A systematic Overview of Fundamentals and Methods of Business Intelligence

Ketan Gupta [1] and Nasmin Jiwani [2]
[1,2] Research Scientist, University of The Cumberlands, USA
ketan1722@gmail.com; nasminjiwani@gmail.com

ARTICLE INFO

Article History:
Received Nov 11, 2020
Revised March 31, 2021
Accepted Sep 15 , 2021

Keywords:
business intelligence, managers, academic journals, research

ABSTRACT

Access to relevant information and expertise is essential for organizations in the demanding business climate of today. Business intelligence (BI) is a general term for the tools, methods, and products that assist managers in comprehending the state of their company's operations. Additionally, BI technologies can help firms meet their informational knowledge demands. In this article, new investigations and papers from academic journals in this field are systematically reviewed to categorize and prioritize the ideas and techniques of business intelligence. This is done in response to the rising trend of BI research in BI concepts and applications. As a result, research on BI was categorized into three categories: managerial, technical, and system-enabled methods. Each category's specification and future research directions were presented.

IJSDCS

Introduction

Business pioneers may make spectacular and significant use of information thanks to business intelligence. Associations have a greater chance of influencing information for the upper hand when using the tools obtained from the BI segments. When used properly, the data may guide corporate decisions that can proactively respond to market trends and other external factors. While many businesses today collect and retain copious amounts of raw data, very few harness that data's power to create business insights and improvements. The only thing that is truly constant about business is that it is always changing. With the help of business intelligence, today's business pioneers may successfully improve their businesses and manage their organizations by making decisions based on the facts and gaining a more in-depth understanding of their prospects[1].

BI in real-time

This particular area of business intelligence (BI) is becoming more and more common in a mobile environment. A company can respond to recurring email patterns, informative frameworks, or even sophisticated presentations using programming applications. Because everything is ongoing, a businessperson may provide unique offerings that capitalize on the current situation. Promoting professionals may use data to create innovative limited-time deals, like a voucher for hot soup on a chilly day. CEOs may be interested in tracking the time of day and location of users when they connect to a website so that marketing may give exceptional developments gradually while the user is locked in on the page[2].

Database Management

Company innovators have the opportunity to examine interconnected segments that can assist drive business by filtering through subsets of information via information warehousing. Analyzing sales data over an extended period can aid in improving product development or adjusting one-time contributions. Information warehousing may also be used to examine business form insights, particularly how they relate to one another. For instance, business owners might compare shipment times across different offices to see which processes and teams function most effective overall. Additionally, storing enormous amounts of information in ways that are useful to multiple departments inside the business itself is a part of information warehousing[3].

Sources of Data

Various put-away information kinds are part of this BI component. It has to do with collecting the raw data and using programming tools to turn it into significant information sources that every division can utilize to significantly impact the company[4]. By employing this method, BI analysts may create information devices that allow data to be entered into a large collection of spreadsheets, pie charts, tables, or other charts that can be used for a variety of business objectives. For instance, information may be used to introduce people who can help to set realistic group goals. Examining the significance of information sources may also help organizations make decisions that are more grounded in reality and take into account a wider range of organizational needs.

Tools for Business Intelligence

Many tools are used to break down the many components of BI and turn them into actual problem-solving activities under the general phrase "business intelligence."

The current widespread use of the Internet and the extraordinarily inventive nature of our free-showcase economy have fostered the growth of niche markets and new businesses, as well as consulting organizations and other commercial endeavors that have contributed to the creation of countless BI gadgets[5]. These specific business tools can help entrepreneurs examine several areas of their firm in greater depth and detail. The most well-known tools in use today are business and information analysis, predictive analysis, cloud technology, portable business intelligence, Big Data consulting, and visual analytics [6].

IJSDCS

Business intelligence (BI) analysis's fundamental categories?
Organizations dealing with data should be aware of the four basic classifications into which business intelligence is often divided.

