

Prediction of Animal Strike on US Commercial Flights

Final Paper for the CEU MSc in Business Analytics program

Gábor Horváth

June 2017



1 Introduction

The structure of the document follows the Cross Industry Standard Process for Data Mining (CRISP-DM) process model, which is a non-proprietary, documented, and freely available data mining model (Shearer 2000). Whenever the model sections can be matched to (and can fulfill) the requirements stated by CEU for the Final Paper I'm using the appropriate section identified by the CRIPS-DM model. Please keep in mind that the model supports the full end-to-end process of a data mining project, but the project does not require the use of all the model elements.

2 Business Understanding

2.1 Determine Business Objectives

2.1.1 Business Objectives

There are two main objectives what the project is aiming to complete.

1. Create a statistical analysis to identify those reasons (based on the data available), which are determining the the risk of an animal strike for an airport.
2. Create a prediction model, which can be used to predict the risk of an animal strike for a given flight.

The result of the statistical analysis could be used in the completion of the model building and evaluation the recommended order of the completion is the order of the objectives stated above.

2.1.2 Business Success Criteria

- Identification of features determining the risk potential of an airport
- Working model for animal strike prediction

2.2 Assess Situation

2.2.1 Inventory of resources

- Flight Data
- Animal Strike Data
- R
- Buckets

2.2.2 Requirements, Assumptions, and Constraints

- Additional requirements:
 - No additional requirements identified on top of the requirements already stated in this document.
- Assumptions
 - No initial assumptions made.
- Constraints
 - No initial hard constraints identified.

2.2.3 Risks and Contingencies

- Risks
 - No initial risks identified
- Contingencies
 - No initial contingencies identified

2.2.4 Terminology

The project is using different terminologies from the different domains. The terms/definitions used will not be marked or explained in details, if based on the context the reader can easily identify the domain of the particular term. In case there are uncertainties about a term (and it's not explained in the paper), the following sources can be used for the definitions:

-
- Aviation:
 - Aviation Terms / Directory: <http://www.aviation-terms.com/index2.php>
 - Aviation Glossary: <http://www.aerofiles.com/glossary.html>
 - Aviation Glossaries: https://www.flightsimaviation.com/_glossaries.html?s=aviation_terms
 - Data Mining
 - Data Mining Glossary: <http://www.thearling.com/glossary.htm>
 - Data Mining - Terminologies: https://www.tutorialspoint.com/data_mining/dm_terminologies.htm
 - Data Mining and Predictive Analytics Glossary: <http://www.kdnuggets.com/2015/06/data-mining-predictive-analytics-glossary.html>
 - Data Science / Big Data
 - Data Science Glossary: <http://www.datascienceglossary.org/>
 - Analytics and Big Data Glossary: <http://data-informed.com/glossary-of-big-data-terms/>
 - Data Science Glossary: <http://www.kdnuggets.com/2015/09/data-science-glossary.html>

2.2.5 Costs and Benefits

This is a one-man project, no significant cost is expected. Main benefit is to put to and almost end-to-end scenario the topics covered during the courses and discovering bits and bolts of the techniques for creating the project.

2.3 Determine Data Mining Goals

2.3.1 Data Mining Goals

- Understand, Analyse, Clean and Merge the source data correctly
- Create the required attributes
- Generate the required records (if applicable)

2.3.2 Data Mining Success Criteria

- Identification of featured determining the risk potential of an airport
- Working model for animal strike prediction

2.4 Produce Project Plan

2.4.1 Project Plan

The project is managed in an agile way, where all the tasks, requirements, issues, solutions, and ideas are kept in a project at [buckets](#).

2.4.2 Initial Assessment of Tools and Techniques

- Programming language:
 - R: <https://www.r-project.org/>
 - IDE for the programming language:
 - RStudio: <https://www.rstudio.com/>
 - Documentation is created using:
 - knitr: <https://yihui.name/knitr/>
 - MiKTeX: <https://miktex.org/>
 - ReporteRs: <https://cran.r-project.org/web/packages/ReporteRs/index.html>
 - Data visualization:
-

-
- ggplot2: <http://ggplot2.org/>
 - Data manipulation:
 - access2csv: <https://github.com/AccelerationNet/access2csv>
 - dtplyr: <https://cran.r-project.org/web/packages/dtplyr/index.html>
 - Project plan / task management:
 - Buckets: <https://www.buckets.co/>
 - Source code repository:
 - GitHub: <https://github.com/>

Note: The list above do not contain the list of all the tools and packages used to create the project, but the full list will be provided in the source code.

3 Data Understanding

3.1 Collect Initial Data

3.1.1 Initial Data Collection Report

There have been two data sources acquired in the initial phase of the project. These sources are the following:

3.1.1.1 Federal Aviation Administration

- Data source: [Wildlife Strike Database](#)
- The FAA provides the database as a compressed Microsoft Access file.
- The database version used is Version 2016.4-P (as of 24-10-2016).
- The database contains 180,177 Strike Reports from 1-1-1990 through 30-4-2016.
- The compressed file size is 44,730,852 bytes.
- The uncompressed Microsoft Access database file size is 193,495,040 bytes.
- The extracted tables are:
 - STRIKE_REPORTS (1990-1999) - 30082 rows - CSV size is 21,523,668 bytes
 - STRIKE_REPORTS (2000-2009) - 69960 rows - CSV size is 51,833,820 bytes.
 - STRIKE_REPORTS (2010-Current) - 70577 rows - CSV size is 53,973,874 bytes.
 - STRIKE_REPORTS_BASH (1990-Current).csv - 8046 rows - CSV size is 5,412,394 bytes.

