actual airborne time, excluding any time spent taxiing on the ground.
9. **DISTANCE:**
   - **Description**: The total distance traveled during the flight.
   - **Importance:** Helps determine the length of the flight and the fuel consumption.
10. **WHEELS_ON:**
    - **Description:** The time when the aircraft's wheels touch down on the runway.
    - **Importance:** Marks the end of the flight phase and the beginning of the landing process.
11. **TAXI_IN:**
    - **Description:** The time taken for the aircraft to taxi from the runway to the gate after landing.
    - **Importance:** Reflects the efficiency of airport operations in managing post-landing aircraft movements.
12. **SCHEDULED_ARRIVAL:**
    - **Description:** The planned time for the aircraft to arrive at the destination airport.
    - **Importance:** Indicates the expected arrival time based on the flight schedule**.**
13. **ARRIVAL_TIME:**
    - **Description:** The actual time when the aircraft arrives at the destination airport.
    - **Importance**: Shows how closely the flight adhered to its scheduled arrival time.
14. **ARRIVAL_DELAY:**
    - **Description:** The delay in arrival, measured in minutes.
    - **Importance:** Reflects the extent of delay from the scheduled arrival time.
15. **AIR_SYSTEM_DELAY:**
    - **Description:** Delay caused by air traffic control system issues.
    - **Importance:** Indicates delays due to air traffic management problems.
16. **SECURITY_DELAY:**
    - **Description:** Delay caused by security-related issues.
    - **Importance:** Reflects delays due to security measures or concerns.
17. **AIRLINE_DELAY:**
    - **Description**: Delay caused by issues within the airline.
    - **Importance:** Indicates delays due to mechanical failures, staffing problems, or other operational challenges.
18. **LATE_AIRCRAFT_DELAY:**

- **Description:** Delay caused by the late arrival of a previous aircraft.
- **Importance**: Shows delays resulting from earlier flights affecting the current schedule.

19. **WEATHER_DELAY:**
    - **Description:** Delay caused by weather conditions.
    - **Importance:** Reflects delays due to adverse weather such as storms, fog, or snow.

Describe(Numerical Column)

	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	WHEELS_OFF	SCHEDULED_TIME	ELAPSED_TIME	AIR_TIME	DISTANCE	WHEELS_ON	TAXI_IN	ARRIVAL_TIME	ARRIVAL_DELAY
count	1009060.000000	1009060.000000	1008346.000000	1008346.000000	1048573.000000	1005504.000000	1005504.000000	1048575.000000	1007279.000000	1007279.000000	1007279.000000	1005504.000000
mean	1333.704944	11.334851	16.653802	1357.381529	140.252598	136.938118	112.747698	803.407664	1485.931935	7.549438	1492.203594	7.612191
std	482.741534	39.223721	10.070062	483.035110	74.634578	73.948180	71.869516	594.236215	503.351462	6.352526	507.109030	42.093672
min	1.000000	-61.000000	1.000000	1.000000	20.000000	15.000000	7.000000	31.000000	1.000000	1.000000	1.000000	-82.000000
25%	928.000000	-5.000000	11.000000	944.000000	85.000000	82.000000	60.000000	368.000000	1110.000000	4.000000	1115.000000	-12.000000
50%	1329.000000	-1.000000	14.000000	1342.000000	122.000000	119.000000	94.000000	641.000000	1516.000000	6.000000	1521.000000	-3.000000
75%	1731.000000	11.000000	19.000000	1745.000000	173.000000	169.000000	144.000000	1046.000000	1911.000000	9.000000	1917.000000	12.000000
max	2400.000000	1988.000000	225.000000	2400.000000	718.000000	766.000000	687.000000	4983.000000	2400.000000	202.000000	2400.000000	1971.000000

AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY
228528.000000	228528.000000	228528.000000	228528.000000	228528.000000
13.692554	0.057328	18.203577	22.921458	3.545277
25.524897	1.779647	46.323146	41.888498	23.611555
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000
4.000000	0.000000	2.000000	4.000000	0.000000
19.000000	0.000000	18.000000	29.000000	0.000000
830.000000	241.000000	1971.000000	1313.000000	1152.000000

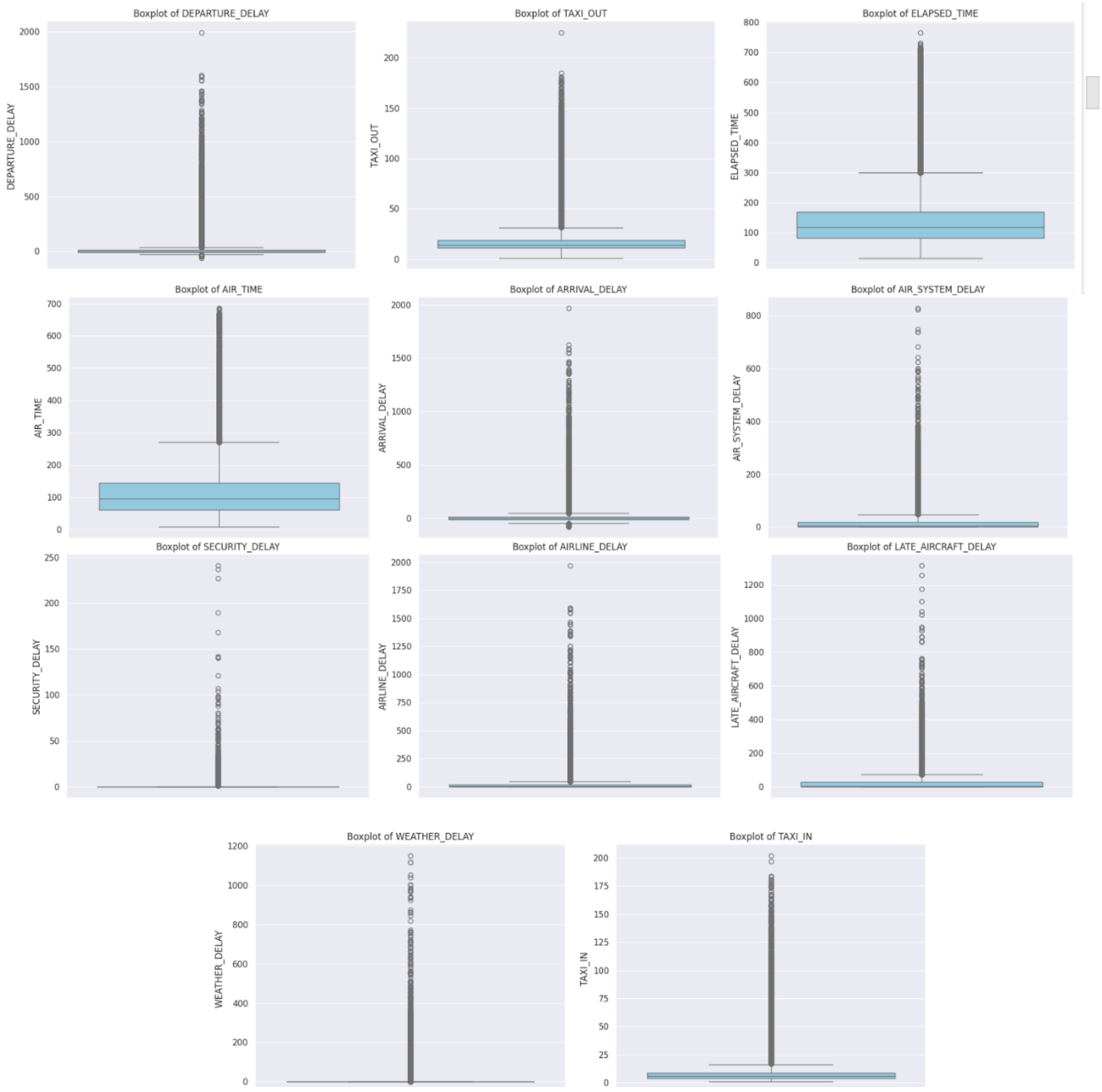
Outliers can be identified by looking at extreme values (like minimum and maximum) in comparison to the mean and quartile ranges. In your summary statistics, the columns with potential outliers are:

1. **DEPARTURE_DELAY**
    - **Mean:** 11.33 minutes
    - **Max:** 1,988 minutes
    - The maximum value (1,988 minutes, or about 33 hours) is significantly higher than the mean and even the 75th percentile (11 minutes), indicating potential outliers.

2. **TAXI_OUT**
    - **Mean**: 16.65 minutes
    - **Max:** 225 minutes The maximum taxi-out time (225 minutes, nearly 4 hours) seems to be far from the 75th percentile (19 minutes), indicating outliers.
3. **ELAPSED_TIME**
    - **Mean:** 136.94 minutes
    - **Max:** 766 minutes
    - The maximum elapsed time (766 minutes) is quite far from the mean, indicating possible outliers.
4. **AIR_TIME**
    - **Mean:** 112.75 minutes
    - **Max**: 687 minutes
    - The maximum air time (687 minutes, or more than 11 hours) is much larger than the mean (112 minutes), suggesting outliers.
5. **ARRIVAL_DELAY**
    - **Mean:** 7.61 minutes
    - **Max:** 1,971 minutes
    - The maximum arrival delay (1,971 minutes) is a major outlier compared to the 75th percentile (12 minutes).
6. **AIR_SYSTEM_DELAY, SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY**
    - Maximum values for these delay types are extremely high compared to their means and 75th percentiles (maximum weather delay of 1,152 minutes), suggesting significant outliers.
7. **TAXI_IN**
    - **Mean:** 7.55 minutes
    - **Max:** 202 minutes
    - The maximum value (202 minutes) is significantly larger than the mean and 75th percentile (9 minutes), indicating outliers.

These columns show high deviation between the maximum values and the mean or quartiles, making them likely candidates for containing outliers.

Visualization for Columns that's Have Outliers



Possible Reasons for Outliers in Flight Data:

1. **DEPARTURE_DELAY**
    - **Reason for Outliers:**
        - **Severe Weather:** Extreme weather conditions can cause significant delays, leading to very high departure delays.

        - **Air Traffic Control Delays:** Congestion or delays in air traffic control can significantly push back departure times.
        - **Operational Issues:** Mechanical failures or other airline-specific issues might lead to substantial delays.
    - **Example:** A flight might experience a departure delay of over 33 hours due to an unexpected mechanical problem or severe weather that forces the plane to wait for clearance or repairs.

2. **TAXI_OUT**
    - **Reason for Outliers:**
        - **Airport Congestion:** High traffic at the airport can cause extended taxi times, especially during peak hours.
        - **Runway or Taxiway Closures:** Temporary closures or maintenance work can result in prolonged taxi times.
        - **Operational Delays:** Issues with airport ground services or long queues for takeoff can contribute to extended taxiing.
    - **Example:** A flight might experience nearly 4 hours of taxi time due to heavy traffic at a major airport or unexpected runway closures.

3. **ELAPSED_TIME**
    - **Reason for Outliers:**
        - **Flight Path Deviations:** Aircraft may take longer routes due to air traffic control instructions or weather conditions, significantly increasing the flight time.
        - **Extended Holding Patterns:** Flights might spend extended periods in holding patterns before landing, adding to the elapsed time.
        - **Technical Issues**: Problems with the aircraft or crew can lead to longer flight times.
    - **Example: A flight taking 766 minutes (over 12 hours) might be rerouted due to severe weather, resulting in an unusually long journey.**

4. **AIR_TIME**
    - **Reason for Outliers:**
        - **Extended Routes:** Flights might have longer air times if they are rerouted or encounter air traffic control delays.
        - **Operational Issues:** Mechanical issues or extended periods of holding in the air can extend the time the aircraft spends airborne.

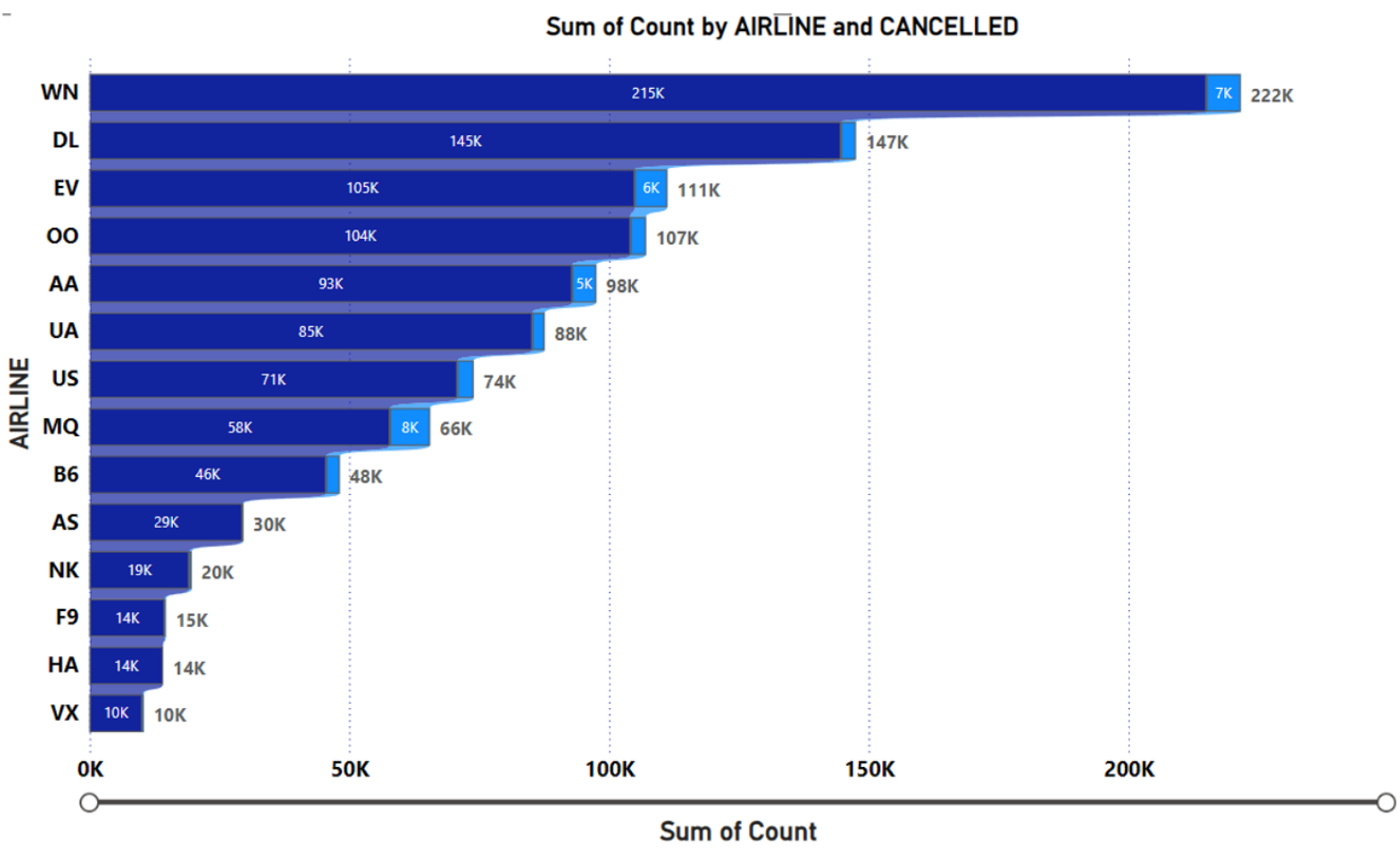
        - **High Altitude Holding:** Unplanned high-altitude holding patterns can add significant time to the airborne portion of the flight.
    - **Example:** A flight might have an air time of over 11 hours if it had to detour significantly from its planned route.