Reporting
Making archives with useful data and informing the reader about what happened are the main goals of reporting [7]. They often span a period set by the report's author and can include information on an organization's overall operations or be as simple as a week-by-week report that effectively evaluates your Facebook campaign.

Evaluation
The analysis looks at the reasons behind an event. This is a crucial component of BI since the information that isn't connected to anybody else is useless. When it was looked at and made into something we could understand and translate, it may have ended up being useful [8]. Three fundamental categories of analysis exist.
Spreadsheet analysis is the process of examining data found in spreadsheets to determine or anticipate the execution of large or explicit units. Consider using Excel to keep track of the number of hours your personnel works.
Software called Ad-Hoc Query enables users to create custom information queries. For instance, asking how many of a given item have been sold during a specific period.
Visualization tools are programs that take raw data and turn it into a visual display that clients can look at and understand. For instance, a pie chart that shows how clients contacted you over a month.

Watching
The ability to routinely examine information and data incrementally, or close to it, at any rate, is one of the more important aspects of BI. When making decisions or receiving portrayals in between reporting periods, this may be exciting. The following are the three basic types of monitoring:
Dashboard: A central location that houses all relevant and useful metrics and data. To make it easier for clients to read, they are frequently talked to visually. For instance, you can access the new Google+ dashboards by logging into your company's Google+ profile and selecting Dashboard from the drop-down menu on the left.
KPIs (Key Performance Indicators) KPIs track how well a specific activity or job is being carried out. Return on Investment, for instance (ROI).
Business performance management (BPM) is a framework designed to ensure that the execution goals for your organization or activities are accomplished and outcomes are communicated. For instance, obtaining several new clients. Some businesses and BI vendors will refer to this as the balanced scorecard.

Foresight

To try and predict what will happen based on the information now available and diverse trends, many businesses use BI methodologies. Prediction is one of the most amazing types of BI, which is why so many companies frequently hire companies to do it for them or rely on software that automates a large portion of the process. The two main categories of prediction are:

Data mining is the process of looking for patterns and connections in large informational groupings. Information mining's main goal is to delete or transform data into something we can understand and further use.
Any modeling that aims to predict the outcome of an action or the likelihood of a result is referred to as predictive modeling.

Data Analysis

To assist businesses in making more data-driven choices, business intelligence integrates business analytics, data mining, data visualization, data tools, infrastructure, and best practices. Let's see the case study to understand it better.

Case study

Import all necessary libraries, and give them aliases if needed. We will use NumPy for numerical analysis, pandas to treat data as data frames, and sqlite3 to query our data provided in the files as an SQLite Database.

```
import NumPy as np # linear algebra

import os

import pandas as PD # data processing

import sqlite3 as sq# enable the creation and handling of a local DB
```

Which aircraft has flown the most?

Assumption

When saying "flown the most" we refer to the number of flights and not accumulated distance flown.

Process

Look at the data in the individual_flights table, count how many individual flights there were per aircraft, and then enhance the result with the name of the aircraft instead of the code. We will provide the number of flights as well the name of the aircraft.

Create the query string to then feed into the Pandas read_sql function.

```
q1 = ('SELECT air.aircraft_type, COUNT(in.flight_id) AS number_of_flights '

    'FROM individual_flights AS indv '

    'JOIN aircraft AS air '

    'ON in.aircraft_id = aiairircraft_id '

    'GROUP BY 1 '
```

```
'ORDER BY number_of_flights DESC ')

# The function read_sql takes a query string and a database connection and performs the query.
r1 = pd.read_sql(q1,conn)

#We only need the first result, so we use iloc[[0]].
print(r1.iloc[[0]])
```

Result

```
Aircraft_Type  number_of_flights
0        Goose              1008
```

Answer

The aircraft that has flown the most (number of flights) is **Goose**, with 1008 flights.

Which aircraft has carried the most passengers compared to the cost?

We lack specific data on how many passengers flew on each trip, therefore for simplicity, we will assume it was the max capacity of each aircraft per flight.