3.1.1.2 United States Department of Transportation

- Data source: [Bureau of Transportation Statistics](#)
- The BTS provides the database as separate compressed CSV files. One file contains data of one month.
- The timestamp of the first CSV file available is 1-1-1987.
- The timestamp of the first data available is 1-10-1987.
- The timestamp of the last data acquired from BTS in the project is 31-12-2016.
- The number of files is 360.
 - Compressed size of the files is 6,196,385,360 bytes.
 - Uncompressed size of the files is 71,146,030,010 bytes.
- The download speed of the public access to these files seems to be limited, which needs to be taken into account in case of reproducing the results.

3.2 Describe Data

3.2.1 Data Description Report

The two main data sources have the following column explanations, which is attached to the downloaded files as well, by the data provider agencies.

3.2.1.1 Animal strike data

| Column name | Explanation of Column Name and Codes |
|-------------|---|
| INDEX NR | Individual record number |
| OPID | Airline operator code |
| OPERATOR | A three letter International Civil Aviation Organization code for aircraft operators. (BUS = business, PVT = private aircraft other than business, GOV = government aircraft, MIL - military aircraft.) |
| ATYPE | Aircraft |

| Column name | Explanation of Column Name and Codes |
|----------------------|---|
| AMA | International Civil Aviation Organization code for Aircraft Make |
| AMO | International Civil Aviation Organization code for Aircraft Model |
| EMA | Engine Make Code (see Engine Codes tab below) |
| EMO | Engine Model Code (see Engine Codes tab below) |
| AC_CLASS | Type of aircraft (see Aircraft Type tab below) |
| AC_MASS | 1 = 2,250 kg or less: 2 = ,2251-5700 kg: 3 = 5,701-27,000 kg: 4 = 27,001-272,000 kg: 5 = above 272,000 kg |
| NUM_ENGS | Number of engines |
| TYPE_ENG | Type of power A = reciprocating engine (piston): B = Turbojet: C = Turboprop: D = Turbofan: E = None (glider): F = Turboshift (helicopter): Y = Other |
| ENG_1_POS | Where engine # 1 is mounted on aircraft (see Engine Position tab below) |
| ENG_2_POS | Where engine # 2 is mounted on aircraft (see Engine Position tab below) |
| ENG_3_POS | Where engine # 3 is mounted on aircraft (see Engine Position tab below) |
| ENG_4_POS | Where engine # 4 is mounted on aircraft (see Engine Position tab below) |
| REG | Aircraft registration |
| FLT | Flight number |
| REMAINS_COLLECTED | Indicates if bird or wildlife remains were found and collected |
| REMAINS_SENT | Indicates if remains were sent to the Smithsonian Institution for identification |
| INCIDENT_DATE | Date strike occurred |
| INCIDENT_MONTH | Month strike occurred |
| INCIDENT_YEAR | Year strike occurred |
| TIME_OF_DAY | Light conditions |
| TIME | Hour and minute in local time |
| AIRPORT_ID | International Civil Aviation Organization airport identifier for location of strike whether it was on or off airport |
| AIRPORT | Name of airport |
| STATE | State |
| FAAREGION | FAA Region where airport is located |
| ENROUTE | If strike did not occur on approach, climb, landing roll, taxi or take-off, aircraft was enroute. This shows location. |
| RUNWAY | Runway |
| LOCATION | Various information about aircraft location if enroute or airport where strike evidence was found. Some locations show the two airports for the flight departure and arrival if pilot was unaware of the strike. |
| HEIGHT | Feet Above Ground Level |
| SPEED | Knots (indicated air speed) |
| DISTANCE | Miles from airport |
| PHASE_OF_FLT | Phase of flight during which strike occurred |
| DAMAGE | |
| Blank | Unknown |
| M = minor | When the aircraft can be rendered airworthy by simple repairs or replacements and an extensive inspection is not necessary. |
| M? = uncertain level | The aircraft was damaged, but details as to the extent of the damage are lacking. |
| S = substantial | When the aircraft incurs damage or structural failure which adversely affects the structure strength, performance or flight characteristics of the aircraft and which would normally require major repair or replacement of the affected component. |
| D = Destroyed | When the damage sustained makes it inadvisable to restore the aircraft to an airworthy condition. |

| Column name | Explanation of Column Name and Codes |
|---------------|--|
| STR_RAD | Struck radome |
| DAM_RAD | Damaged radome |
| STR_WINDSHLD | Struck windshield |
| DAM_WINDSHLD | Damaged windshield |
| STR_NOSE | Struck nose |
| DAM_NOSE | Damaged nose |
| STR_ENG1 | Struck Engine 1 |
| DAM_ENG1 | Damaged Engine 1 |
| STR_ENG2 | Struck Engine 2 |
| DAM_ENG2 | Damaged Engine 2 |
| STR_ENG3 | Struck Engine 3 |
| DAM_ENG3 | Damaged Engine 3 |
| STR_ENG4 | Struck Engine 4 |
| DAM_ENG4 | Damaged Engine 4 |
| INGESTED | Engine ingested the bird/ animal |
| STR_PROP | Struck Propeller |
| DAM_PROP | Damaged Propeller |
| STR_WING_ROT | Struck Wing or Rotor |
| DAM_WING_ROT | Damaged Wing or Rotor |
| STR_FUSE | Struck Fuselage |
| DAM_FUSE | Damaged Fuselage |
| STR_LG | Struck Landing Gear |
| DAM_LG | Damaged Landing Gear |
| STR_TAIL | Struck Tail |
| DAM_TAIL | Damaged Tail |
| STR_LGHTS | Struck Lights |
| DAM_LGHTS | Damaged Lights |
| STR_OTHER | Struck Other than parts shown above |
| DAM_OTHER | Damaged Other than parts shown above |
| OTHER_SPECIFY | What part was struck other than those listed above |
| EFFECT | Effect on flight |
| EFFECT_OTHER | Effect on flight other than those listed on the form |
| SKY | Type of cloud cover, if any |
| PRECIP | Precipitation |
| SPECIES_ID | International Civil Aviation Organization code for type of bird or other wildlife |
| SPECIES | Common name for bird or other wildlife |
| BIRDS_SEEN | Number of birds/wildlife seen by pilot |
| BIRDS_STRUCK | Number of birds/wildlife struck |
| SIZE | Size of bird as reported by pilot is a relative scale. Entry should reflect the perceived size as opposed to a scientifically determined value. If more than one species was struck, larger bird is entered. |
| WARNED | Pilot warned of birds/wildlife |
| COMMENTS | As entered by database manager. Can include name of aircraft owner, types of reports received, updates, etc. |
| REMARKS | Most of remarks are from the form but some are data entry notes and are usually in parentheses. |
| AOS | Time aircraft was out of service in hours. If unknown, it is blank. |
| COST_REPAIRS | Estimated cost of repairs of replacement in dollars (USD) |
| COST_OTHER | Estimated other costs, other than those in previous field in dollars (USD). May include loss of revenue, hotel expenses due to flight cancellation, costs of fuel dumped, etc. |