5. **ARRIVAL_DELAY**
    - **Reason for Outliers:**
        - **Severe Weather at Destination**: Adverse weather conditions at the arrival airport can cause significant delays in landing.
        - **Airport Congestion:** High traffic at the destination airport can lead to long waits for landing and gate availability.
        - **Operational Issues**: Similar to departure delays, issues at the arrival airport or with the aircraft can result in extended delays.
    - **Example:** A flight might experience a 1,971-minute delay if it's rerouted or experiences severe weather conditions at the destination.
6. **AIR_SYSTEM_DELAY, SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY**
    - **Reason for Outliers:**
        - **System Failures:** Failures or delays in the air traffic control system can cause significant delays.
        - **Security Threats:** Unexpected security threats or issues can result in extended delays.
        - **Operational Challenges:** Internal airline problems or late arrivals of previous flights can lead to prolonged delays.
        - **Weather Conditions:** Extreme weather conditions can cause substantial delays across multiple aspects of flight operations.
    - **Example:** A 1,152-minute weather delay might occur due to a major storm affecting multiple flights and airport operations, leading to extended delays.

These outliers are often the result of exceptional circumstances or operational challenges that significantly deviate from normal flight operations.

General Analysis:

1.Airline and Cancelled



Data Interpretation:

The data shows the number of canceled and non-canceled flights for each airline, and several insights can be drawn:

1. **Higher Cancellation Rates in Large Airlines:**
    - Example: WN (Southwest Airlines): It has 214,980 canceled flights, which represents 20.49% of the total flights. In contrast, it has 6,606 non-canceled flights, representing only 0.63%.
        - This indicates that WN operates a large number of flights but also experiences a high cancellation rate.
2. **Smaller Airlines May Face Fewer Cancellations:**

- **Example: MQ (Air Wisconsin): It has 57,786 canceled flights, representing 5.51% of total flights, while it has 7,727 non-canceled flights (0.737%).**
    - This suggests that MQ faces fewer cancellations, possibly due to its smaller scale of operations compared to larger airlines.

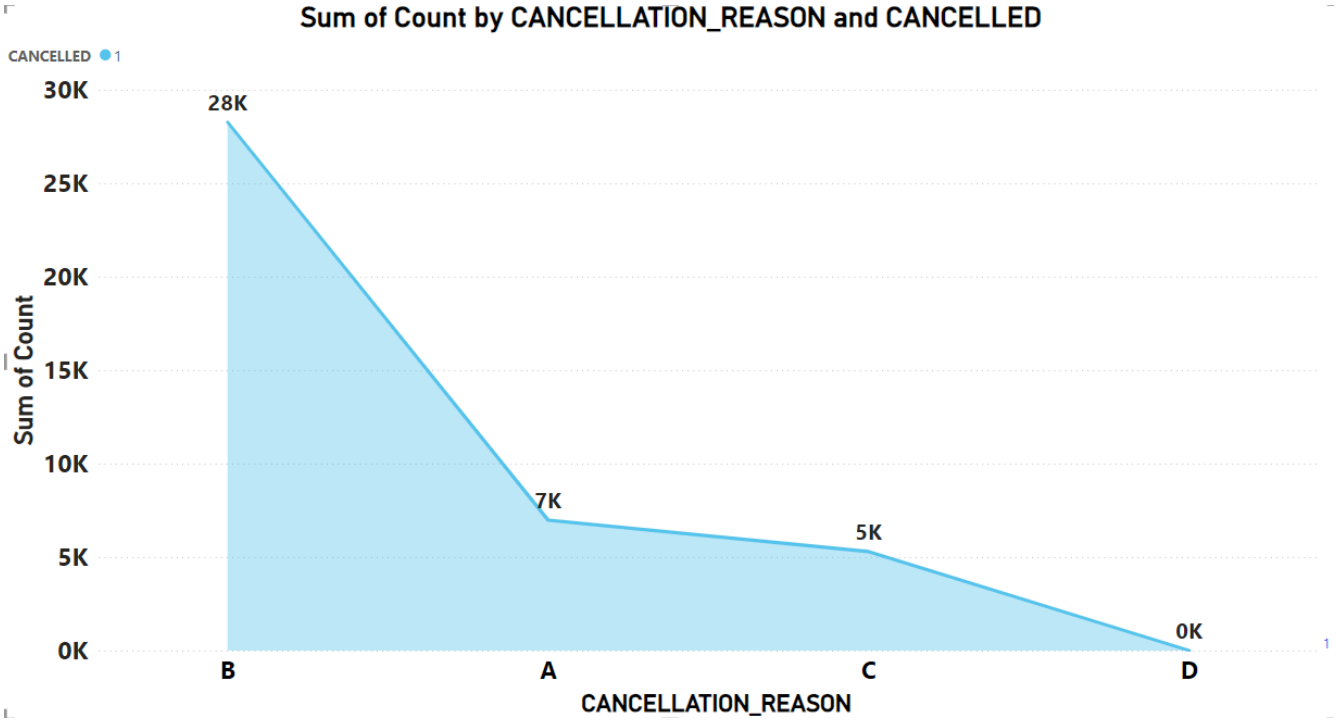
3. **Impact of Operational or Seasonal Factors:**
    - Airlines like DL (Delta Airlines) and UA (United Airlines) have relatively high cancellations (144,625 and 85,203 respectively). These cancellations might be related to operational factors like weather or regulatory issues.
4. **Comparison Between Major and Regional Airlines:**
    - Major airlines like DL and UA show different cancellation patterns compared to regional airlines like EV. This could be due to differences in network complexity or operational challenges.
5. **Operational Efficiency of Specific Airlines:**
    - Some airlines like HA (Hawaiian Airlines) and AS (Alaska Airlines) have relatively lower cancellation rates (14,100 and 29,417 cancellations respectively). This could reflect better operational efficiency or more effective management of flight schedules.

Conclusion:

- WN and DL are large airlines with a significant number of cancellations, which is expected given the scale of their operations.
- Regional airlines like MQ and EV may experience fewer absolute cancellations but could have higher cancellation rates relative to their total flights.
- The data suggests a correlation between airline size and the number of cancellations, with operational and seasonal factors playing a significant role in determining cancellation rates.

2.Cancelation Reason and Cancelled

CANCELLED	CANCELLATION_REASON	count
1	B	28260
	A	6974
	C	5291
	D	2



The analysis of flight cancellations reveals that weather conditions were the leading cause, accounting for 28,260 cancellations, which represents approximately 2.69% of all flights. Carrier-related issues were the second most frequent cause, with 6,974 cancellations, making up about 0.66% of the total flights. Issues related to the National Airspace System (NAS) were responsible for 5,291 cancellations, or roughly 0.50% of all flights. Security concerns were the least common reason, with only 2 flights canceled, which constitutes approximately 0.0002% of the total flights. This distribution underscores that weather-

related factors are the predominant cause of flight cancellations, while security concerns contribute minimally.

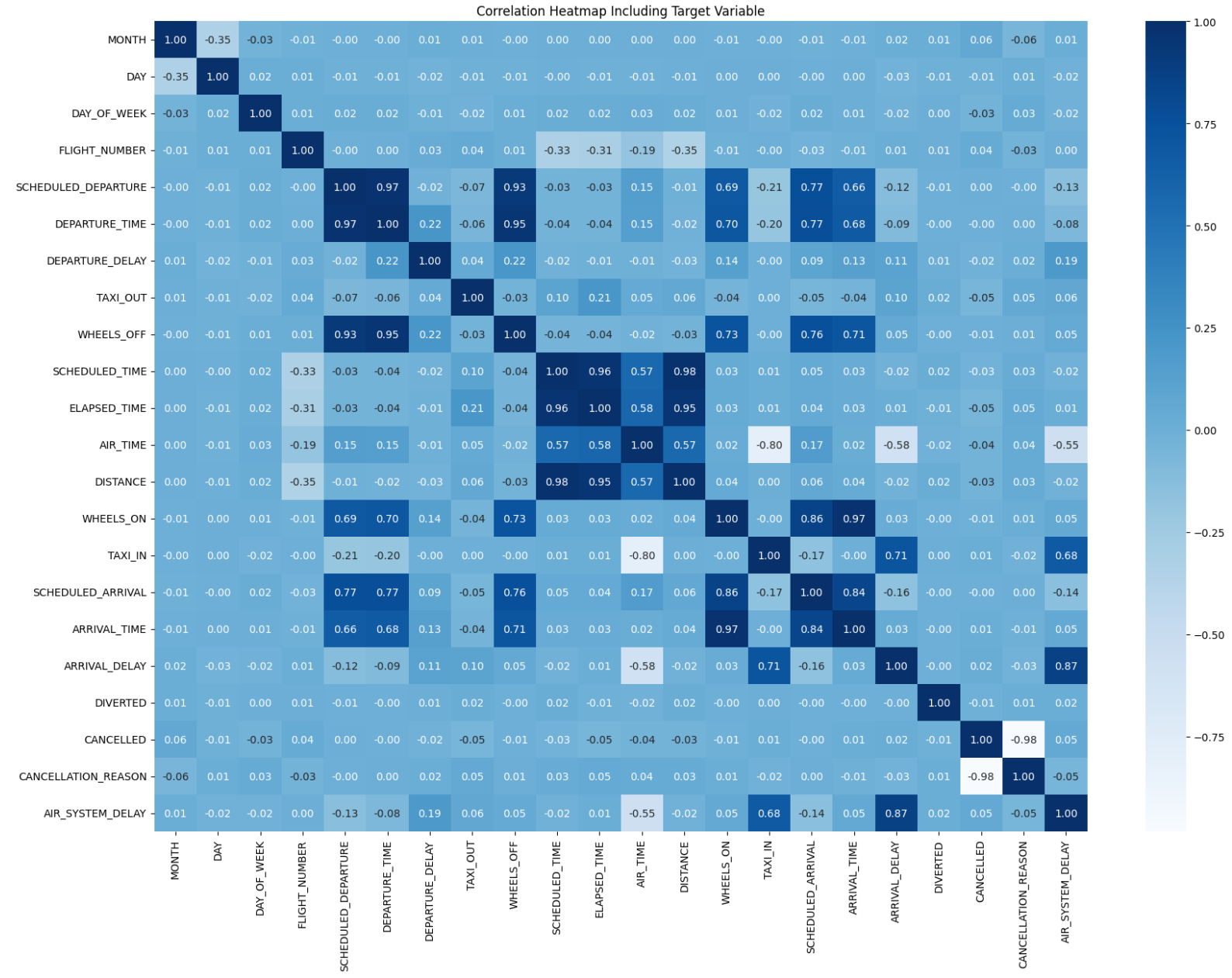
Here's a breakdown of why each reason might lead to flight cancellations:

1. **Weather (B):**
    - Why: Severe weather conditions, such as storms, hurricanes, blizzards, or heavy rain, can create hazardous flying conditions. Poor visibility, strong winds, or icy runways can make it unsafe for flights to take off or land. Airlines prioritize safety, so flights are canceled to avoid putting passengers and crew at risk.

2. **Carrier (A):**
    - Why: Mechanical issues, maintenance problems, or crew shortages within the airline can lead to cancellations. For instance, if an aircraft has a critical malfunction or if there's a lack of available pilots or flight attendants, the airline may cancel the flight to ensure that safety standards are met and to address the issue properly.
3. **National Airspace System (NAS) (C):**
    - Why: The National Airspace System manages air traffic and airspace usage. Cancellations can occur due to air traffic control delays, congestion in airspace, or other operational challenges within the airspace system. For example, if there are significant delays in air traffic control or if airspace is closed due to unforeseen circumstances, flights may be canceled to manage the flow of air traffic and avoid conflicts.
4. **Security (D):**
    - Why: Security-related cancellations are typically due to threats or incidents that require additional security measures. This could involve heightened security alerts, suspicious activities, or security breaches. Such situations may require immediate action, including flight cancellations, to ensure passenger safety and address any security concerns.

Understanding these reasons helps in managing and mitigating the impact of cancellations and improving overall airline operations and passenger experience.

3.Correlation Heatmap Including Target Variable



Analyzing the Correlation Heatmap

Understanding the Visual The provided heatmap is a powerful tool for visualizing the relationships between different variables within a dataset. Each square in the heatmap represents the correlation between two variables, with the color intensity indicating the strength and direction of the correlation.

Strong Positive Correlations:

- Flight Number and Scheduled Departure: These two variables have a very strong positive correlation, which makes sense as flights are typically assigned a unique number before their scheduled departure time.
- Scheduled Departure and Departure Time: Similarly, there's a strong positive correlation between these two variables, indicating that flights generally depart very close to their scheduled times.
- Elapsed Time and Air Time: These variables show a strong positive correlation, as the total flight time is largely determined by the time spent in the air.
- Distance and Air Time: A strong positive correlation exists between distance and air time, reflecting the direct relationship between the distance traveled and the time spent flying.

Strong Negative Correlations:

- Arrival Delay and Departure Delay: A strong negative correlation suggests that flights that depart earlier are less likely to arrive late, and vice versa.
- Diverted and Cancelled: These variables have a strong negative correlation, indicating that flights are less likely to be diverted if they are not cancelled.