The value we will consider as "most cost-efficient" or, "that has carried the most passengers compared to the cost" will be the smallest found value for "cost per passenger". There are 2 ways to express this answer:

A: passengers / GBP or B: GBP / passenger
By using A, we need to interpret "fraction of a passenger" because the values are likely to be decimals, even when mathematically correct, which makes noise for general interpretation and business communications. On the other hand, by using B, will be more aligned with common logic and more natural sounding.

This is to say, if expressed as A, we are interested in the biggest value, if expressed as B, we are interested in the smallest one.

We will provide both forms to illustrate.

```
#Answer form A.

q2 = ('SELECT airc.aircraft_type, CAST((COUNT(indv.flight_id)*airc.capacity) AS FLOAT)/CAST(airc.cost AS FLOAT) AS passengers_per_cost '
    'FROM individual_flights AS indv '
    'JOIN aircraft AS air '
    'ON in.aircraft_id = aiairircraft_id '
    'GROUP BY 1 '
    'ORDER BY passengers_per_cost DESC')

#Answer form B.
q3 = ('SELECT air.aircraft_type, CAST(air. cost AS FLOAT)/CAST((COUNT(in.flight_id)*aiairapacity) AS FLOAT) AS cost_per_passenger '
    'FROM individual_flights AS indv '
    'JOIN aircraft AS air '
    'ON in.aircraft_id = aiairircraft_id '
    'GROUP BY 1 '
```

```
    'ORDER BY cost_per_passenger ASC')

r2 = pd.read_sql(q2,conn)

r3 = pd.read_sql(q3,conn)

print('Form A, passengers/GBP: \n')
print(r2.iloc[:3])

print('\n')
print('Form B, GBP/passenger: \n')
print(r3.iloc[:3])
```

```
Form A, passengers/GBP:

  Aircraft_Type  passengers_per_cost
0         Goose             0.168000
1     Miniflock             0.027700
2    Thundercat             0.002765


Form B, GBP/passenger:

  Aircraft_Type  cost_per_passenger
0         Goose            5.952381
1     Miniflock           36.101083
2    Thundercat          361.663653
```

Answer

Aircraft has carried the most passengers compared to the cost is **Goose**.

Which airport has transported the most passengers through it?

Assumption

Assume, again, max capacity per flight. This would be better expressed as "potentially most passengers transported"

Process

Calculate the number of passengers transported as the number of flights of each aircraft type, times its capacity, all together grouped by the airport.

It is worth mentioning that every individual flight will transport N passengers, but that will double count the number of people because it will be attributed to both airports (departure and destination) per flight, therefore, N would be attributed to airport A and airport B.

We will leave the number in the answer query.

```
# Adding an OR in the JOIN ON clause will account for inbound and outbound flights all alike.
q4 = ("""
   SELECT airport_name
    , SUM(n_passenger_per_aircraft) AS n_passengers
    FROM (
       SELECT indef.aircraft_id
       , airp.airport_name
       , COUNT(DISTINCT flight_id)*a.capacity AS n_passenger_per_aircraft
       FROM individual_flights AS indvf
       JOIN airports AS airp
```

```
        ON airp.airport_code = indvf.destination_airport_code
          OR air.airport_code = indef.departure_airport_code
      JOIN aircraft AS a
        ON a.aircraft_id = indef.aircraft_id
      GROUP BY 1, 2)
    GROUP BY 1
    ORDER BY 2 DESC""")

r4 = pd.read_sql(q4,conn)

print('Number of passengers per airport (inbound and outbound) \n')
print(r4.iloc[:3])
Number of passengers per airport (inbound and outbound)

      airport_name  n_passengers
0  Amazon Mothership       2423400
1   Nestland Airport       1999700
2         Flocktopia       1685200
```

Answer

The airport that has transported more passengers through it is **Amazon Mothership**.

Is there a difference when considering outbound and inbound passenger flow?