| Column name | Explanation of Column Name and Codes |
|-----------------------|--|
| COST_REPAIRS_INFL_ADJ | Costs adjusted for inflation |
| COST_OTHER_INFL_ADJ | Other cost adjusted for inflation |
| REPORTED_NAME | Name(s) of person(s) filing report |
| REPORTED_TITLE | Title(s) of person(s) filing report |
| REPORTED_DATE | Date report was written |
| SOURCE | Type of report. Note: for multiple types of reports this will be indicated as Multiple. See “Comments” field for details |
| PERSON | Only one selection allowed. For multiple reports, see field “Reported Title” |
| NR_INJURIES | Number of people injured |
| NR_FATALITIES | Number of human fatalities |
| LUPDATE | Last time record was updated |
| TRANSFER | Unused field at this time |
| INDICATED_DAMAGE | Indicates whether or not aircraft was damaged |

3.2.1.2 Flight data

| Column name | Explanation of Column Name and Codes |
|--------------------|---|
| Year | Year |
| Quarter | Quarter (1-4) |
| Month | Month |
| DayofMonth | Day of Month |
| DayOfWeek | Day of Week |
| FlightDate | Flight Date (yyyymmdd) |
| UniqueCarrier | Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years. |
| AirlineID | An identification number assigned by US DOT to identify a unique airline (carrier). A unique airline (carrier) is defined as one holding and reporting under the same DOT certificate regardless of its Code, Name, or holding company/corporation. |
| Carrier | Code assigned by IATA and commonly used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the Unique Carrier Code. |
| TailNum | Tail Number |
| FlightNum | Flight Number |
| OriginAirportID | Origin Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused. |
| OriginAirportSeqID | Origin Airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time. |
| OriginCityMarketID | Origin Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market. |
| Origin | Origin Airport |
| OriginCityName | Origin Airport, City Name |
| OriginState | Origin Airport, State Code |
| OriginStateFips | Origin Airport, State Fips |
| OriginStateName | Origin Airport, State Name |
| OriginWac | Origin Airport, World Area Code |

| Column name | Explanation of Column Name and Codes |
|----------------------|--|
| DestAirportID | Destination Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused. |
| DestAirportSeqID | Destination Airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time. |
| DestCityMarketID | Destination Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market. |
| Dest | Destination Airport |
| DestCityName | Destination Airport, City Name |
| DestState | Destination Airport, State Code |
| DestStateFips | Destination Airport, State Fips |
| DestStateName | Destination Airport, State Name |
| DestWac | Destination Airport, World Area Code |
| CRSDepTime | CRS Departure Time (local time: hhmm) |
| DepTime | Actual Departure Time (local time: hhmm) |
| DepDelay | Difference in minutes between scheduled and actual departure time. Early departures show negative numbers. |
| DepDelayMinutes | Difference in minutes between scheduled and actual departure time. Early departures set to 0. |
| DepDel15 | Departure Delay Indicator, 15 Minutes or More (1=Yes) |
| DepartureDelayGroups | Departure Delay intervals, every (15 minutes from <-15 to >180) |
| DepTimeBlk | CRS Departure Time Block, Hourly Intervals |
| TaxiOut | Taxi Out Time, in Minutes |
| WheelsOff | Wheels Off Time (local time: hhmm) |
| WheelsOn | Wheels On Time (local time: hhmm) |
| TaxiIn | Taxi In Time, in Minutes |
| CRSArrTime | CRS Arrival Time (local time: hhmm) |
| ArrTime | Actual Arrival Time (local time: hhmm) |
| ArrDelay | Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers. |
| ArrDelayMinutes | Difference in minutes between scheduled and actual arrival time. Early arrivals set to 0. |
| ArrDel15 | Arrival Delay Indicator, 15 Minutes or More (1=Yes) |
| ArrivalDelayGroups | Arrival Delay intervals, every (15-minutes from <-15 to >180) |
| ArrTimeBlk | CRS Arrival Time Block, Hourly Intervals |
| Cancelled | Cancelled Flight Indicator (1=Yes) |
| CancellationCode | Specifies The Reason For Cancellation |
| Diverted | Diverted Flight Indicator (1=Yes) |
| CRSElapsedTime | CRS Elapsed Time of Flight, in Minutes |
| ActualElapsedTime | Elapsed Time of Flight, in Minutes |
| AirTime | Flight Time, in Minutes |
| Flights | Number of Flights |
| Distance | Distance between airports (miles) |
| DistanceGroup | Distance Intervals, every 250 Miles, for Flight Segment |
| CarrierDelay | Carrier Delay, in Minutes |
| WeatherDelay | Weather Delay, in Minutes |
| NASDelay | National Air System Delay, in Minutes |
| SecurityDelay | Security Delay, in Minutes |
| LateAircraftDelay | Late Aircraft Delay, in Minutes |
| FirstDepTime | First Gate Departure Time at Origin Airport |