Moderate Correlations:

- Departure Delay and Taxi Out: There's a moderate positive correlation, suggesting that longer taxi times might contribute to flight delays.
- Arrival Delay and Taxi In: A similar moderate positive correlation exists between arrival delays and taxi-in times.

Additional Insights

- Clustering: The heatmap also reveals clusters of variables that are highly correlated with each other. For example, the variables related to flight scheduling and departure times form a cluster.
- Outliers: Some variables, such as "Diverted" and "Cancelled," have relatively low correlations with other variables, indicating that they might be less influenced by the other factors in the dataset.

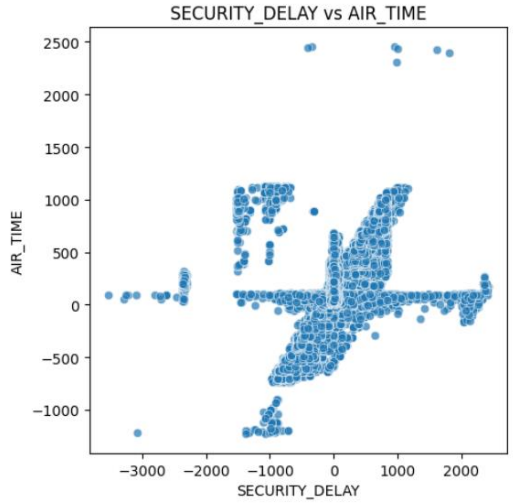
Correlation Between Predictive Columns and Other Columns

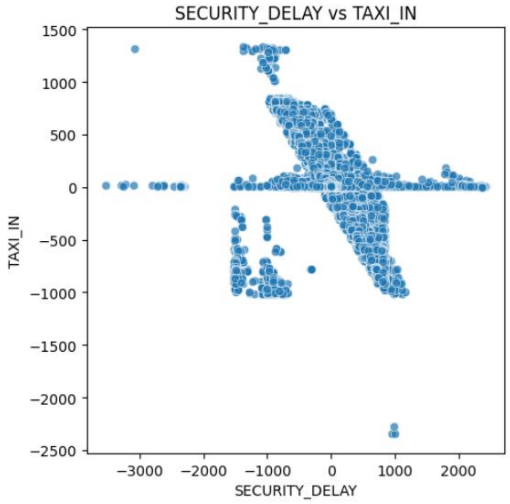
The relationships between predictive columns (such as SECURITY_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, LATE_AIRCRAFT_DELAY)and other columns in the dataset illustrate how these predictive columns impact various aspects of flight operations.

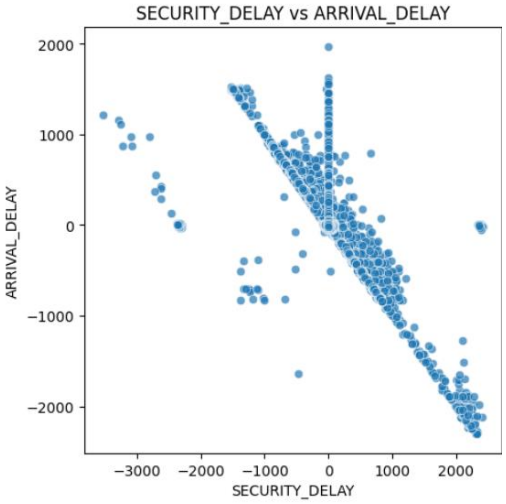
Here’s an interpretation of the main correlations between predictive columns and other columns:

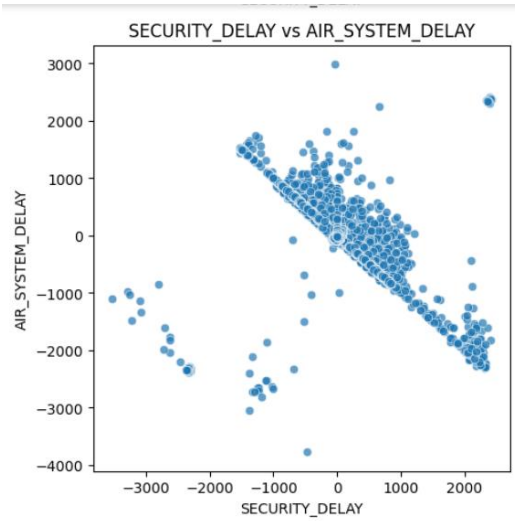
- **SECURITY_DELAY**

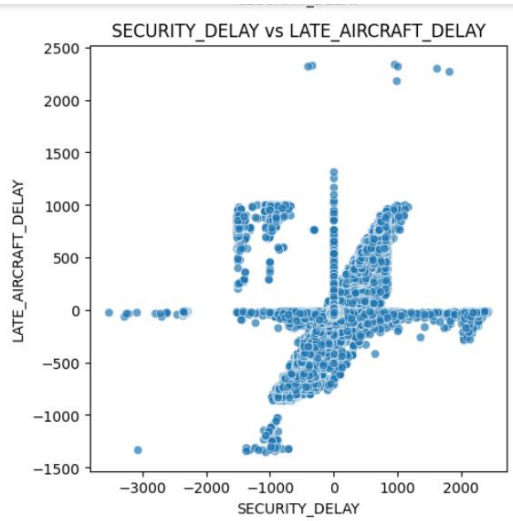
```
Strong correlations with SECURITY_DELAY:
AIR_TIME                 0.569277
TAXI_IN                 -0.699357
ARRIVAL_DELAY           -0.852978
AIR_SYSTEM_DELAY        -0.651244
SECURITY_DELAY           1.000000
LATE_AIRCRAFT_DELAY      0.672701
WEATHER_DELAY           -0.798556
```

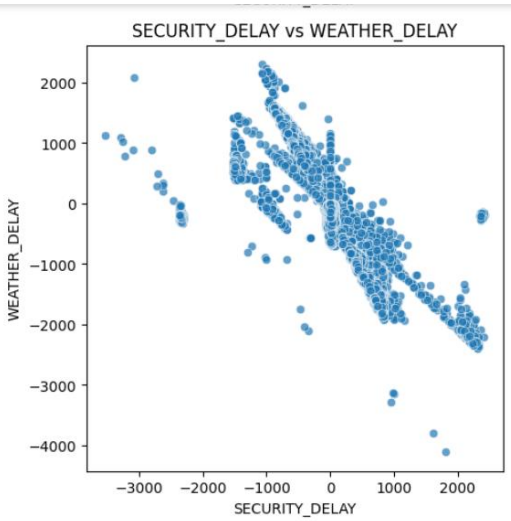










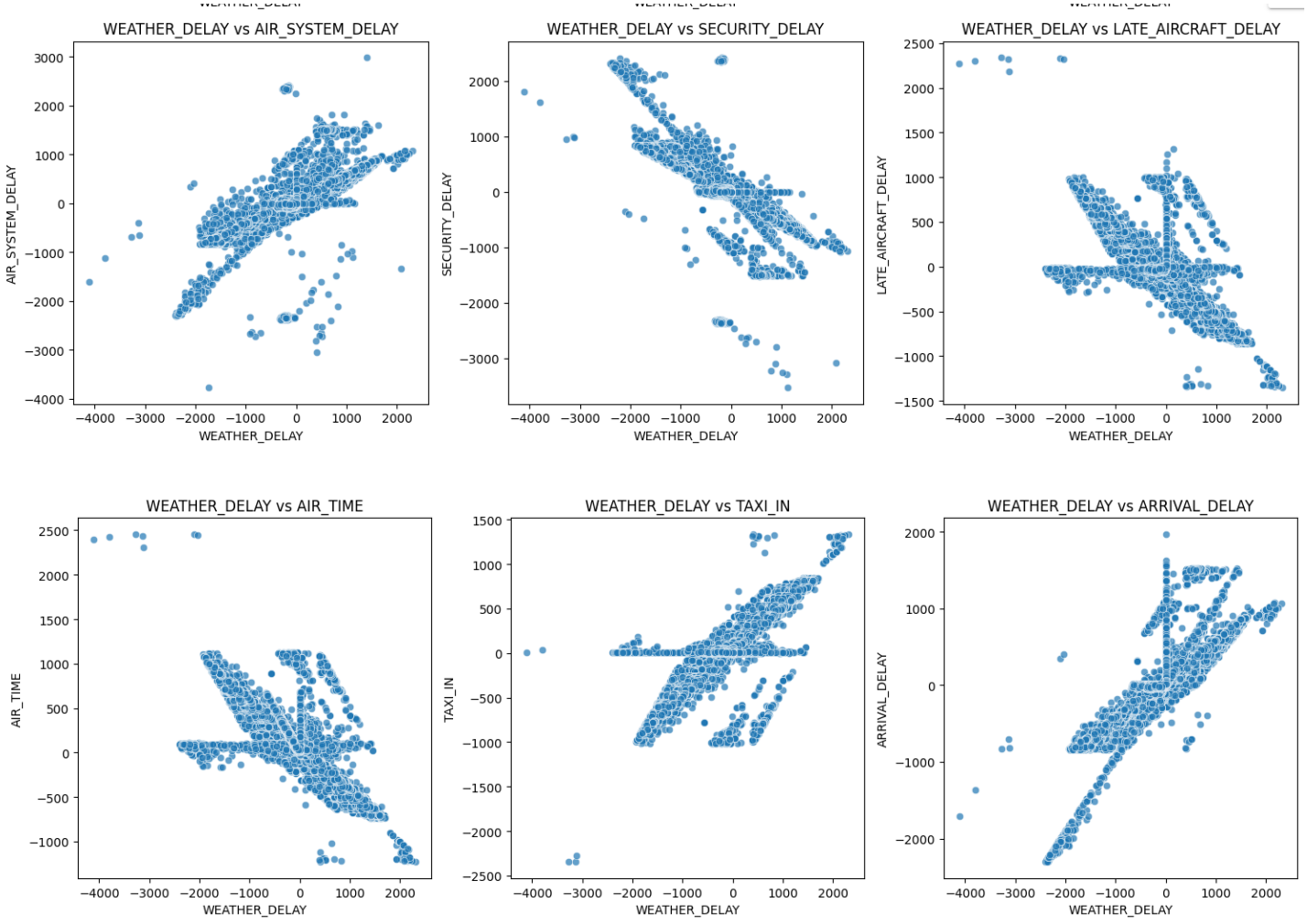


The SECURITY_DELAY column shows various degrees of correlation with other numerical features in the flight dataset. Here is an interpretation of these correlations:

- **AIR_TIME:** There is a moderate positive correlation with SECURITY_DELAY. As SECURITY_DELAY increases, AIR_TIME also tends to increase. This might suggest that longer security delays could be associated with increased air times, possibly due to operational adjustments or delays in boarding processes impacting flight schedules.
- **TAXI_IN:** There is a strong negative correlation with SECURITY_DELAY. As SECURITY_DELAY increases, TAXI_IN tends to decrease. This may indicate that longer security delays could lead to shorter taxi-in times, potentially due to changes in scheduling or reduced congestion at the airport as a result of delays.
- **ARRIVAL_DELAY:** There is a strong negative correlation with ARRIVAL_DELAY. An increase in SECURITY_DELAY is associated with a decrease in ARRIVAL_DELAY. This could suggest that longer security delays might be offset by adjustments in arrival schedules or operational changes that reduce the impact on arrival times.
- **AIR_SYSTEM_DELAY:** There is a moderate negative correlation with AIR_SYSTEM_DELAY. As SECURITY_DELAY increases, AIR_SYSTEM_DELAY tends to decrease. This might be due to adjustments in air traffic management or operational systems to accommodate delays, reducing the overall system delay.
- **LATE_AIRCRAFT_DELAY:** There is a moderate positive correlation with SECURITY_DELAY. Higher SECURITY_DELAY is associated with an increase in LATE_AIRCRAFT_DELAY. This could indicate that delays related to security measures contribute to increased delays in subsequent aircraft scheduling or operations.
- **WEATHER_DELAY:** There is a strong negative correlation with WEATHER_DELAY. As SECURITY_DELAY increases, WEATHER_DELAY tends to decrease. This might suggest that as security delays increase, weather-related delays might have less impact or be less frequent, possibly due to operational adjustments or different influencing factors on delays

- **WEATHER_DELAY**

```
Strong correlations with WEATHER_DELAY:
AIR_TIME                -0.843470
TAXI_IN                  0.857281
ARRIVAL_DELAY            0.878368
AIR_SYSTEM_DELAY         0.805164
SECURITY_DELAY          -0.798556
LATE_AIRCRAFT_DELAY     -0.777546
WEATHER_DELAY            1.000000
```

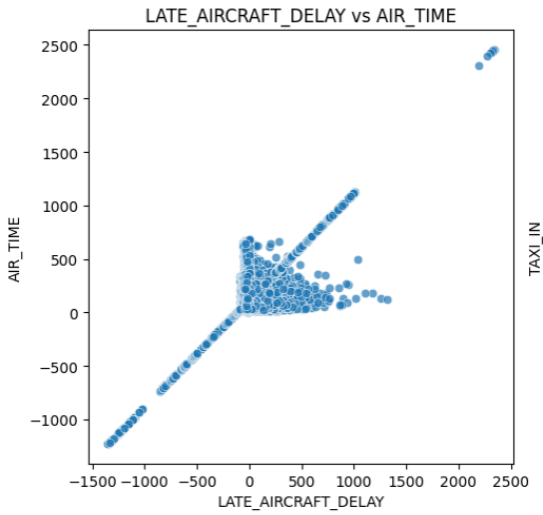


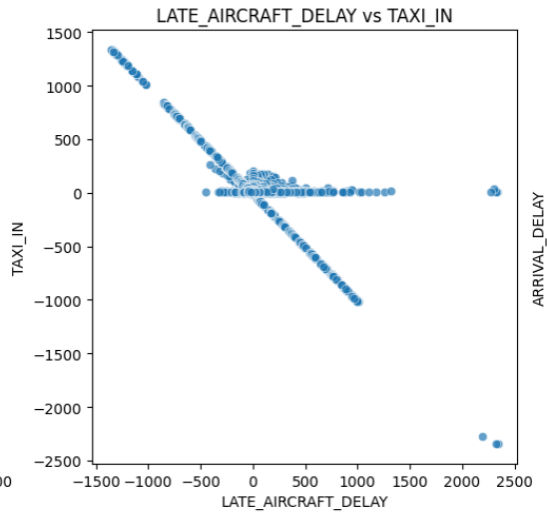
The WEATHER_DELAY column shows varying degrees of correlation with other numerical features in the flight dataset. The following correlations have been observed:

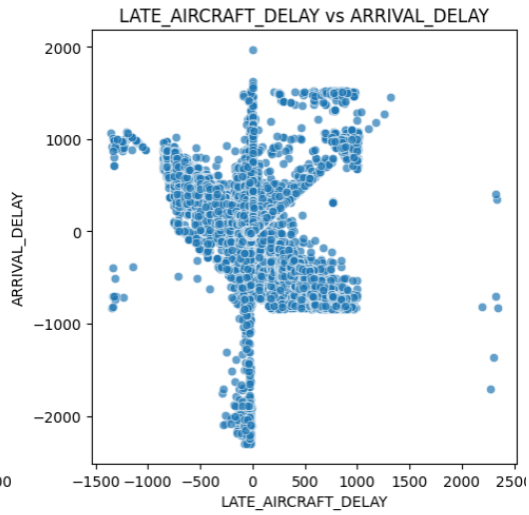
1. **AIR_TIME:**
    - **Interpretation:** A strong negative correlation with WEATHER_DELAY. This suggests that as WEATHER_DELAY increases, AIR_TIME tends to decrease. This might be due to weather-related delays affecting the duration of the flight, possibly leading to reduced flying time.
2. **TAXI_IN:**
    - **Interpretation:** A strong positive correlation with WEATHER_DELAY. This indicates that higher WEATHER_DELAY is associated with longer TAXI_IN times. Bad weather conditions might lead to extended taxi times due to operational delays and increased congestion.
3. **ARRIVAL_DELAY:**
    - **Interpretation:** A very strong positive correlation with WEATHER_DELAY. This suggests that as WEATHER_DELAY increases, ARRIVAL_DELAY also increases significantly. Severe weather conditions likely contribute to longer delays upon arrival.
4. **AIR_SYSTEM_DELAY:**
    - **Interpretation:** A strong positive correlation with WEATHER_DELAY. This suggests that weather delays are associated with higher AIR_SYSTEM_DELAY, possibly due to system disruptions caused by weather conditions**.**
5. **SECURITY_DELAY:**
    - **Interpretation:** A strong negative correlation with WEATHER_DELAY. This indicates that when weather delays are high, SECURITY_DELAY tends to be lower. It may imply that security delays are less influenced by weather and are relatively stable regardless of weather conditions.
6. **LATE_AIRCRAFT_DELAY:**
    - **Interpretation:** A strong negative correlation with WEATHER_DELAY. This suggests that higher WEATHER_DELAY is associated with a decrease in LATE_AIRCRAFT_DELAY. This might be because weather delays can sometimes lead to adjustments in scheduling that reduce the impact of late aircraft.
7. **WEATHER document DELAY:**
    - **Interpretation:** Perfect positive correlation, which is expected as it represents the same variable or a duplicate entry. It reinforces that the WEATHER document DELAY column is identical to WEATHER_DELAY.

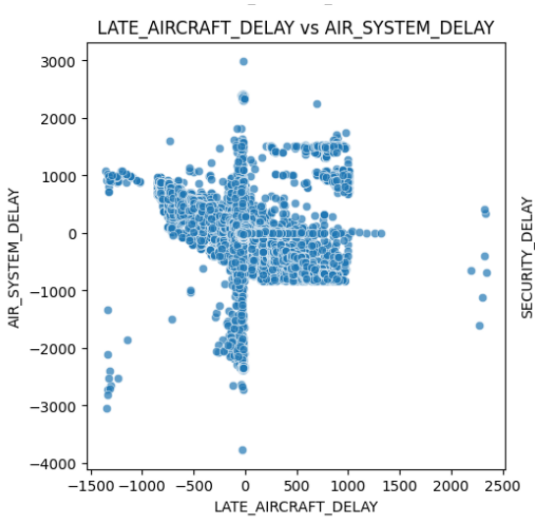
- **LATE_AIRCRAFT_DELAY**

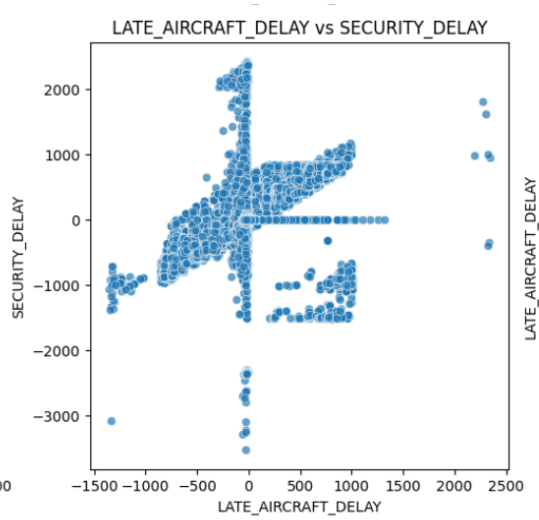
```
Strong correlations with LATE_AIRCRAFT_DELAY:
AIR_TIME                0.769362
TAXI_IN                -0.957326
ARRIVAL_DELAY          -0.615540
AIR_SYSTEM_DELAY       -0.640603
SECURITY_DELAY          0.672701
LATE_AIRCRAFT_DELAY     1.000000
WEATHER_DELAY          -0.777546
```

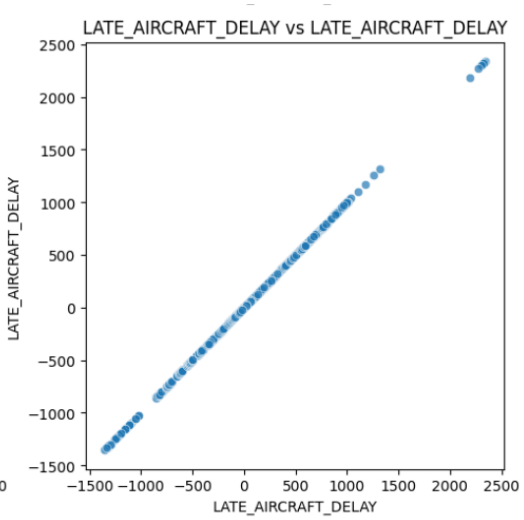












The LATE_AIRCRAFT_DELAY column exhibits various correlations with other numerical features in the flight dataset. Here is an interpretation of these correlations and potential reasons behind them:

1. **AIR_TIME:**
    - **Interpretation:** There is a strong positive correlation between AIR_TIME and LATE_AIRCRAFT_DELAY. This suggests that as LATE_AIRCRAFT_DELAY increases, AIR_TIME also tends to increase. A possible reason for this correlation could be that delays in aircraft arriving late can cause subsequent flights to have extended air times due to adjustments or congestion.
2. **TAXI_IN:**
    - **Interpretation:** This shows a very strong negative correlation with LATE_AIRCRAFT_DELAY. As LATE_AIRCRAFT_DELAY increases, TAXI_IN tends to decrease significantly. This may be due to the fact that when aircraft are late, there might be fewer opportunities or space available for them to taxi in, leading to a reduced taxi-in time due to operational constraints or scheduling adjustments.
3. **ARRIVAL_DELAY:**
    - **Interpretation:** There is a moderate negative correlation with ARRIVAL_DELAY. Higher LATE_AIRCRAFT_DELAY is associated with a decrease in ARRIVAL_DELAY. This might indicate that late arrivals are often adjusted in schedules or that subsequent delays are mitigated, leading to a reduction in arrival delay time.
4. **AIR_SYSTEM_DELAY:**
    - **Interpretation:** A moderate negative correlation with AIR_SYSTEM_DELAY. This suggests that as LATE_AIRCRAFT_DELAY increases, the AIR_SYSTEM_DELAY tends to decrease. This might be due to adjustments in air traffic management systems or operational changes to accommodate delays, reducing the overall system delay.
5. **SECURITY_DELAY:**
    - **Interpretation:** There is a moderate positive correlation with SECURITY_DELAY. An increase in LATE_AIRCRAFT_DELAY is associated with an increase in SECURITY_DELAY. This could be due to additional security measures or inspections needed when dealing with late-arriving aircraft, impacting the security delay.
6. **WEATHER_DELAY:**
    - **Interpretation:** A strong negative correlation with WEATHER_DELAY. As LATE_AIRCRAFT_DELAY increases, WEATHER_DELAY tends to decrease. This may be because weather delays can affect different aspects of flight operations, and late aircraft might be less influenced by weather conditions due to adjustments in scheduling or other operational factors.

7. **LATE_AIRCRAFT_DELAY:**

Interpretation: Perfect positive correlation, which is expected since it represents the same variable or a duplicate entry. This indicates that LATE_AIRCRAFT_DELAY is identical to itself, reinforcing consistency in the data.

Data preprocessing

Handling Missing Values

- High Proportion of Missing Values: Columns such as (TAIL_NUMBER, DEPARTURE_TIME, DEPARTURE_DELAY, TAXI_OUT, WHEELS_OFF, ELAPSED_TIME, AIR_TIME, WHEELS_ON, TAXI_IN, ARRIVAL_TIME, ARRIVAL_DELAY, and CANCELLATION_REASON)have a significant number of missing values. Strategies such as placeholder values, imputation based on related columns, or statistical methods should be used to address these gaps.
- Minimal Missing Values: Columns such as (YEAR, MONTH, DAY, DAY_OF_WEEK, AIRLINE, FLIGHT_NUMBER, ORIGIN_AIRPORT, DESTINATION_AIRPORT, SCHEDULED_DEPARTURE, DISTANCE, SCHEDULED_ARRIVAL, DIVERTED, and CANCELLED) have no missing values, indicating complete data.

1. Tail Numbers

- **Action:** Replace missing values in the `TAIL_NUMBER` column with the placeholder value `'UNKNOWN_TAIL_NUMBER'`.
- **Reason:** Tail numbers are crucial for identifying individual aircraft, and replacing missing values with a placeholder ensures that all entries are accounted for, even if the actual tail number is unknown.

2. Cancellation Reasons

- **Action:** Fill missing values in the `CANCELLATION_REASON` column with `'UNKNOWN'`.
- **Reason:** If the reason for cancellation is missing, using a placeholder value helps maintain data integrity and allows for analysis without leaving gaps.

3. Departure Times

- **Action:** For missing values in the `DEPARTURE_TIME` column, use the scheduled departure time from the `SCHEDULED_DEPARTURE` column.
- **Reason:** The scheduled departure time can be a reasonable estimate when the actual departure time is missing, ensuring that all records have a departure time.

4. Departure Delays

- **Action:** Calculate `DEPARTURE_DELAY` as the difference between `DEPARTURE_TIME` and `SCHEDULED_DEPARTURE`.
- **Reason:** Departure delay is inherently derived from the departure time and the scheduled time, so if either value is missing, the delay cannot be accurately determined.

5. Taxi-Out Times

- **Action:** For missing values in the `TAXI_OUT` column, use the difference between `WHEELS_OFF` and `DEPARTURE_TIME`.
- **Reason:** Taxi-out time is the duration between the departure time and wheels-off time. Filling in missing values using this calculation ensures that the taxi-out duration is accurately represented.

6. Wheels-Off Times

- **Action:** For missing values in the `WHEELS_OFF` column, use the sum of `DEPARTURE_TIME` and `TAXI_OUT`.
- **Reason:** Wheels-off time is typically the departure time plus taxi-out time. This approach provides a reasonable estimate for missing wheels-off times.

7. Fill Remaining Wheels-Off Times

- **Action:** If there are still missing values in the `WHEELS_OFF` column, replace them with the median value of this column.
- **Reason:** Using the median provides a robust estimate for missing values, minimizing the impact of outliers.

8. Fill Remaining Taxi-Out Times

- **Action:** If there are still missing values in the `TAXI_OUT` column, replace them with the median value of this column.
- **Reason:** The median value offers a stable estimate for missing taxi-out times, especially in cases where the distribution might be skewed.

9. Scheduled Times

- **Action:** For missing values in the `SCHEDULED_TIME` column, use the difference between `SCHEDULED_ARRIVAL` and `SCHEDULED_DEPARTURE`.
- **Reason:** Scheduled time is the duration between the scheduled arrival and departure times. This calculation provides an accurate estimate for missing scheduled times.

10. Arrival Times - **Action:** For missing values in the `ARRIVAL_TIME` column, use the sum of `SCHEDULED_DEPARTURE` and `ELAPSED_TIME`. - **Reason:** Arrival time is typically the scheduled departure time plus elapsed time. This approach fills missing values based on the expected arrival time.

11. Elapsed Times - **Action:** For missing values in the `ELAPSED_TIME` column, use the difference between `ARRIVAL_TIME` and `DEPARTURE_TIME`. - **Reason:** Elapsed time is calculated as the difference between arrival and departure times. This calculation provides a precise estimate for missing elapsed times.

12. Fill Remaining Arrival Times - **Action:** If there are still missing values in the ARRIVAL_TIME column, replace them with the median value of this column. - **Reason:** The median value helps in providing a stable estimate for missing arrival times, mitigating the effects of any extreme values.

13. Fill Remaining Elapsed Times - **Action:** If there are still missing values in the ELAPSED_TIME column, replace them with the median value of this column. - **Reason:** The median value is used to fill remaining gaps in elapsed times, ensuring a stable estimate unaffected by extreme values.

14. Air Times - **Action:** For missing values in the AIR_TIME column, use the calculation based on ELAPSED_TIME minus TAXI_OUT, adjusted for the difference between WHEELS_OFF and DEPARTURE_TIME. - **Reason:** Air time is derived from elapsed time minus taxi-out time. This calculation provides a meaningful estimate when air time is missing.

15. Wheels-On Times - **Action:** For missing values in the WHEELS_ON column, use the sum of WHEELS_OFF and ELAPSED_TIME. - **Reason:** Wheels-on time is the wheels-off time plus elapsed time. This estimate fills missing values based on expected wheels-on times.

16. Taxi-In Times - **Action:** For missing values in the TAXI_IN column, use the difference between ELAPSED_TIME, AIR_TIME, and TAXI_OUT. - **Reason:** Taxi-in time is calculated as the remaining time after accounting for air time and taxi-out time. This approach provides a reasonable estimate for missing taxi-in times.