Parting from the previous results, analyze if slicing it by outbound or inbound would yield different results.

```
# Only count if outbound from airport X
q5 = ("""SELECT airport_name
    , SUM(n_outbound_passengers) AS outbound_passengers
    FROM (
        SELECT indef.aircraft_id
      , airp.airport_name
      , COUNT(DISTINCT flight_id)*a.capacity AS n_outbound_passengers
      FROM individual_flights AS indvf
      JOIN airports AS airp
      ON air.airport_code = indef.departure_airport_code
      JOIN aircraft AS a
      ON a.aircraft_id = indef.aircraft_id
      GROUP BY 1, 2)
     GROUP BY 1
     ORDER BY 2 DESC""")

# Only count if inbound from airport X
q6 = ("""SELECT airport_name
    , SUM(n_inbound_passengers) AS inbound_passengers
    FROM (
      SELECT indef.aircraft_id
      , airp.airport_name
      , COUNT(DISTINCT flight_id)*a.capacity AS n_inbound_passengers
      FROM individual_flights AS indvf
      JOIN airports AS airp
      ON air.airport_code = indef.destination_airport_code
      JOIN aircraft AS a
      ON a.aircraft_id = indef.aircraft_id
      GROUP BY 1, 2)
    GROUP BY 1
    ORDER BY 2 DESC""")

r5 = pd.read_sql(q5,conn)
print('Number of passengers per airport (outbound) \n')
print(r5.iloc[:3])
```

```
print('\n')

r6 = PD.read_sql(q6,conn)
print('Number of passengers per airport (inbound) \n')
print(r6.iloc[:3])
Number of passengers per airport (outbound)

        airport_name  outbound_passengers
0  Amazon Mothership              1406000
1    Nestland Airport             1222500
2          Flocktopia              428900


Number of passengers per airport (inbound)

        airport_name  inbound_passengers
0          Flocktopia             1256300
1  Amazon Mothership             1017400
2    Nestland Airport             777200
```

Answer

For outbound flights, the airport that has transported more passengers through it is **Amazon Mothership**, but for inbound flights, it would be Flocktopia.

Is there a difference when considering the size of the airports?

Parting from the first result, we need to account for the size of the airport and calculate a ratio.

Process

The total number of passengers per aircraft, the sum for all aircraft that passed by each airport, and divided by the size of the airport.

```
q7 = ("""SELECT airport_name
     , CAST(SUM(n_passenger_per_aircraft) AS FLOAT)/CAST(airport_size AS FLOAT) AS passagers_per
_m2
     FROM (
        SELECT indef.aircraft_id
        , airp.airport_name
        , airp.airport_size
        , COUNT(DISTINCT flight_id)*a.capacity AS n_passenger_per_aircraft
        FROM individual_flights AS indvf
        JOIN airports AS airp
           ON airp.airport_code = indvf.destination_airport_code
              OR air.airport_code = indef.departure_airport_code
        JOIN aircraft AS a
           ON a.aircraft_id = indef.aircraft_id
        GROUP BY 1, 2, 3)
     GROUP BY 1
     ORDER BY 2 DESC""")

r7 = PD.read_sql(q7,conn)
print(r7.iloc[[0]])
        airport_name  passagers_per_m2
0  Amazon Mothership            242.34
```

Answer

Amazon Mothership is the airport with more potential passenger traffic about its size.

What was the best year for Revenue Passenger-Miles for each

airline?

The fields that track international values have frequent null values. We will consider them as zeros for simplicity but that must be accounted for before doing any kind of analysis because that will bias the results.

Process

Calculate the SUM of all RPM (Domestic, International, or both at the same time, so, Total), and then select the MAX per airline, which will yield us the year as well. By selecting a specific type of RPM or both we will obtain different answers, therefore all three must be provided.