| Column name | Explanation of Column Name and Codes |
|----------------------|--|
| TotalAddGTime | Total Ground Time Away from Gate for Gate Return or Cancelled Flight |
| LongestAddGTime | Longest Time Away from Gate for Gate Return or Cancelled Flight |
| DivAirportLandings | Number of Diverted Airport Landings |
| DivReachedDest | Diverted Flight Reaching Scheduled Destination Indicator (1=Yes) |
| DivActualElapsedTime | Elapsed Time of Diverted Flight Reaching Scheduled Destination, in Minutes. The ActualElapsedTime column remains NULL for all diverted flights. |
| DivArrDelay | Difference in minutes between scheduled and actual arrival time for a diverted flight reaching scheduled destination. The ArrDelay column remains NULL for all diverted flights. |
| DivDistance | Distance between scheduled destination and final diverted airport (miles). Value will be 0 for diverted flight reaching scheduled destination. |
| Div1Airport | Diverted Airport Code1 |
| Div1AirportID | Airport ID of Diverted Airport 1. Airport ID is a Unique Key for an Airport |
| Div1AirportSeqID | Airport Sequence ID of Diverted Airport 1. Unique Key for Time Specific Information for an Airport |
| Div1WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code1 |
| Div1TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code1 |
| Div1LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code1 |
| Div1WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code1 |
| Div1TailNum | Aircraft Tail Number for Diverted Airport Code1 |
| Div2Airport | Diverted Airport Code2 |
| Div2AirportID | Airport ID of Diverted Airport 2. Airport ID is a Unique Key for an Airport |
| Div2AirportSeqID | Airport Sequence ID of Diverted Airport 2. Unique Key for Time Specific Information for an Airport |
| Div2WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code2 |
| Div2TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code2 |
| Div2LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code2 |
| Div2WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code2 |
| Div2TailNum | Aircraft Tail Number for Diverted Airport Code2 |
| Div3Airport | Diverted Airport Code3 |
| Div3AirportID | Airport ID of Diverted Airport 3. Airport ID is a Unique Key for an Airport |
| Div3AirportSeqID | Airport Sequence ID of Diverted Airport 3. Unique Key for Time Specific Information for an Airport |
| Div3WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code3 |
| Div3TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code3 |
| Div3LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code3 |
| Div3WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code3 |
| Div3TailNum | Aircraft Tail Number for Diverted Airport Code3 |
| Div4Airport | Diverted Airport Code4 |
| Div4AirportID | Airport ID of Diverted Airport 4. Airport ID is a Unique Key for an Airport |
| Div4AirportSeqID | Airport Sequence ID of Diverted Airport 4. Unique Key for Time Specific Information for an Airport |
| Div4WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code4 |
| Div4TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code4 |
| Div4LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code4 |
| Div4WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code4 |
| Div4TailNum | Aircraft Tail Number for Diverted Airport Code4 |
| Div5Airport | Diverted Airport Code5 |
| Div5AirportID | Airport ID of Diverted Airport 5. Airport ID is a Unique Key for an Airport |
| Div5AirportSeqID | Airport Sequence ID of Diverted Airport 5. Unique Key for Time Specific Information for an Airport |
| Div5WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code5 |

| Column name | Explanation of Column Name and Codes |
|------------------|--|
| Div5TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code5 |
| Div5LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code5 |
| Div5WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code5 |
| Div5TailNum | Aircraft Tail Number for Diverted Airport Code5 |

3.3 Explore Data

3.3.1 Data Exploration Report

3.3.1.1 Animal strike data

3.3.1.2 Flight data

TODO

3.4 Verify Data Quality

3.4.1 Data Quality Report

TODO

4 Data Preparation

4.1 Data Set

4.1.1 Data Set Description

TODO

4.2 Select Data

4.2.1 Rationale for Inclusion / Exclusion

TODO

4.3 Clean Data

4.3.1 Data Cleaning Report

TODO

4.4 Construct Data

4.4.1 Derived Attributes

TODO

4.4.2 Generated Records

TODO

4.5 Integrate Data

4.5.1 Merged Data

TODO

4.6 Format Data

4.6.1 Reformatted Data

TODO

5 Modeling

TODO

5.1 Select Modeling Technique for Model 1

5.1.1 Modeling Technique

TODO

5.1.2 Modeling Assumptions

TODO

5.2 Generate Test Design for Model 1

5.2.1 Test Design

TODO

5.3 Build Model for Model 1

5.3.1 Parameter Settings

TODO

5.3.2 Models

TODO

5.3.3 Model Description

TODO

5.4 Assess Model for Model 1

5.4.1 Model Assessment

TODO

5.4.2 Revised Parameter Settings

TODO

5.5 Select Modeling Technique for Model 2

5.5.1 Modeling Technique

TODO

5.5.2 Modeling Assumptions

TODO

5.6 Generate Test Design for Model 2

5.6.1 Test Design

TODO

5.7 Build Model for Model 2

5.7.1 Parameter Settings

TODO

5.7.2 Models

TODO

5.7.3 Model Description

TODO

5.8 Assess Model for Model 2

5.8.1 Model Assessment

TODO

5.8.2 Revised Parameter Settings

TODO

6 Evaluation

6.1 Evaluate Results

6.1.1 Assessment of Data Mining Result with Business Success Criteria

TODO

6.1.2 Approved Models

TODO

6.2 Review Process

6.2.1 Review of Process

TODO

6.3 Determine Next Steps

6.3.1 List of Possible Actions

TODO

6.3.2 Decision

TODO

7 Deployment

7.1 Plan Deployment

7.1.1 Deployment Plan

TODO

7.2 Plan Monitoring and Maintenance

7.2.1 Monitoring and Maintenance Plan

TODO

7.3 Produce Final Report

7.3.1 Final Report

TODO

7.3.2 Final Presentation

TODO

7.4 Review Project

7.4.1 Experience Documentation

TODO

8 Contributors

Student: Gábor Horváth

Mentor: Gergely Daróczi

9 Environment

The following language, tool and library versions have been used to create the project:

R Studio version 1.0.143

R version 3.4.0 (2017-04-21) 72570

Package versions:

- RODBC version 1.3.15
- knitr version 1.15.1
- data.table version 1.10.4
- dplyr version 0.5.0
- dtplyr version 0.0.2
- ReporteRs version 0.8.8
- ReporteRsjars version 0.0.2
- installr version 0.19.0
- stringr version 1.2.0
- ggplot2 version 2.2.1
- yaml version 2.1.14

Base package versions:

- stats version 3.4.0
- graphics version 3.4.0
- grDevices version 3.4.0
- utils version 3.4.0
- datasets version 3.4.0
- methods version 3.4.0
- base version 3.4.0

MiKTeX Package Manager 2.9.6200 (MiKTeX 2.9.6210 64-bit)

Copyright (C) 2005-2016 Christian Schenk

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Business Understanding | 2 |
| 2.1 | Determine Business Objectives | 2 |
| 2.1.1 | Business Objectives | 2 |
| 2.1.2 | Business Success Criteria | 2 |
| 2.2 | Assess Situation | 2 |
| 2.2.1 | Inventory of resources | 2 |
| 2.2.2 | Requirements, Assumptions, and Constraints | 2 |
| 2.2.3 | Risks and Contingencies | 2 |
| 2.2.4 | Terminology | 2 |
| 2.2.5 | Costs and Benefits | 3 |
| 2.3 | Determine Data Mining Goals | 3 |
| 2.3.1 | Data Mining Goals | 3 |
| 2.3.2 | Data Mining Success Criteria | 3 |
| 2.4 | Produce Project Plan | 3 |
| 2.4.1 | Project Plan | 3 |
| 2.4.2 | Initial Assessment of Tools and Techniques | 3 |
| 3 | Data Understanding | 5 |
| 3.1 | Collect Initial Data | 5 |
| 3.1.1 | Initial Data Collection Report | 5 |
| 3.2 | Describe Data | 5 |
| 3.2.1 | Data Description Report | 5 |
| 3.3 | Explore Data | 11 |
| 3.3.1 | Data Exploration Report | 11 |
| 3.4 | Verify Data Quality | 11 |
| 3.4.1 | Data Quality Report | 11 |
| 4 | Data Preparation | 12 |
| 4.1 | Data Set | 12 |
| 4.1.1 | Data Set Description | 12 |
| 4.2 | Select Data | 12 |
| 4.2.1 | Rationale for Inclusion / Exclusion | 12 |
| 4.3 | Clean Data | 12 |
| 4.3.1 | Data Cleaning Report | 12 |
| 4.4 | Construct Data | 12 |
| 4.4.1 | Derived Attributes | 12 |
| 4.4.2 | Generated Records | 12 |
| 4.5 | Integrate Data | 12 |
| 4.5.1 | Merged Data | 12 |
| 4.6 | Format Data | 12 |
| 4.6.1 | Reformatted Data | 12 |
| 5 | Modeling | 13 |
| 5.1 | Select Modeling Technique for Model 1 | 13 |
| 5.1.1 | Modeling Technique | 13 |
| 5.1.2 | Modeling Assumptions | 13 |
| 5.2 | Generate Test Design for Model 1 | 13 |
| 5.2.1 | Test Design | 13 |
| 5.3 | Build Model for Model 1 | 13 |
| 5.3.1 | Parameter Settings | 13 |
| 5.3.2 | Models | 13 |

| | | |
|-----------|---|-----------|
| 5.3.3 | Model Description | 13 |
| 5.4 | Assess Model for Model 1 | 13 |
| 5.4.1 | Model Assessment | 13 |
| 5.4.2 | Revised Parameter Settings | 13 |
| 5.5 | Select Modeling Technique for Model 2 | 14 |
| 5.5.1 | Modeling Technique | 14 |
| 5.5.2 | Modeling Assumptions | 14 |
| 5.6 | Generate Test Design for Model 2 | 14 |
| 5.6.1 | Test Design | 14 |
| 5.7 | Build Model for Model 2 | 14 |
| 5.7.1 | Parameter Settings | 14 |
| 5.7.2 | Models | 14 |
| 5.7.3 | Model Description | 14 |
| 5.8 | Assess Model for Model 2 | 14 |
| 5.8.1 | Model Assessment | 14 |
| 5.8.2 | Revised Parameter Settings | 14 |
| 6 | Evaluation | 15 |
| 6.1 | Evaluate Results | 15 |
| 6.1.1 | Assessment of Data Mining Result with Business Success Criteria | 15 |
| 6.1.2 | Approved Models | 15 |
| 6.2 | Review Process | 15 |
| 6.2.1 | Review of Process | 15 |
| 6.3 | Determine Next Steps | 15 |
| 6.3.1 | List of Possible Actions | 15 |
| 6.3.2 | Decision | 15 |
| 7 | Deployment | 16 |
| 7.1 | Plan Deployment | 16 |
| 7.1.1 | Deployment Plan | 16 |
| 7.2 | Plan Monitoring and Maintenance | 16 |
| 7.2.1 | Monitoring and Maintenance Plan | 16 |
| 7.3 | Produce Final Report | 16 |
| 7.3.1 | Final Report | 16 |
| 7.3.2 | Final Presentation | 16 |
| 7.4 | Review Project | 16 |
| 7.4.1 | Experience Documentation | 16 |
| 8 | Contributors | 17 |
| 9 | Environment | 18 |
| 10 | Appendix 1 - Final project requirements | 21 |
| 11 | Appendix 2 - Project Plan | 23 |
| 12 | Appendix 3 - Business Needs | 25 |
| 13 | Appendix 4 - Estimate of Resource Needs | 30 |
| 14 | Appendix 5 - Source code | 32 |
| | References | 50 |

10 Appendix 1 - Final project requirements

The following pages contain the Final project requirements received from the CEU Business School.