17. Arrival Delays - **Action:** For missing values in the ARRIVAL_DELAY column, use the difference between ARRIVAL_TIME and SCHEDULED_ARRIVAL. - **Reason:** Arrival delay is the difference between actual and scheduled arrival times. This calculation fills missing values based on expected delays.

18. Air System Delays - **Action:** For missing values in the AIR_SYSTEM_DELAY column, use the sum of DEPARTURE_DELAY and ARRIVAL_DELAY. - **Reason:** Air system delays can be considered as the sum of departure and arrival delays. This method provides a reasonable estimate for missing values.

19. Security Delays - **Action:** For missing values in the SECURITY_DELAY column, use the difference between DEPARTURE_DELAY and ARRIVAL_DELAY. - **Reason:** Security delay can be estimated as the difference between departure and arrival delays, providing a meaningful value for missing data.

20. Airline Delays - **Action:** For missing values in the AIRLINE_DELAY column, use the difference between DEPARTURE_DELAY and TAXI_OUT. - **Reason:** Airline delay is typically the departure delay minus taxi-out time. This calculation fills missing values based on expected airline delays.

21. Late Aircraft Delays - **Action:** For missing values in the LATE_AIRCRAFT_DELAY column, use the difference between AIR_TIME and ELAPSED_TIME. - **Reason:** Late aircraft delay can be estimated as the difference between air time and elapsed time, filling gaps in the dataset with reasonable values.

22. Weather Delays - **Action:** For missing values in the WEATHER_DELAY column, use the difference between ARRIVAL_DELAY and AIR_TIME. - **Reason:** Weather delay can be derived from the difference between arrival delay and air time, providing a meaningful estimate for missing values.

Feature Engineering

▪ Outlier Detection and Treatment

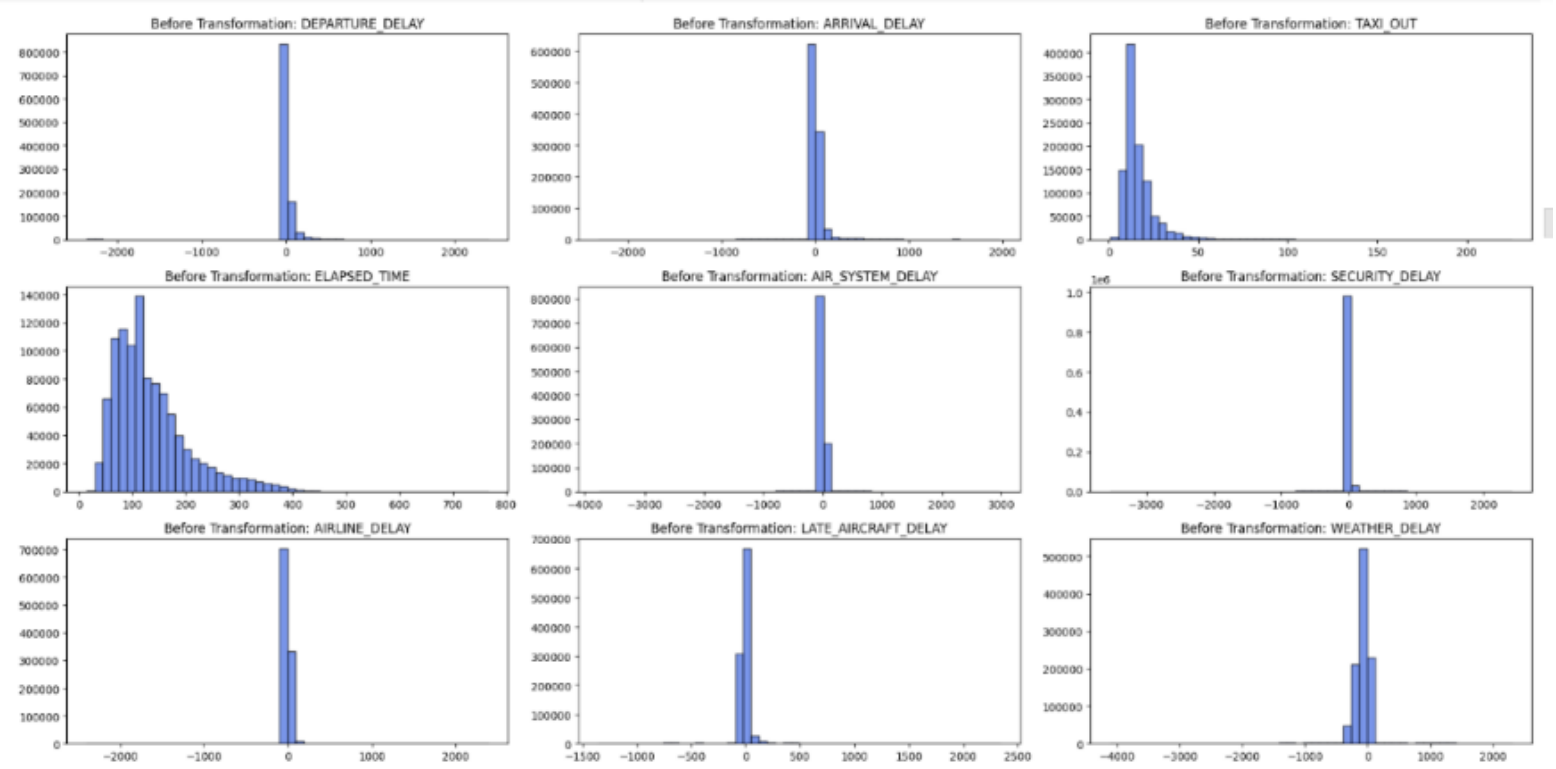
Square Root Transformation

Objective
The square root transformation is applied to certain columns in your dataset to address issues related to the distribution of data, particularly when the data exhibits skewness or extreme outliers.

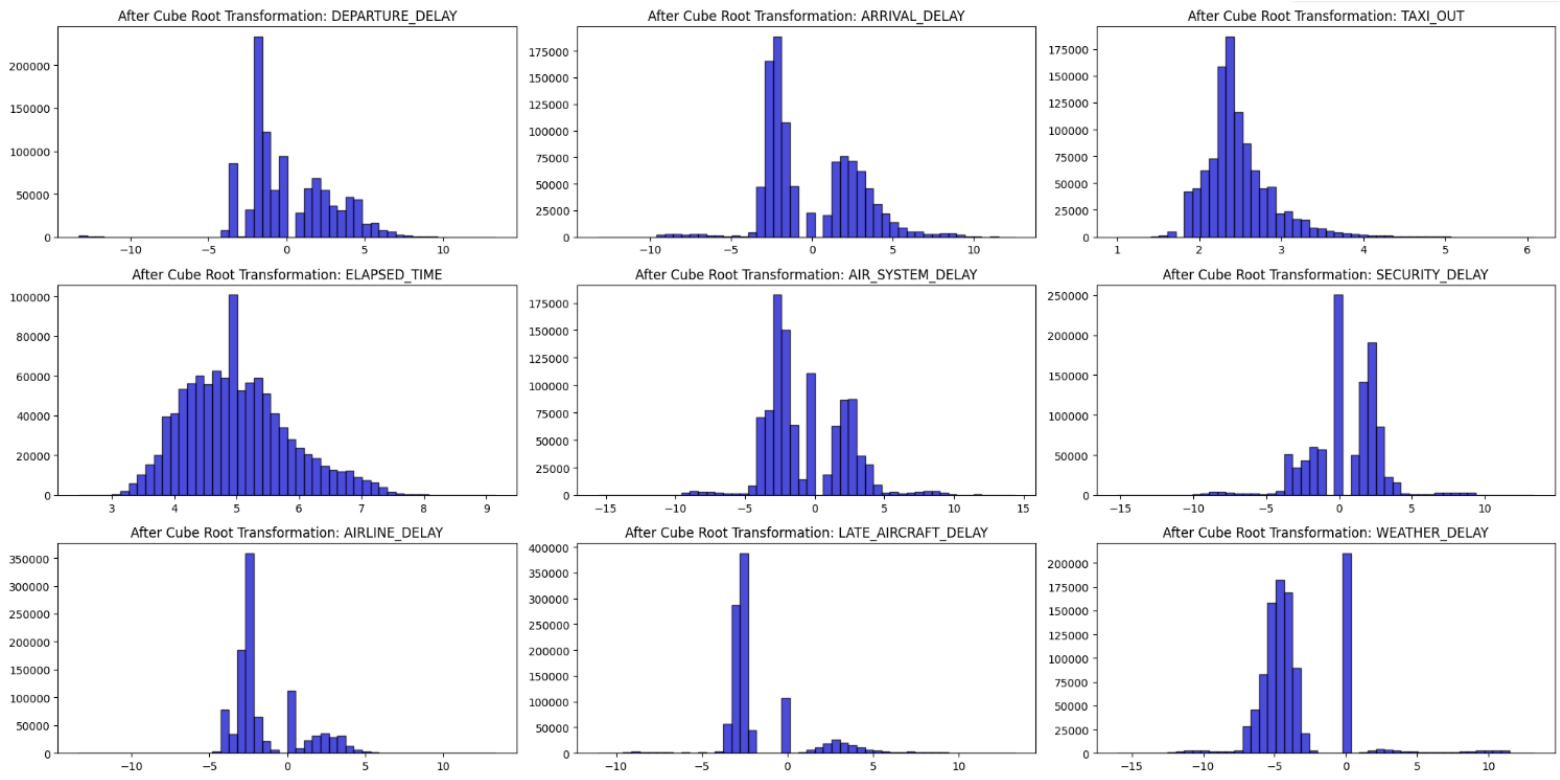
Why Use Square Root Transformation?

1. **Normalization of Skewed Data:** Many datasets, especially those involving count data or measurements, can be highly skewed. The square root transformation helps to reduce skewness and make the data distribution more normal (Gaussian-like). This is crucial for many statistical and machine learning methods that assume normally distributed data.
2. **Handling Outliers:** The square root transformation is particularly effective in reducing the impact of extreme values. By compressing the scale of the data, it diminishes the influence of outliers, leading to more stable and effective model training. This helps in managing the variability caused by extreme values.
3. **Variance Stabilization:** It helps in stabilizing the variance across the range of the data, making the data more homogeneous and suitable for further analysis. This ensures that the variance does not disproportionately affect the model's performance.

Before Using Square Root for Outliers



After Using Square Root for Outliers



▪ Delay Columns into a Single Column

Objective: To predict the type of flight delay by consolidating multiple delay-related columns into a single column that categorizes the delay reason.

Process:

1. **Identify Delay Columns:** The dataset includes several columns related to different types of delays:
    - **SECURITY_DELAY**
    - **AIRLINE_DELAY**
    - **LATE_AIRCRAFT_DELAY**
    - **WEATHER_DELAY**

2. **Define Delay Classification Criteria:** The goal is to classify the delay reason based on the presence and magnitude of delays recorded in the identified columns**. The classification follows these rules:**

    - **Security Delay:** If SECURITY_DELAY is greater than 0.
    - **Airline Delay:** If AIRLINE_DELAY is greater than 0 and none of the other delay columns (SECURITY_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY) are greater than 0.
    - **Late Aircraft Delay:** If LATE_AIRCRAFT_DELAY is greater than 0 and none of the other delay columns (SECURITY_DELAY, AIRLINE_DELAY, WEATHER_DELAY) are greater than 0.
    - **Weather Delay:** If WEATHER_DELAY is greater than 0 and none of the other delay columns (SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY) are greater than 0.
    - **No Delay:** If all the delay columns are 0 or less.
    -

3. **Create a Classification Function:** A function is defined to apply the classification criteria to each row in the Data Frame. This function checks the values in the delay columns and assigns a corresponding delay reason type.
4. **Apply the Classification Function:** The classification function is applied to the Data Frame using the apply method. This operation creates a new column (Delay reason type) that holds the delay reason for each flight based on the defined criteria.
5. **Verify Results:** After creating the (Delay reason type) column, verify its correctness by inspecting a few rows of the Data Frame to ensure that the new column accurately reflects the delay reasons based on the delay columns.

Delay_reason_type	count
Security Delay	526250
No Delay	311890
Airline Delay	123791
Late Aircraft Delay	58076
Weather Delay	28568

Output:
The output will show the first few rows of the Data Frame, including the newly created (Delay reason type)column alongside the original delay-related columns.

- ## Prepare Feature Matrix and Target Variable

Objective: Separate the features and target variable from the dataset.

- Prepare Features:
  - Use the drop() method to remove the target variable column from the data Frame.
  - Select the target variable column from the data Frame.

Separating features and the target variable is a crucial step in preparing data for machine learning. Here's why:

1. Model Training:
   - Features (X): The independent variables that the model will use to make predictions.
   - Target Variable (y): The dependent variable that the model will predict based on the features.
2. Data Preparation:
   - Feature Matrix (X): Contains all relevant input data that will be used to train the model.
   - Target Vector (y): Contains the output or label that the model aims to predict.

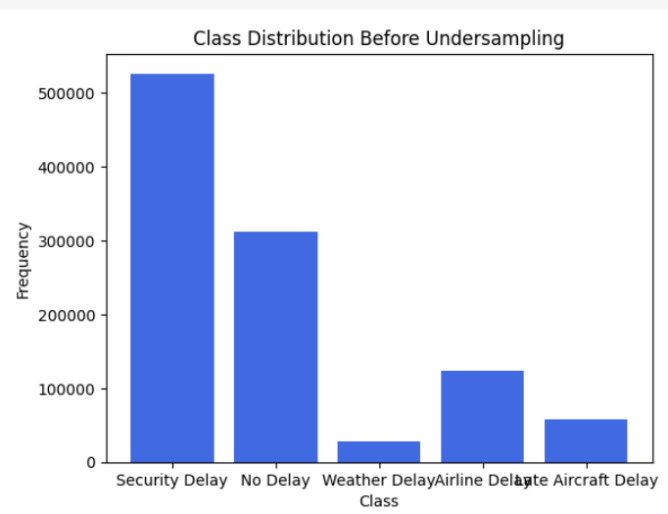
Accessing the underlying values of features and target variables using the `.values` attribute is essential for several reasons:

- **Compatibility with Machine Learning Libraries**:

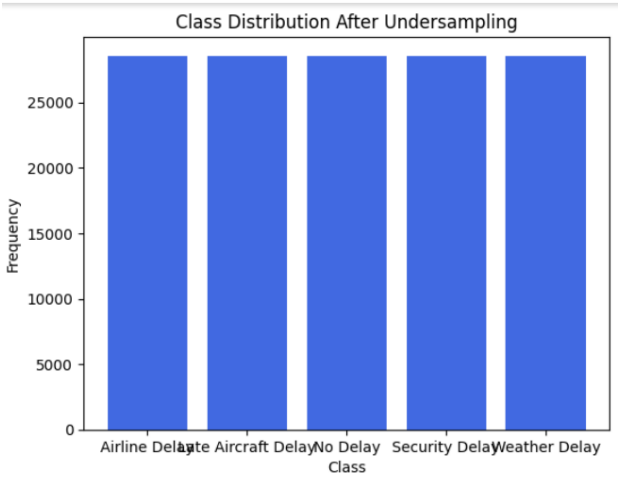
- Many machine learning libraries and algorithms, especially those in scikit-learn and other numerical computing libraries, require inputs in the form of NumPy arrays rather than pandas Data Frames or Series.