Considering RPM Domestic and RPM International separated:

```
q8 = ("""SELECT Airline_Name, Year, MAX(RPM_Domestic)
  FROM (
    SELECT Airline_Code, CAST(substr(Date, -4) AS INT) AS Year
    , SUM(COALESCE(RPM_Domestic,0)) AS RPM_Domestic
    FROM flight_summary_data
    GROUP BY 1, 2) AS sub
  JOIN
    airlines
  ON subq.airline_code = airlines.airline_code
  GROUP BY 1""")

r8 = pd.read_sql(q8,conn)

print(r8)

print('\n')

q9 = ("""SELECT Airline_Name, Year, MAX(RPM_International)
  FROM (
    SELECT Airline_Code, CAST(substr(Date, -4) AS INT) AS Year
    , SUM(COALESCE(RPM_International,0)) AS RPM_International
    FROM flight_summary_data
    GROUP BY 1, 2) AS sub
  JOIN
    airlines
  ON subq.airline_code = airlines.airline_code
  GROUP BY 1""")

r9 = pd.read_sql(q9,conn)

print(r9)
   Airline_Name  Year  MAX(RPM_Domestic)
0  Amazon Airlines  2015          9175044
1       Flock Air  2016         13405774
2   Goose Airways  2016         34637841


   Airline_Name  Year  MAX(RPM_International)
0  Amazon Airlines  2016              2792855.0
1       Flock Air  2016              3912894.0
2   Goose Airways  2015             15219579.0
```

Considering RPM Domestic and RPM International all together

```
q10 = ("""SELECT Airline_Name, Year, MAX(RPM_Total)
  FROM (
    SELECT Airline_Code, CAST(substr(Date, -4) AS INT) AS Year
    , SUM((RPM_Domestic + COALESCE(RPM_International,0))) AS RPM_Total
```

```
    FROM flight_summary_data
    GROUP BY 1, 2) AS sub
  JOIN
    airlines
  ON subq.airline_code = airlines.airline_code
  GROUP BY 1""")

r10 = pd.read_sql(q10,conn)

print(r10)
```

Answer

The best years for each airline are the following, for the different types of RPM.

Domestic RPM:

Amazon Airlines 2015

Flock Air 2016

Goose Airways 2016

International RPM:

Amazon Airlines 2016

Flock Air 2016

Goose Airways 2015

Total RPM:

Amazon Airlines 2015

Flock Air 2016

Goose Airways 2016

What **was** the number of passenger miles each airline recorded?

Assumption

When talking about several passenger miles, it is referring to RPM, since that is the metric that most closely represents what is asked. We will answer the question as if it was made "What was the total RPM each airline recorded?".

```
q11 = ("""SELECT Airline_Name, SUM((RPM_Domestic + COALESCE(RPM_Internatio
nal,0))) AS RPM_Total
      FROM flight_summary_data
      JOIN airlines
      ON flight_summary_data.airline_code = airlines.airline_code
      GROUP BY 1 ORDER BY RPM_Total""")

r11 = pd.read_sql(q11,conn)
```

```
print(r11)
     Airline_Name    RPM_Total
0  Amazon Airlines  148321579.0
1       Flock Air  173421998.0
2   Goose Airways  575923428.0
```

How **do** domestic passenger flow and international passenger flow vary by month of the year for Flocktopia airport?

A single plot summarising this would be useful

Assumption

Flow is the number of, because the flow is defined as volume/time unit, and in this case, we do not have that kind of specific information.

Assume the question was asked as frame above because otherwise, it would have required to answer "if a single plot summarising this would be useful", and in which case the answer would be no, but for reasons such as several points and visibility, seasonality and other variants that might be affecting the variation of flow.

Notes

SQLite works differently and it is somehow more limited to dealing with dates and formats, so the solution presented here is not elegant or best practice.

```
q12 = """SELECT SUM(passengers_domestic) AS total_passengers_domestic
, SUM(passengers_international) AS total_passengers_international
, CAST(substr(Date, -4) AS INT) AS Year
, CAST(substr(Date, 4, 2) AS INT) AS Month
FROM flight_summary_data WHERE Airport_Code = "FKT"
GROUP BY 3, 4
ORDER BY Year ASC, Month ASC"

r12 = pd.read_sql(q12,conn)

r12 = r12.set_index(['Year','Month']).diff()

print(r12.plot(figsize=(18,10)))
AxesSubplot(0.125,0.125;0.775x0.755)
```