Final project

The goal of the final project is to expose the students in Business Analytics to a complete analytics workflow with a variety of tasks. They will use the full spectrum of skills acquired in the program, challenge themselves and learn something useful in the process and create value for the partner company. Throughout the project, students will interact with clients in the host company, analysts, IT engineers, and vendors of analytics solutions.

Examples

Insurance Ltd. sells many insurance products through a variety of channels. Customer data are stored in separate data silos for each market segment (e.g., life, home, car, travel), and there are often duplicates across sales channels (e.g., brokers do not check for existing customers but enter everyone as a new customer). In order to analyze customer behavior (e.g. churn) in all segments jointly, senior analysts need to merge all data by the same user. This requires entity resolution and unique user ID in all data silos. The student will study the various datasets, research entity resolution tools, conduct some tests with one or more prototype, and propose a solution to senior management.

Webstore Kft. is an online store of sporting goods. They want to evaluate the effectiveness of past social marketing campaigns. The management would like to know the average spending of new customers. Clickthrough rates are measured, but Webstore does not have information on conversion: if and what the newly acquired customers bought. Discussing with the person responsible for social campaigns, and the person running the website and maintaining the log, the student helps approximately identify new customers in the log and estimate their spending. She presents the results under alternative assumptions to the management. Together, they also propose a method for tracking conversion better.

Banking Ltd. is a financial company issuing credit cards. They have an existing model for predicting credit card non-payments which they want to improve. They have just launched a Hadoop project so they require a student with Hadoop expertise. Student meets with clients and analysts to understand current model and the need for improvement. Research current dataset and other data that can potentially be used to help predict default. Working as part of the analytics team, builds a prototype of a new machine learning model and tests its performance. Presents results to clients.

Resource needs

Each student has a **mentor** appointed by CEU and a **host** in their host company. The host company provides access to the necessary **space**, **people**, computer, software **tools** and **data**. The precise resource needs depend on the project and are negotiated in advance with the help of the mentor.

Benefits to host company

- Temporary staff with high technical skills and sensitive to the business environment; more dependable than entry-level interns.
- Consultations with CEU mentor.
- Access to latest technologies and trends.
- New perspective on a particular analytics problem or the analytics workflow.

Responsibilities

Student

- Select a host company and a project.
- Meet with mentor early on to discuss plans.
- Meet with mentor biweekly during the implementation of the project.
- Identify and understand business needs of host company clients.
- Select appropriate tools and provide best effort to address those needs.
- Complete deliverables by deadlines below.
- Maintain code of academic ethics, workplace rules of host company, and nondisclosure as agreed in project plan.
- Immediately raise concerns about project with mentor.

Mentor

- Help select a topic.
- Meet biweekly with the student to monitor progress and provide feedback.
- Verify project is feasible within the time frame.
- Discuss with host in case of concerns and problems.
- Verify successful project delivery at all stages.

Host

1. Propose analytics topics relevant to the host company.
2. Together with the mentor, identify the special needs in training, skills and tools.
3. Discuss with mentor and student the proposed project and agree on a plan.
4. Provide access for student to space, people, tools and data needed for successful completion of project.
5. Introduce student to other stakeholders at the company.

Deliverables

- Project plan. Describe the project and the resource needs in one page. Any special need in training, tools or any restrictions (e.g., non-disclosure agreement) should be specified here. Signed by student, host and mentor. Due [April 3, 2017](#).
- Business needs. Student documents business needs as gathered from clients. User stories, scope of the project. Due April 30.
- Estimate of resource needs. Students estimates the resource needs of the project. Who needs to be involved? What time do they need to devote to the project? Any new software or data needs to be purchased? Due April 30.
- Preliminary report. This contains the description of the business needs and the scope of the project, results of the analysis with exhibits, and recommendations for management. Due to host and mentor by June 30.

11 Appendix 2 - Project Plan

The following pages contain the Project Plan, which is the first deliverable described as the “Describe the project and the resource needs in one page. Any special need in training, tools or any restrictions (e.g., non-disclosure agreement) should be specified here.” in the final project requirements.

Project Plan of the Final Paper

for the CEU MSc in Business Analytics program

Gábor Horváth

2017



1 High level description

The goal of the project is to show - creating a risk evaluation of wildlife strikes of flights in the US - the techniques, methods, interpretations and understanding of the data analytic. The project is based on the Cross Industry Standard Process for Data Mining (CRISP-DM) process model, which is widely used worldwide for various scientific and business related data analytic projects. The use of the CRISP-DM process model will enable to for the project to cover all those areas (i.e. Business Understanding, Data understanding, Modelling, Evaluation, etc.), which are crucial of managing and delivering a successful data analytic project.

2 Resource needs

2.1 Training requirements

No additional organized / official training requirements are required above the trainings received during the courses in the program. There are tools and techniques used to fulfill the project which have not been described in the program at CEU, but there are several useful user manuals available on the webpages of the tool's creators, which would enable the use of these tools and resources for any student who have been part of the program.

2.2 Tools & resources used

Fulfilling the completion need for the project the following tools are planned to be used:

- Programming language:
 - R: <https://www.r-project.org/>
- IDE for the programming language:
 - RStudio: <https://www.rstudio.com/>
- Documentation is created using:
 - knitr: <https://yihui.name/knitr/>
 - MiKTeX: <https://miktex.org/>
 - ReporteRs: <https://cran.r-project.org/web/packages/ReporteRs/index.html>
- Data visualization:
 - ggplot2: <http://ggplot2.org/>
- Data manipulation:
 - access2csv: <https://github.com/AccelerationNet/access2csv>
 - dplyr: <https://cran.r-project.org/web/packages/dplyr/index.html>
- Project plan / task management:
 - Buckets: <https://www.buckets.co/>
- Source code repository:
 - GitHub: <https://github.com/>

Note: The list above do not contain the list of all the tools and packages used to create the project, but the full list will be provided in the source code.