- **Performance Optimization**:
  - NumPy arrays are often faster and more memory-efficient than pandas Data Frames or Series for numerical operations and matrix manipulations. Converting to NumPy arrays can speed up computations and reduce overhead.
- **Matrix Operations**:
  - Many machine learning models and mathematical operations are based on matrix algebra. NumPy arrays facilitate efficient matrix operations like dot products, which are central to algorithms like linear regression, neural networks, and others.
- **Integration with External Libraries**:
  - Some specialized libraries or functions expect data in NumPy array format. Converting your data ensures compatibility with a broader range of tools and libraries.
- **Consistency in Data Handling**:
  - When performing operations that involve both features and target variables, having them in NumPy array format ensures consistency and simplifies the code

Resampling for Class Imbalance



The image shows the class distribution of a dataset before under sampling. The x-axis represents the different classes, while the y-axis represents the frequency of each class. The bar chart indicates that the dataset is highly imbalanced, with a significant number of instances belonging to the "No Delay" class, followed by "Security Delay" and "Weather Delay". The remaining classes, "Airline Delay", "Late Aircraft Delay", have much fewer instances.
This imbalance can pose challenges for machine learning models, as they may be biased towards the majority class and struggle to accurately predict the minority classes. Under sampling is a technique used to address this imbalance by randomly removing instances from the majority class to create a more balanced distribution.



the image shows the class distribution of a dataset after under sampling. The x-axis represents the different classes, while the y-axis represents the frequency of each class. The bar chart indicates that the under-sampling process has successfully balanced the class distribution, as all classes now have approximately the same frequency.

This is a desirable outcome for machine learning models, as it helps to prevent bias towards the majority class and improves the model's ability to accurately predict all classes.

Why Use Under sampling:

- **Data Imbalance:** In your case, if the dataset is too large and the imbalance is causing poor performance of your machine learning model, under sampling can help by balancing the classes.
- **Performance:** Models trained on balanced datasets often perform better because they are not biased towards the majority class.
- **Efficiency:** For very large datasets, reducing the number of majority class samples can make the training process faster and more **manageable.**

- ## Categorical Data with Encoders

1.One Hot Encoder for Categorical Variables

	0	1	2	3	4
0	1.000000	0.000000	0.000000	0.000000	0.000000
1	1.000000	0.000000	0.000000	0.000000	0.000000
2	1.000000	0.000000	0.000000	0.000000	0.000000
3	1.000000	0.000000	0.000000	0.000000	0.000000
4	1.000000	0.000000	0.000000	0.000000	0.000000
5	1.000000	0.000000	0.000000	0.000000	0.000000
6	1.000000	0.000000	0.000000	0.000000	0.000000
7	1.000000	0.000000	0.000000	0.000000	0.000000
8	1.000000	0.000000	0.000000	0.000000	0.000000

Purpose:

- One Hot Encoder is used to convert categorical variables into a format that can be provided to machine learning algorithms. It is specifically used when you have categorical data with more than two categories, where each category is transformed into a separate binary column. This approach ensures that the model does not assume any ordinal relationship among the categories.

Usage:

- One Hot Encoder is typically used for features (independent variables) in your dataset. It creates a binary matrix where each column represents a unique category of the feature, and each row corresponds to the presence or absence of that category for a specific observation.

Label Encoder for Categorical Variables

Purpose:

- Label Encoder is used to convert categorical values into numerical values. This is particularly useful when you have categorical data that needs to be represented as numeric values for machine learning algorithms that require numerical input.

Usage:

- Label Encoder is commonly applied to target variables or categorical features that represent distinct classes or labels. Each unique category is assigned a unique integer value.

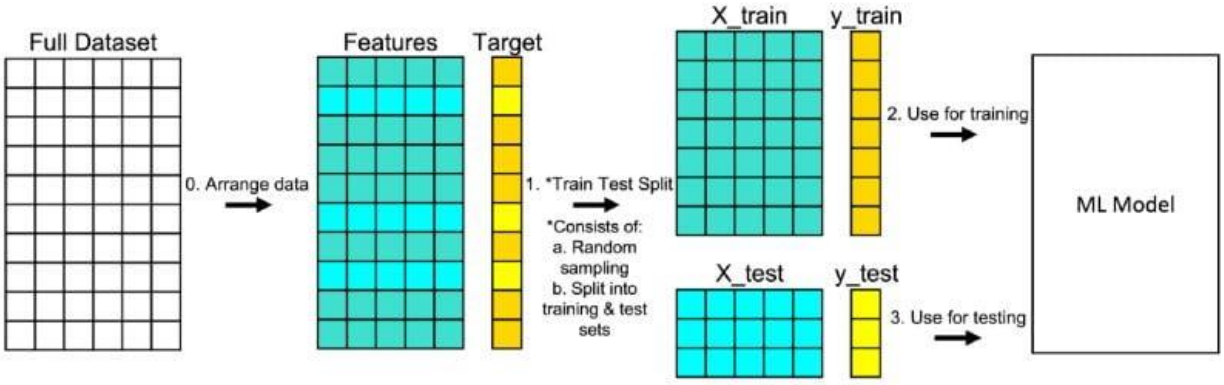
The CANCELLATION_REASON column contains categorical values that represent different reasons for flight cancellations. In a dataset with many features, adding additional columns through encoding methods like OneHotEncoding would increase the number of columns, making the dataset larger and more complex for machine learning models to process.

Why Label Encoder?

- **Handling Many Features Efficiently:** Given that the dataset already contains a large number of columns, adding more through OneHotEncoding would further increase the dimensionality. This can lead to higher computational costs and potentially overfitting. Label Encoder, on the other hand, encodes categories into a single column, preventing this complexity.
- **Efficient Transformation:** Label Encoder transforms the categorical values into integer labels, reducing the dataset's overall size without losing important information about the cancellation reasons.
- **Maintaining Manageability:** By keeping the encoded data compact, Label Encoder helps manage large datasets efficiently, allowing the model to focus on other features without being overwhelmed by additional columns.

This method ensures that the dataset remains computationally efficient and avoids unnecessary complexity from too many features.

- **Splitting the Dataset: train, test, split**



Reasons for Train-Test Split:

1. **Model Evaluation**: The train-test split allows us to train the model on one portion of the data (training set) and evaluate it on another (testing set). This mimics real-world scenarios where the model encounters unseen data, helping us gauge how well it will perform in practice.

2. **Prevent Overfitting**: If we train and test the model on the same dataset, it might perform exceptionally well on that data but fail when encountering new data (overfitting). By splitting the data, we avoid this and ensure the model doesn't just memorize the data but generalizes well.

3. **Hyperparameter Tuning**: Having a separate testing set allows for more effective hyperparameter tuning and model selection. We can compare different models or parameters based on their performance on unseen data, leading to better choices.

4. **Measuring Generalization**: The test set provides an unbiased evaluation of the model's performance, helping measure how well it generalizes to new, unseen data. This is important for understanding the model's true predictive power beyond the data it was trained on.
   By performing a train-test split, we ensure a fair and reliable method to assess the model's performance and prevent overfitting.

Parameter	Value	Explanation
test_size	**0.2**	**Specifies that 20% of the data should be used for testing, while 80% is used for training.**
random_state	**42**	**Ensures that the split is reproducible, providing the same train-test split each time the code is run.**

Scaling

Standard Scaler

Standard Scaler is a preprocessing technique used to standardize features by removing the mean and scaling to unit variance. This technique is essential for many machine learning algorithms, especially those that assume features are centered around zero and have the same scale.

1. What is Standard Scaler?

Standard Scaler transforms features by removing the mean and scaling to unit variance. This process is also known as standardization or z-score normalization. The formula for standardization is:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

.

2. When to Use Standard 'Scaler

- **Algorithm Sensitivity:** When using algorithms that are sensitive to the scale of features, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Principal Component Analysis (PCA).
- **Feature Distribution:** When your features are normally distributed or approximately normally distributed.
- **Feature Comparison:** When you need to compare features with different units or scales.

3. How Standard Scaler Works

1. **Compute Mean and Standard Deviation:**
    - Calculate the mean (μ\muμ) and standard deviation (σ\sigmaσ) of each feature across the training dataset.
2. **Apply Standardization Formula:**
    - For each feature value, subtract the mean and divide by the standard deviation using the formula.
3. **Transform the Data:**
    - Replace the original feature values with their standardized versions.

4. Advantages and Disadvantages

Advantages:

- **Centres Data**: Centers the data around zero, making it easier for algorithms that assume data is centered.

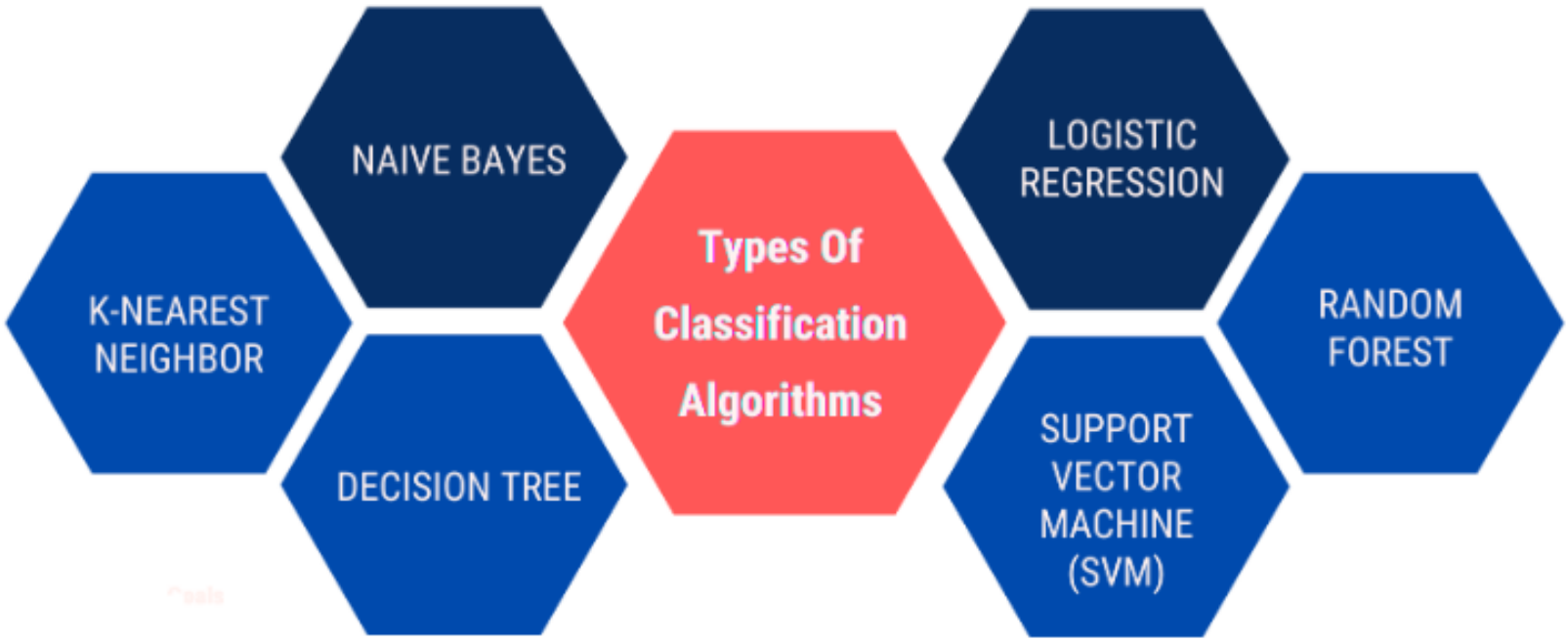
- **Unit Variance:** Scales data to have unit variance, which is beneficial for algorithms sensitive to feature scaling.

Disadvantages:

- **Sensitive to Outliers:** Outliers can significantly affect the mean and standard deviation, potentially skewing the standardized values.
- **Assumes Normal Distribution:** Standard Scaler works best when features are approximately normally distributed.