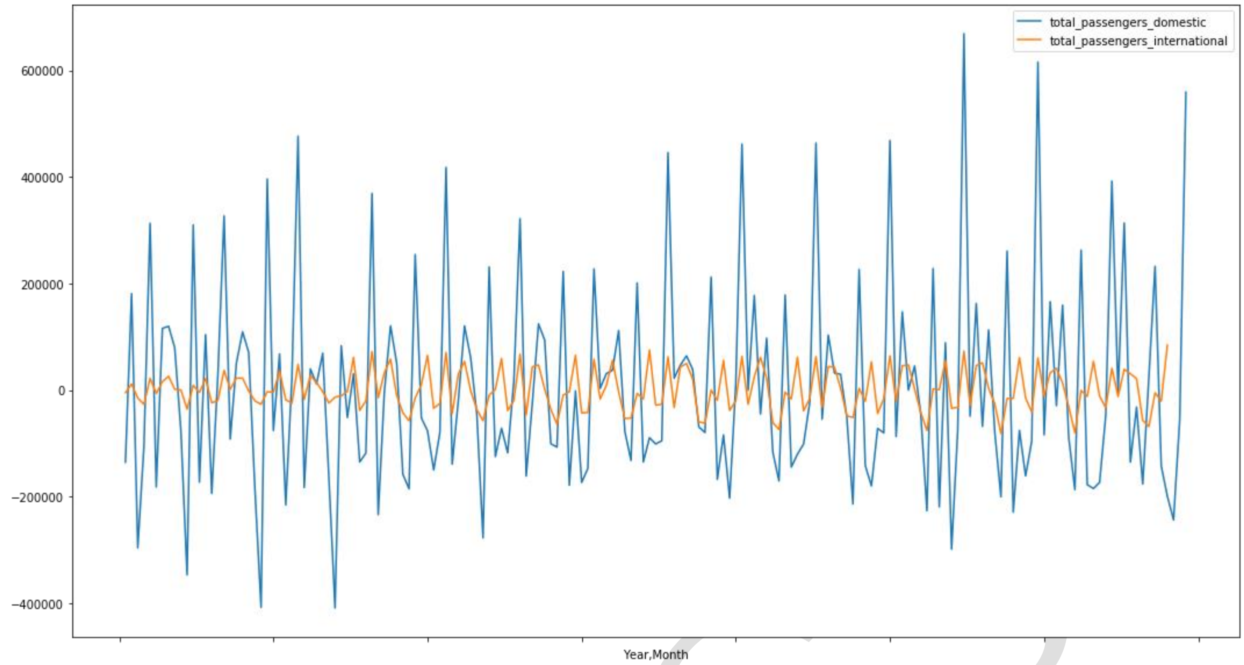

Figure 1Number of Passengers w.r.t. time

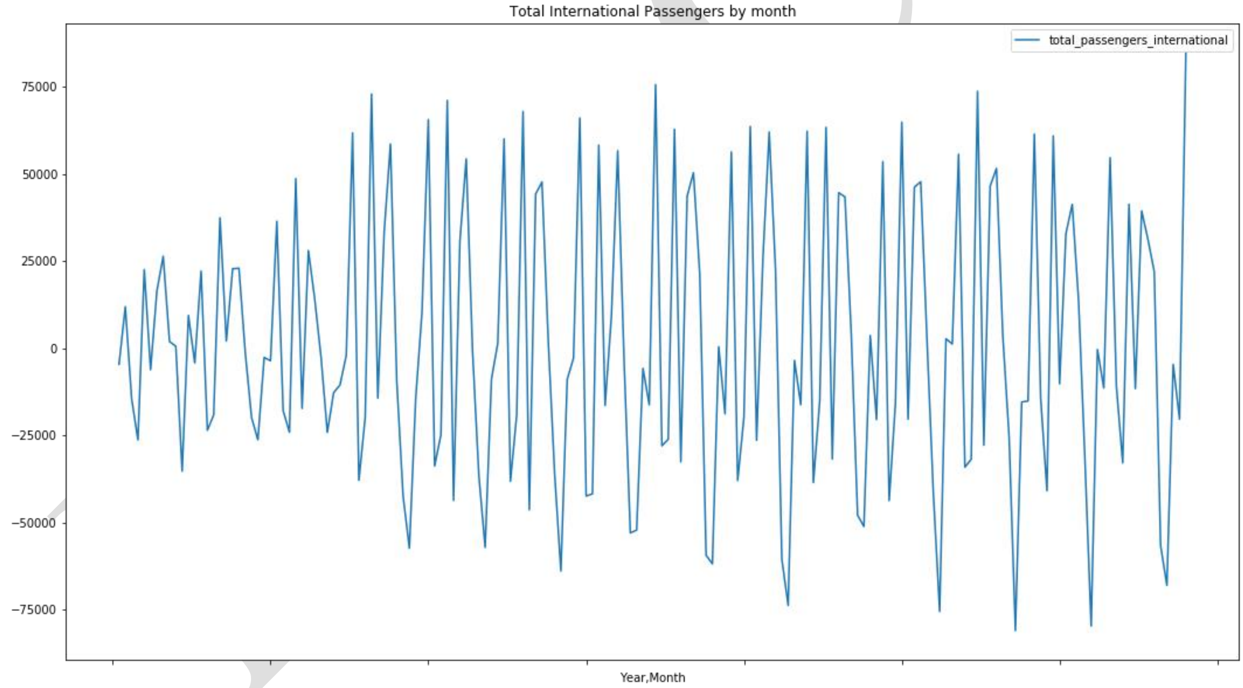

Figure 2 Total International Passengers every month

What was the best year for growth for each airline?
Process
Define what is to be considered as "Growth", then split what the factors of Growth would b e, compose an index for growth, and then compare all growth indexes to see which one did better.

We have ASM and RPM, as well the number of flights Domestic and for some flights Inter national, per Airline and Airport. To list some of the possible ways to evaluate growth, we have: Decrease the difference between ASM - RPM, therefore captivating more passengers out of the ones that they had the capacity for. Increase on ASM, therefore increasing the fle et or the size of the planes, or the number of flights. Increase in several flights alone.

For simplicity and because it seems to be the one providing the best answer, we will use A SM as our growth indicator, and the more ASM an airline has over time, the more we will say they grew.

To calculate the growth we will take the AVG(ASM_Domestic) per Airline per Year.

Assumption
Assume that all airlines have ASM_Domestic for each the same number of months because otherwise might be a little biased due to how the average is calculated. To simplify even fu rther for time restrictions I will assume the MAX ASM will be the latest from the 2017 and 2002 averages, and the min the other one. Seems logical but something of that sort should not be done lightly.