2.3 Data sources

The project will use the following data provided by multiple US government agencies:

- Federal Aviation Administration: [Wildlife Strike Database](#)
- United States Department of Transportation: [Bureau of Transportation Statistics](#)

Note: In case data enrichment would be required for the successful risk modelling, additional data sources might be used as well. These possible additional data sources will be listed in the Final Paper.

2.4 Restrictions

Restrictions apply as per the restrictions set by the tools, data providers and owners of additional resources used. No additional restrictions have been identified and set regarding the use of the results of this project.

2.5 Contributors

Student: Gábor Horváth
Mentor: Gergely Daróczi

12 Appendix 3 - Business Needs

The following pages contain the Business Needs, which is the second deliverable described as the “Student documents business needs as gathered from clients. User stories, scope of the project.” in the final project requirements.

Business needs of the Final Paper

for the CEU MSc in Business Analytics program

Gábor Horváth

2017



2 Business understanding

2.1 Determine Business Objectives

2.1.1 Business Objectives

There are two main objectives what the project is aiming to complete.

1. Create a statistical analysis to identify those reasons (based on the data available), which are determining the risk of an animal strike for an airport.
2. Create a prediction model, which can be used to predict the risk of an animal strike for a given flight.

The result of the statistical analysis could be used in the completion of the model building and evaluation the recommended order of the completion is the order of the objectives stated above.

2.1.2 Business Success Criteria

- Identification of features determining the risk potential of an airport
- Working model for animal strike prediction

2.2 Assess Situation

2.2.1 Inventory of resources

- Flight Data
- Animal Strike Data
- R
- Buckets

2.2.2 Requirements, Assumptions, and Constraints

- Additional requirements:
 - No additional requirements identified on top of the requirements already stated in this document.
- Assumptions
 - No initial assumptions made.
- Constraints
 - No initial hard constraints identified.

2.2.3 Risks and Contingencies

- Risks
 - No initial risks identified
- Contingencies
 - No initial contingencies identified

2.2.4 Terminology

The project is using different terminologies from the different domains. The terms/definitions used will not be marked or explained in details, if based on the context the reader can easily identify the domain of the particular term. In case there are uncertainties about a term (and it's not explained in the paper), the following sources can be used for the definitions:

1 Introduction

The structure of the document follows the Cross Industry Standard Process for Data Mining (CRISP-DM) process model, which is a non-proprietary, documented, and freely available data mining model (Shearer 2000). Whenever the model sections can be matched to (and can fulfill) the requirements stated by CEU for the Final Paper I'm using the appropriate section identified by the CRIPS-DM model. Please keep in mind that the model supports the full end-to-end process of a data mining project, but the project does not require the use of all the model elements.

1

- Aviation:
 - Aviation Terms / Directory: <http://www.aviation-terms.com/index2.php>
 - Aviation Glossary: <http://www.aerofiles.com/glossary.html>
 - Aviation Glossaries: https://www.flightsimaviation.com/_glossaries.html?s=aviation_terms
- Data Mining
 - Data Mining Glossary: <http://www.theartling.com/glossary.htm>
 - Data Mining - Terminologies: https://www.tutorialspoint.com/data_mining/dm_terminologies.htm
 - Data Mining and Predictive Analytics Glossary: <http://www.kdnuggets.com/2015/06/data-mining-predictive-analytics-glossary.html>
- Data Science / Big Data
 - Data Science Glossary: <http://www.datascienceglossary.org/>
 - Analytics and Big Data Glossary: <http://data-informed.com/glossary-of-big-data-terms/>
 - Data Science Glossary: <http://www.kdnuggets.com/2015/09/data-science-glossary.html>

2.2.5 Costs and Benefits

This is a one-man project, no significant cost is expected. Main benefit is to put to and almost end-to-end scenario the topics covered during the courses and discovering bits and bolts of the techniques for creating the project.

2.3 Determine Data Mining Goals

2.3.1 Data Mining Goals

- Understand, Analyse, Clean and Merge the source data correctly
- Create the required attributes
- Generate the required records (if applicable)

2.3.2 Data Mining Success Criteria

- Identification of featured determining the risk potential of an airport
- Working model for animal strike prediction

2.4 Produce Project Plan

2.4.1 Project Plan

The project is managed in an agile way, where all the tasks, requirements, issues, solutions, and ideas are kept in a project at [buckets](#).

2.4.2 Initial Assessment of Tools and Techniques

- Programming language:
 - R: <https://www.r-project.org/>
- IDE for the programming language:
 - RStudio: <https://www.rstudio.com/>
- Documentation is created using:
 - knitr: <https://yihui.name/knitr/>
 - MiKTeX: <https://miktex.org/>
 - ReporteRs: <https://cran.r-project.org/web/packages/ReporteRs/index.html>
- Data visualization:

- ggplot2: <http://ggplot2.org/>
- Data manipulation:
 - access2csv: <https://github.com/AccelerationNet/access2csv>
 - dplyr: <https://cran.r-project.org/web/packages/dplyr/index.html>
- Project plan / task management:
 - Buckets: <https://www.buckets.co/>
- Source code repository:
 - GitHub: <https://github.com/>

Note: The list above do not contain the list of all the tools and packages used to create the project, but the full list will be provided in the source code.

3 Data Understanding

3.1 Collect Initial Data

3.1.1 Initial Data Collection Report

This report will be part of the following documents:

- Preliminary Report
- Final Paper

3.2 Describe Data

3.2.1 Data Description Report

The two main data sources have the following column explanations, which is attached to the downloaded files as well, by the data provider agencies.