MACHINE LEARNING

MACHINE LEARNING ALGORITHMS



1.KNN stands for K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple and intuitive machine learning algorithm used for classification and regression. Here's a breakdown of how it works and how you can use it:

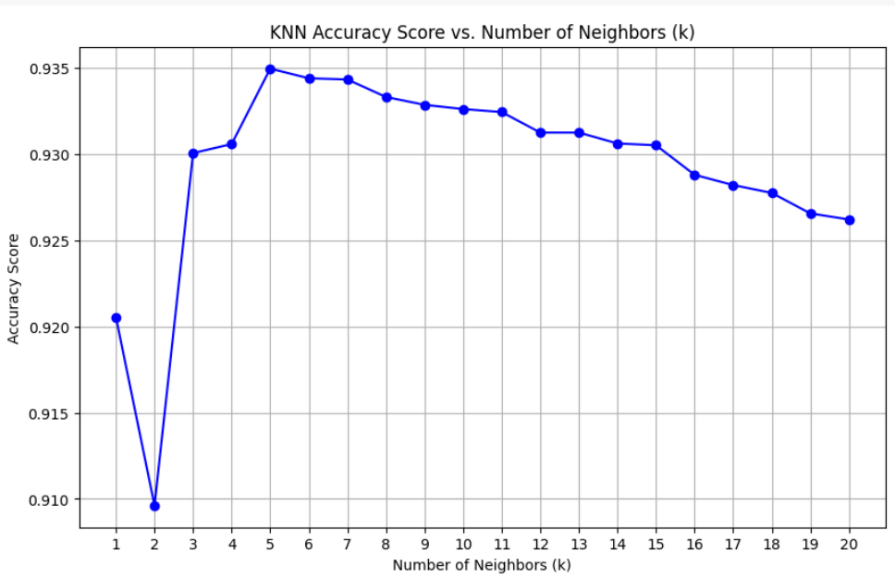
How KNN Works:

1. **Basic Idea:**
    - KNN classifies a new data point based on the majority class among its k nearest neighbors in the training set. For regression, it predicts the value based on the average (or weighted average) of the values of its k nearest neighbors.
2. **Distance Metric:**
    - KNN uses distance metrics (e.g., Euclidean distance, Manhattan distance) to find the k closest points to the query point.
3. **Choosing k:**
    - The choice of k (number of neighbors) affects the performance of the model. A small k may lead to overfitting, while a large k may smooth out the decision boundary too much, leading to underfitting.
4. **Classification:**
    - For classification, the class with the majority vote among the k neighbors is assigned to the query point.
5. **Regression:**
    - For regression, the prediction is typically the average of the target values of the k nearest neighbors.

n classification using K-Nearest Neighbors (KNN), the distance between the new point and each point in the training dataset is calculated to identify the nearest neighbors. You can use distance metrics such as Euclidean distance or Manhattan distance. For Euclidean distance, the formula is:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^{M} (x_{i,m} - x_{j,m})^2}$$

Perimetric Study:



K-Nearest Neighbors (KNN) Classification Results

Parameter	Value	Accuracy
n_neighbors	**5**	**93**

2.Decision Trees

Basic Idea:

- Decision Trees split the data into branches based on feature values, creating a tree-like structure. Each node in the tree represents a feature, and branches represent the decision made based on that feature.
- It recursively splits the dataset until it either perfectly classifies the data or stops due to some criteria (e.g., maximum depth).

How it Works:

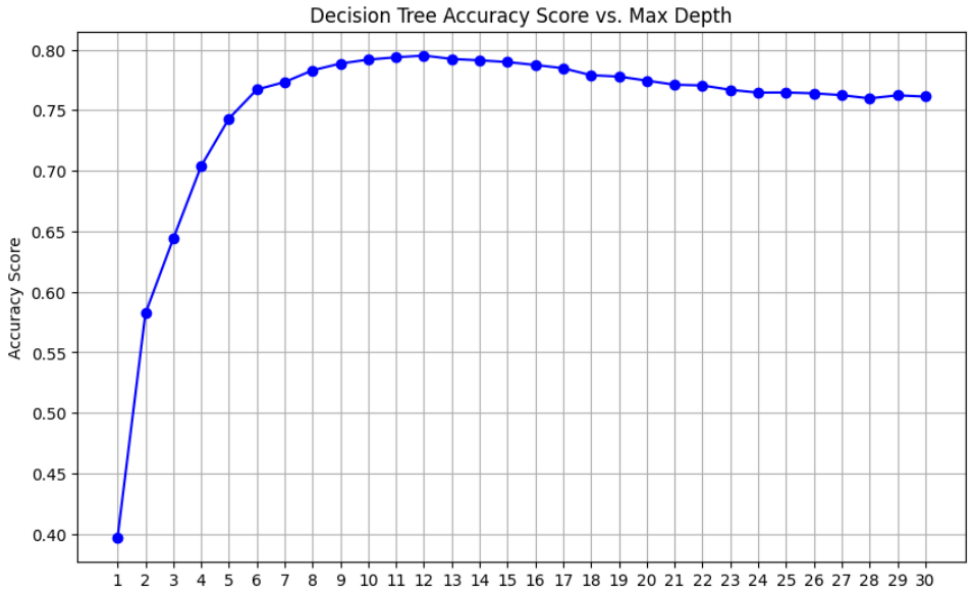
1. **Splitting**:
    - The dataset is split based on the feature that provides the best separation (e.g., using Gini Impurity or Information Gain).
2. **Decision Nodes and Leaves**:
    - Decision nodes are where the data is split, and leaves represent the final classification or regression output.

3. **Classification**:
    - For classification, once the data reaches a leaf node, the most frequent class label among the samples in that leaf is assigned.
4. **Regression**:
    - For regression, the prediction is based on the average target value of the samples in the leaf node.
5. **Stopping Criteria**:
    - The tree stops growing based on criteria like maximum depth or minimum samples in a node, to avoid overfitting.

Gini Impurity:

$$^{2}_{i}p \sum_{i=1}^{n} - 1 = Gini$$

Perimetric Study:



Decision Trees Classification Results:

Parameter	Hyperparameter	Value	Accuracy
Decision Trees	**max_depth**	**3**	**0.80**

3.Random Forest

Basic Idea:

- Random Forest is an ensemble method that builds multiple decision trees (typically trained on random subsets of the data and features). The final prediction is made by aggregating the predictions from all the trees.

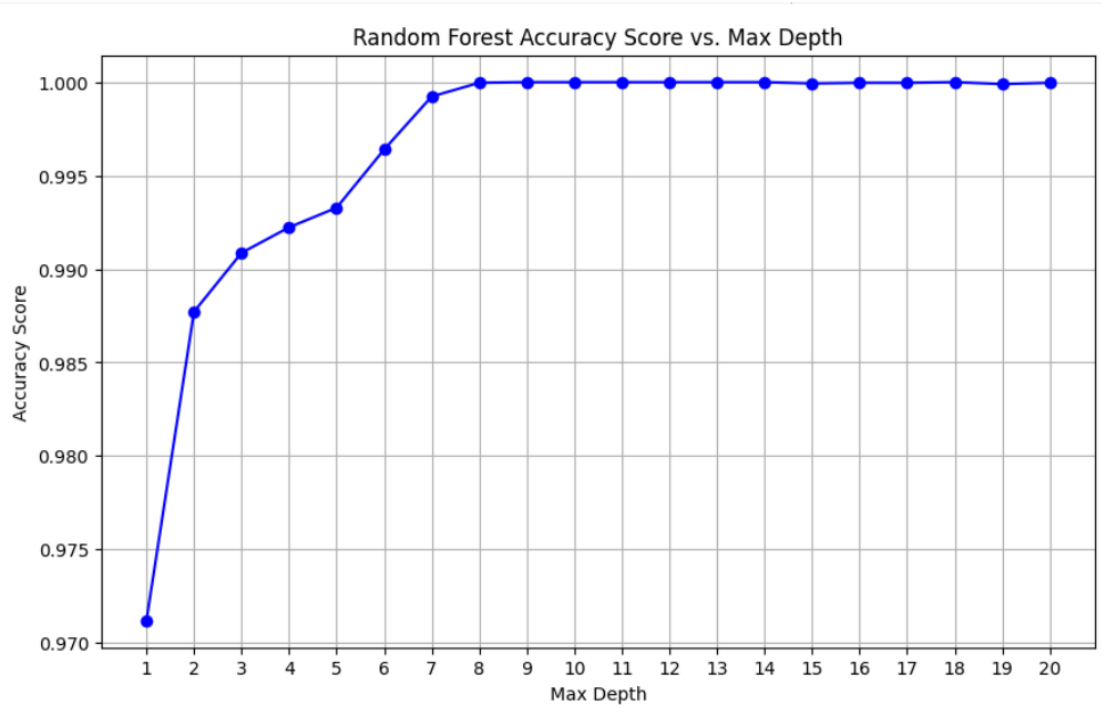
How it Works:

1. **Bagging**:
    - Random subsets of the data are drawn (with replacement) to build each tree, which reduces variance and prevents overfitting.
2. **Random Feature Selection**:
    - At each split in a tree, a random subset of features is selected, which introduces diversity among the trees.
3. **Classification**:
    - For classification, each tree votes on the class, and the class with the most votes is chosen (majority voting).
4. **Regression**:
    - For regression, the predictions from each tree are averaged to provide the final prediction.
5. **Hyperparameters**:
    - Key hyperparameters include the number of trees (n_estimators), maximum depth of each tree, and the number of features considered at each split.

      y^=mode{T1(x),T2(x),…,TB(x)}

$$T_b(x) \sum_{b=1}^{B} \frac{1}{B} = \hat{} y$$

Perimetric Study:



Random Forest Model Performance

Parameter	Hyperparameter	Value	Accuracy
Random Forest	**max_depth**	**8**	**0.99**

4.Support Vector Classifier (SVC)

Basic Idea:

- Support Vector Classifier finds the hyperplane that best separates the data into classes by maximizing the margin between the two closest points of different classes (support vectors).

How it Works:

1. **Hyperplane**:
    - SVC tries to find the optimal hyperplane that separates the classes. In 2D, this is a line; in higher dimensions, it's a plane or hyperplane.

      The equation of the **Hyperplane** is: $f(x) = w^T x + b$

2. **Margin Maximization**:
    - SVC works by maximizing the distance (margin) between the hyperplane and the nearest data points (support vectors) from both classes.

    Margin: $$\text{Margin} = \frac{2}{\|w\|}$$
3. **Kernels**:
    - For non-linearly separable data, SVC can transform the data using kernels (e.g., linear, polynomial, radial basis function) to find a better separating hyperplane.
4. **Classification**:
    - For classification, the model assigns a class label based on which side of the hyperplane the new data point falls.
5. **Soft Margin**:
    - SVC can use a soft margin to allow for misclassifications in the case of overlapping classes. This is controlled by the C parameter (trade-off between correct classification and margin maximization).

SVC Model Performance

Parameter	Hyperparameter	Value	Accuracy
SVC	--------	----------	**0.99**

5.Logistic Regression

Basic Idea:

Logistic Regression estimates the probability that a given data point belongs to a particular class. It is used for binary classification problems by modeling the relationship between the input features and the binary outcome using a sigmoid function.

How it Works:

Probability Estimation:

Logistic Regression predicts the probability of a data point belonging to class 1 (or class 0). This is done by applying the sigmoid function to a linear combination of the input features.

$$p(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

Decision Boundary:

Logistic Regression assigns a class label based on the probability prediction. Typically, a threshold of 0.5 is used:

- If the predicted probability is ≥ 0.5, the data point is classified as class 1.
- If the probability is < 0.5, it is classified as class 0.

Linear Relationship:

The model assumes a linear relationship between the input features and the log-odds (logit) of the outcome. This means the decision boundary between the classes is linear.

$$\text{logit}(p) = w^T x + b$$

Classification:

For classification, Logistic Regression uses the predicted probabilities to assign a label. If the data point has a probability greater than or equal to the threshold (usually 0.5), it is classified as class 1; otherwise, it's class 0.

Regularization:

To avoid overfitting, Logistic Regression often uses **L2 regularization** (Ridge), which adds a penalty for large coefficients. The strength of the penalty is controlled by the regularization parameter **C**. Smaller values of **C** imply stronger regularization.

Multiclass Extensions:

Although Logistic Regression is primarily a binary classifier, it can be extended to handle multiclass problems using strategies like **One-vs-Rest (OvR)** or **Soft max Regression**.

Logistic Regression Model Performance

Parameter	Hyperparameter	Value	Accuracy
Logistic Regression	**----------**	**----------**	**0.99**

Machine Learning and ANN

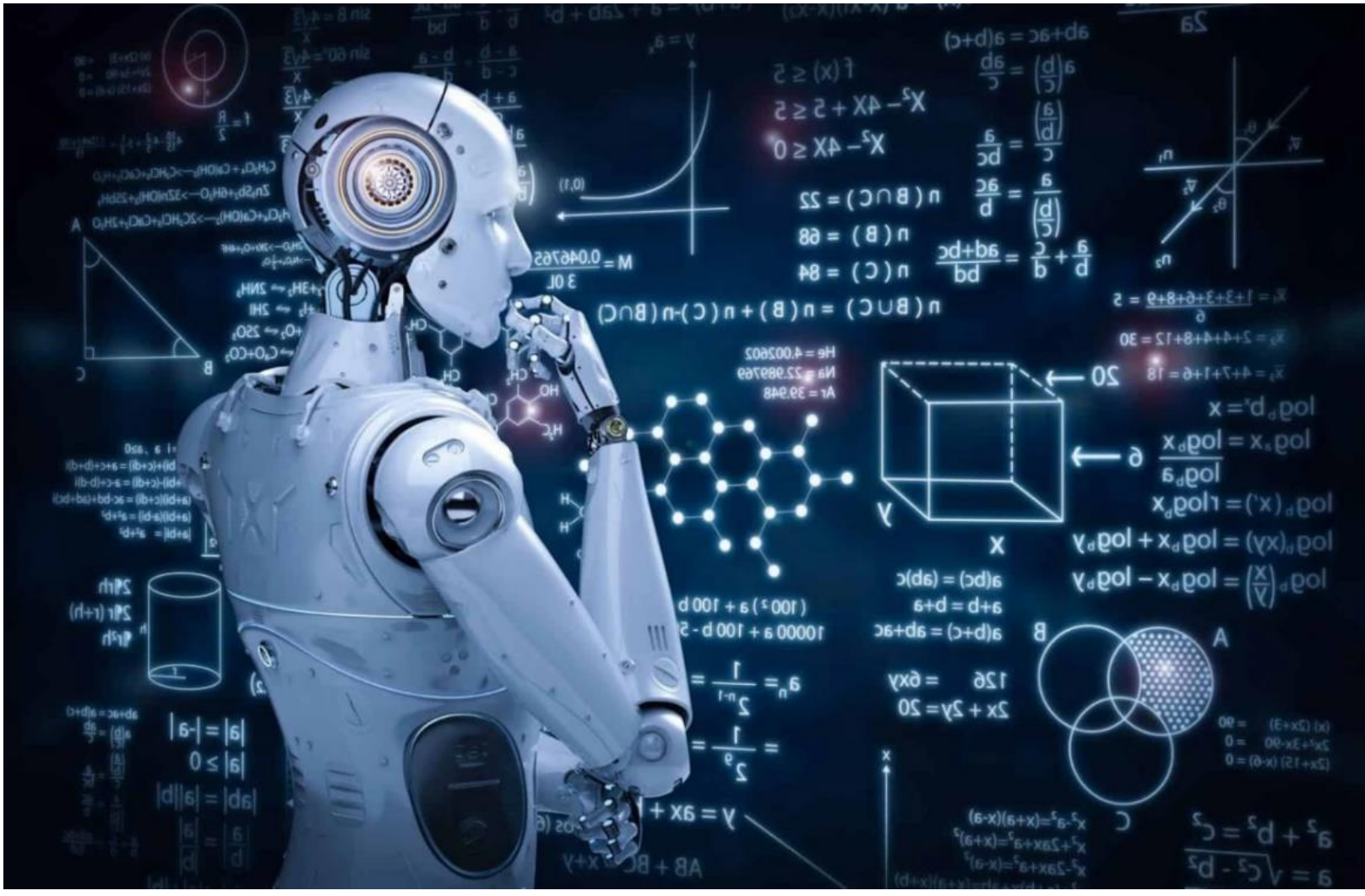
The performance of traditional machine learning models may be lower than that of artificial neural networks (ANNs) in some cases due to several key factors:

1. **Data Complexity**

- **Traditional Machine Learning**: Models like logistic regression or decision trees may struggle to capture complex interactions between features if the data contains nonlinear or intertwined relationships.
- **Artificial Neural Networks (ANNs)**: ANNs can handle complex interactions and nonlinear patterns more effectively due to their ability to learn multi-layered representations of the data.