Note
We would account only for ASM_Domestic since ARM_International has a lot of null valu es and will bias the results, but the process would be similar to the one applied before for a prior question involving RPM.

```
q13 = ("""
SELECT SUM(ASM_Domestic) AS sum_asm_domestic
, Airline_Code
, CAST(substr(Date, -4) AS INT) AS Year
FROM flight_summary_data
GROUP BY 2, 3
HAVING CAST(substr(Date, -4) AS INT) <> 2002
""")

Airport query = ("""
SELECT * FROM airports
""")

airportsdf = pd.read_sql(airportsquery,conn)

r13 = PD.read_sql(q13,conn)
# We get the difference between each year for each airline so we can see how much they "g
rew" from the past year. We group by Airline_Code first.
r13['dif'] = r13.groupby(['Airline_Code'])['sum_asm_domestic'].diff()

# Find the indexes of each of the max values for dif. It will have negative values, but since
we are talking about growth they will not be included.
# According to the way we calculate the dif, we need to exclude the year 2002 because it o
nly has a couple of months.
idx = r13.groupby(['Airline_Code'], sort=False)['dif'].transform(max) == r13['dif']

# We print the years with the Max dif.
print(r13[idx])
    sum_asm_domestic Airline_Code  Year        dif
9          10223097           AA  2012  3222153.0
28          15381195           FA  2016  2126154.0
```

```
37         31878820         GA  2010  5460567.0
```

Answer

Goose Airways 2010, Flock Air 2016, and Amazon Airlines 2012

Which airport contributed the most to this?

```
q14 = ("""
SELECT SUM(ASM_Domestic) AS sum_asm_domestic
, Airline_Code
, Airport_Code
, CAST(substr(Date, -4) AS INT) AS Year
FROM flight_summary_data
GROUP BY 2, 3, 4
HAVING (Airline_Code='FA'AND CAST(substr(Date, -4) AS INT)=2016)
OR (Airline_Code='AA'AND CAST(substr(Date, -4) AS INT)=2012)
OR (Airline_Code='GA'AND CAST(substr(Date, -4) AS INT)=2010)
""")
# Cherrypicking with very bad practice in the query the specific years in which I obtained t
he max growth. This should never be made like this.

r14 = PD.read_sql(q14,conn)
print(airports)

# Set and index frame to extract the max value'd airport per airline in the desired year.
indx2 = r14.groupby(['Airline_Code'], sort=True)['sum_asm_domestic'].transform(max) =
= r14['sum_asm_domestic']

# Before printing, join with the airport table to get the airport names.
print(PD.merge(r14[indx2], airports, how='inner', left_on='Airport_Code', right_on='Airpo
rt_Code')[['Airline_Code','Airport_Name','sum_asm_domestic']])
   index Airport_Code      ...      Airport_Employees  Airport_Size
0      0          FKT      ...              1000000       2000000
1      1          NSA      ...                20000         50000
2      2          AMP      ...               100000         10000

[3 rows x 5 columns]
  Airline_Code      Airport_Name  sum_asm_domestic
0           AA  Nestland Airport          8725411
1           FA  Nestland Airport         11400523
2           GA        Flocktopia         22483550
```

Answer

For Amazon Airlines, it was Nestland Airport, for Flock Air it was Nestland Airport as well, and for Goose Airways it was Flocktopia.

Result and Conclusion

From the above case study, it is proved that BI technologies can help firms meet their informational knowledge demands. This will help the organization to take quick and intelligent decisions which will result in optimum use of the resources in other words Revenue of the organization. This research study helps the researchers who are working in the same field.

References

Adomavicius, G., and Tuzhilin, A. 2005. “Toward the Next Generation of Recommender Systems: A Survey of the State-of the-Art and Possible Extensions,” IEEE Transactions on Knowledge and Data Engineering (17:6), pp. 734-749.

Anderson, C. 2004. “The Long Tail,” WIRED Magazine (12:10) (http://www.wired.com/wired/archive/12.10/tail.html).

Associated Press. 2012. “Columbia U Plans New Institute for Data Sciences,” July 30 (http://www.cbsnews.com/8301-505245_162- 57482466/Columbia-u-plans-new-institute-for-data-sciences/,accessed August 3, 2012).

Barabási, A. 2003. Linked: How Everything Is Connected to Everything Else and What it Means for Business, Science, andEveryday Life, New York: Plume.

Batagelj, V., and Mrvar, A. 1998. “Pajek: A Program for Large Network Analysis,” Connections (21), pp. 47-57.

Bettencourt, L. M.A., Cintrón-Arias, A., Kaiser, D. I., and CastilloChávez, C. 2006. “The Power of a Good Idea: Quantitative Modeling of the Spread of Ideas from Epidemiological Models,” Physica A (364), pp. 513-536.

Bitterer, A. 2011. “Hype Cycle for Business Intelligence,” Gartner, Inc., Stamford, CT.
Blei, D. M. 2012. “Probabilistic Topic Models,” Communications of the ACM (55:4), pp. 77-84

IJSDCS