2. **Model Generalization**
   - **Traditional Machine Learning**: Simpler models may be prone to underfitting, leading to poor performance if the data contains intricate patterns.
   - **Artificial Neural Networks (ANNs)**: ANNs can have many layers and neurons, allowing them to learn more complex features and better adapt to the data.
3. **Data Size**
   - **Traditional Machine Learning**: Simple models might underperform if the dataset is large and complex.
   - **Artificial Neural Networks (ANNs)**: ANNs often perform better with large amounts of data, as they can leverage extensive training data to learn more precise patterns.
4. **Feature Engineering and Normalization**
   - **Traditional Machine Learning**: Models may require extensive preprocessing and feature scaling to achieve good performance.
   - **Artificial Neural Networks (ANNs)**: ANNs can be more flexible in handling raw or poorly engineered features, thanks to their ability to automatically learn feature representations.

Deep Learning



Building ANN Model

Artificial Neural Networks (ANNs) are used for a variety of tasks including classification and regression. They are designed to mimic the way the human brain processes information and are particularly useful when dealing with complex datasets.

Why Use an ANN?

1. **Handling Non Linearity:** ANNs can model complex non-linear relationships between inputs and outputs.
2. **Feature Learning:** ANNs can automatically learn and extract features from raw data.
3. **Flexibility:** They can be used for various types of problems such as classification, regression, and even unsupervised learning tasks.

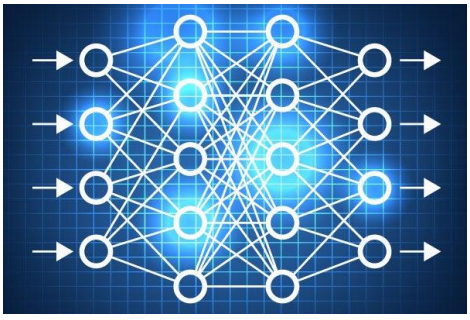
Building an ANN Model

Here's a step-by-step breakdown of building an ANN model using Keras:

1. **Import Libraries:**
   Purpose: Provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays

2. **Initialize the Model:**
   the model is defined as a Sequential object in Keras, which represents a linear stack of layers. This is suitable for most simple neural network architectures where layers are stacked sequentially.

3. **Add Layers:**



Steps to Add Layers:

1. **Input Layer:**
   The first layer must specify the input shape or the number
   for example, if your input has 10 features, you would specify input_dim=10.

2. Hidden Layers:

Add one or more hidden layers. Each hidden layer consists of neurons, activation functions, and other parameters.

- **Dense Layer:** Fully connected layer where each neuron is connected to all neurons in the previous layer.
- **Activation Function:** Introduces non-linearity. Common choices are ReLU, sigmoid, and tanh.

3.Add Output Layer:

the output layer produces the final prediction. The activation function here depends on the type of classification task (e.g., soft max for multi-class classification or sigmoid for binary classification).

4.Compile the Model:

Define the optimizer, loss function, and metrics to be used.
Compiling the model in Keras involves configuring the optimizer, loss function, and metrics used to evaluate the model. This step prepares the model for training.

- **Define the Optimizer**: The optimizer updates the weights of the network based on the gradients computed during backpropagation. It controls how the model learns from the training data.
- **Common Optimizers:**
  SGD (Stochastic Gradient Descent): Updates weights based on the gradient of the loss function with respect to the weights.
  Adam: An adaptive learning rate optimizer that combines features from other optimizers like momentum.
- **the Loss Function:**
  The loss function measures the discrepancy between the predicted values and the actual target values. It is minimized during training**.**
  Categorical Cross-Entropy: Used for multi-class classification problems where targets are one-hot encoded.
  Sparse Categorical Cross-Entropy: Used for multi-class classification problems with integer encoded targets.
  Binary Cross-Entropy: Used for binary classification problems.
  python
- **Define the Metrics:**
  Metrics are used to evaluate the performance of the model during training and testing. They provide insight into how well the model is performing based on the task.
  Accuracy: Measures the proportion of correct predictions.
- **Compile the Model:** the compile method in Keras ties together the optimizer, loss function, and metrics, preparing the model for training.

4. **Set Up Callbacks:**

   Use Early Stopping to prevent overfitting by monitoring validation loss and restoring the best weights.

5. **Fit the Model:**

   Train the model on your dataset using the fit method. Pass in the training data, validation data, and any call backs .he fit method trains the model on the training data, validates it on the validation data, and optionally uses callbacks to monitor the training process.
   Parameters:

- **x**: The input data for training.
- **y**: The target data for training.
- **Validation data**: A tuple () for validating the model during training.
- **epochs**: Number of epochs to train the model.
- **Batch size**: Number of samples per gradient update.
- **callbacks**: List of callback functions to apply during training (…, Early Stopping).

5.Tuning

Parameter	Value
Optimizer	Sgd
Loss Function	Categorical Crossentropy
Activation Function (Hidden Layers)	ReLU (Dense Layers)
Number of Epochs	20
Batch Size	300
Early Stopping (Monitor)	val_loss
Early Stopping (Patience)	3
Layer 1	Dense, 12 units, ReLU
Layer 2	Dense, 8 units, ReLU
Layer 3	Dense, 4 units, ReLU
Output Layer	Dense, 5 units, Softmax

Including this information in a table provides a clear and organized way to present the configuration of your neural network model. Here's why each entry is valuable:

Explanation of Each Table Entry

- **Optimizer: SGD (Stochastic Gradient Descent)**

Purpose: SGD updates the model parameters using only a single or a few training examples at a time. It's simpler and requires less memory compared to adaptive algorithms like Adam. While it may converge more slowly and requires careful tuning, it's effective for large-scale datasets.

- **Loss Function: Categorical Crossentropy**

Purpose: This loss function is used for multi-class classification problems, measuring how well the predicted probability distribution aligns with the true distribution. It helps the model to minimize the error in its predictions.

- **Activation Function (Hidden Layers): ReLU (Rectified Linear Unit)**

Purpose: ReLU introduces non-linearity to the model, which allows it to learn complex patterns. It is effective in hidden layers because it helps to avoid vanishing gradients and speeds up training.

- **Number of Epochs: 20**

Purpose: The number of epochs specifies how many times the entire training dataset is used to update the model weights. Fewer epochs might speed up training but could also lead to underfitting. This setting helps balance training time and model performance.

- **Batch Size: 300**

Purpose: The batch size defines how many samples are processed before the model's parameters are updated. A smaller batch size can provide more updates per epoch and might help generalize better, while also requiring less memory compared to very large batch sizes.

- **Early Stopping (Monitor): val_loss**

Purpose: Early stopping is used to monitor the validation loss and halt training when it no longer improves, helping to prevent overfitting and ensure that the model does not continue to train unnecessarily.

- **Early Stopping (Patience): 3**

Purpose: This is the number of epochs with no improvement in validation loss before stopping the training process. It provides a balance between giving the model enough time to improve and avoiding premature stopping.

- **Layer 1: Dense, 12 units, ReLU**

Purpose: The first hidden layer with 12 units and ReLU activation helps the model start learning patterns from the input data by introducing non-linearity.

- **Layer 2: Dense, 8 units, ReLU**

Purpose: The second hidden layer with 8 units continues the learning process by capturing more abstract and complex features, further building on the patterns identified by the first layer.

- **Layer 3: Dense, 4 units, ReLU**

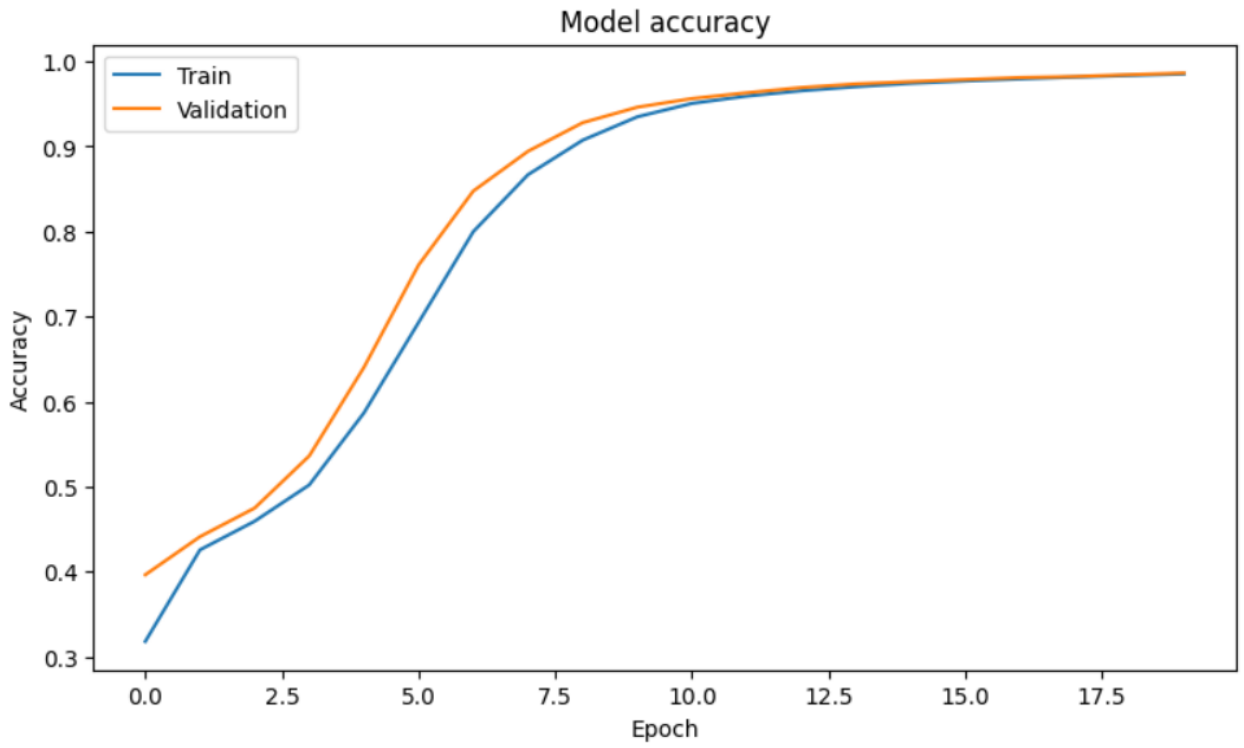
Purpose: The third hidden layer with 4 units further refines the feature representations and prepares the data for the final output layer.

- **Output Layer: Dense, 5 units, SoftMax**

Purpose: The output layer with 5 units and SoftMax activation converts the final hidden layer's output into probabilities for each class, facilitating the classification of the data into one of the predefined classes.

Summary: Including this detailed table and explanation helps document and communicate the choices made during model configuration. It serves as a reference for understanding the model's architecture and training parameters, making it easier to reproduce, analyze, or discuss the model with others.

Model Accuracy:



Objective: Achieve 98% accuracy in the Artificial Neural Network (ANN) model.

The image shows a plot depicting the model accuracy during training and validation.

Key observations:

- **Training and Validation Accuracy:** The blue line represents the training accuracy, while the orange line represents the validation accuracy.
- **Increasing Accuracy:** Both the training and validation accuracy increase over time (epochs). This is a positive sign as it indicates that the model is learning from the data and improving its performance.

- **Overfitting Potential:** The training accuracy is consistently higher than the validation accuracy. This might suggest that the model is overfitting, meaning it is learning the training data too well and might not generalize well to new, unseen data.
- **Gap Between Training and Validation:** The gap between the training and validation accuracy grows over time. This is another indicator of potential overfitting.

Possible Actions:

- **Regularization:** Techniques like L1 or L2 regularization can help prevent overfitting by penalizing large weights in the model.
- **Early Stopping:** If the validation accuracy stops improving after a certain number of epochs, training can be stopped to prevent overfitting.
- **Hyperparameter Tuning:** Adjusting hyperparameters like learning rate or batch size can also help improve model performance and prevent overfitting.

Additional Considerations:

- **Data Quality:** The quality of the training and validation data can significantly impact model performance.
- **Model Complexity:** A more complex model might be prone to overfitting.
- **Dataset Size:** A larger dataset can help prevent overfitting by providing more examples for the model to learn from.

Overall, the plot shows a promising trend of increasing accuracy but also raises concerns about potential overfitting. Further analysis and adjustments to the model and training process are necessary to ensure optimal performance.

Device Specifications

Device name	DESKTO _MEEFTGH
Processor	12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz
Installed RAM	16.0 GB (15.7 GB usable)
Device ID	0EB8B6E7-DE15-4E2B-A4E4-928B3ADD59E4
Product ID	00331-20313-54567-AA688
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows Specifications

Edition	Windows 11 Pro
Version 23H2	23H2
Installed on	9/18/2023
OS build	22631.4037
Experience	Windows Feature Experience Pack 1000.22700.1027.0

Python Version

Python version: 3.10.12

Libraries Version

NumPy version: 1.26.4

Pandas version: 2.1.4

Matplotlib version: 3.7.1

Missingno version: 0.5.2

Seaborn version: 0.13.1

SciPy version: 1.13.1

TensorFlow version: 2.17.0

Keras version: 3.4.1

Scikit-learn version: 1.3.2

Imbalanced-learn version: 0.12.3