3.2.1.1 Animal strike data

| Column name | Explanation of Column Name and Codes |
|-------------------|---|
| INDEX_NR | Individual record number |
| OPID | Airline operator code |
| OPERATOR | A three letter International Civil Aviation Organization code for aircraft operators. (BUS = business, PVT = private aircraft other than business, GOV = government aircraft, MIL - military aircraft.) |
| ATYPE | Aircraft |
| AMA | International Civil Aviation Organization code for Aircraft Make |
| AMO | International Civil Aviation Organization code for Aircraft Model |
| EMA | Engine Make Code (see Engine Codes tab below) |
| EMO | Engine Model Code (see Engine Codes tab below) |
| AC_CLASS | Type of aircraft (see Aircraft Type tab below) |
| AC_MASS | 1 = 2,250 kg or less; 2 = 2,251-5700 kg; 3 = 5,701-27,000 kg; 4 = 27,001-272,000 kg; 5 = above 272,000 kg |
| NUM_ENGS | Number of engines |
| TYPE_ENG | Type of power A = reciprocating engine (piston); B = Turbojet; C = Turboprop; D = Turbofan; E = None (glider); F = Turboshift (helicopter); Y = Other |
| ENG_1_POS | Where engine # 1 is mounted on aircraft (see Engine Position tab below) |
| ENG_2_POS | Where engine # 2 is mounted on aircraft (see Engine Position tab below) |
| ENG_3_POS | Where engine # 3 is mounted on aircraft (see Engine Position tab below) |
| ENG_4_POS | Where engine # 4 is mounted on aircraft (see Engine Position tab below) |
| REG | Aircraft registration |
| FLT | Flight number |
| REMAINS_COLLECTED | Indicates if bird or wildlife remains were found and collected |
| REMAINS_SENT | Indicates if remains were sent to the Smithsonian Institution for identification |
| INCIDENT_DATE | Date strike occurred |
| INCIDENT_MONTH | Month strike occurred |
| INCIDENT_YEAR | Year strike occurred |
| TIME_OF_DAY | Light conditions |
| TIME | Hour and minute in local time |

4

5

| Column name | Explanation of Column Name and Codes |
|----------------------|---|
| AIRPORT_ID | International Civil Aviation Organization airport identifier for location of strike whether it was on or off airport |
| AIRPORT | Name of airport |
| STATE | State |
| FAAREGION | FAA Region where airport is located |
| ENROUTE | If strike did not occur on approach, climb, landing roll, taxi or take-off, aircraft was enroute. This shows location. |
| RUNWAY | Runway |
| LOCATION | Various information about aircraft location if enroute or airport where strike evidence was found. Some locations show the two airports for the flight departure and arrival if pilot was unaware of the strike. |
| HEIGHT | Feet Above Ground Level |
| SPEED | Knots (indicated air speed) |
| DISTANCE | Miles from airport |
| PHASE_OF_FLT | Phase of flight during which strike occurred |
| DAMAGE | |
| Blank | Unknown |
| M = minor | When the aircraft can be rendered airworthy by simple repairs or replacements and an extensive inspection is not necessary. |
| M? = uncertain level | The aircraft was damaged, but details as to the extent of the damage are lacking. |
| S = substantial | When the aircraft incurs damage or structural failure which adversely affects the structure strength, performance or flight characteristics of the aircraft and which would normally require major repair or replacement of the affected component. |
| D = Destroyed | When the damage sustained makes it inadvisable to restore the aircraft to an airworthy condition. |
| STR_RAD | Struck radome |
| DAM_RAD | Damaged radome |
| STR_WINDSHLD | Struck windshield |
| DAM_WINDSHLD | Damaged windshield |
| STR_NOSE | Struck nose |
| DAM_NOSE | Damaged nose |
| STR_ENG1 | Struck Engine 1 |
| DAM_ENG1 | Damaged Engine 1 |
| STR_ENG2 | Struck Engine 2 |
| DAM_ENG2 | Damaged Engine 2 |
| STR_ENG3 | Struck Engine 3 |
| DAM_ENG3 | Damaged Engine 3 |
| STR_ENG4 | Struck Engine 4 |
| DAM_ENG4 | Damaged Engine 4 |
| INGESTED | Engine ingested the bird/ animal |
| STR_PROP | Struck Propeller |
| DAM_PROP | Damaged Propeller |
| STR_WING_ROT | Struck Wing or Rotor |
| DAM_WING_ROT | Damaged Wing or Rotor |
| STR_FUSE | Struck Fuselage |
| DAM_FUSE | Damaged Fuselage |
| STR_LG | Struck Landing Gear |
| DAM_LG | Damaged Landing Gear |
| STR_TAIL | Struck Tail |
| DAM_TAIL | Damaged Tail |

6

| Column name | Explanation of Column Name and Codes |
|-----------------------|--|
| STR_LIGHTS | Struck Lights |
| DAM_LIGHTS | Damaged Lights |
| STR_OTHER | Struck Other than parts shown above |
| DAM_OTHER | Damaged Other than parts shown above |
| OTHER_SPECIFY | What part was struck other than those listed above |
| EFFECT | Effect on flight |
| EFFECT_OTHER | Effect on flight other than those listed on the form |
| SKY | Type of cloud cover, if any |
| PRECIP | Precipitation |
| SPECIES_ID | International Civil Aviation Organization code for type of bird or other wildlife |
| SPECIES | Common name for bird or other wildlife |
| BIRDS_SEEN | Number of birds/wildlife seen by pilot |
| BIRDS_STRUCK | Number of birds/wildlife struck |
| SIZE | Size of bird as reported by pilot is a relative scale. Entry should reflect the perceived size as opposed to a scientifically determined value. If more than one species was struck, larger bird is entered. |
| WARNED | Pilot warned of birds/wildlife |
| COMMENTS | As entered by database manager. Can include name of aircraft owner, types of reports received, updates, etc. |
| REMARKS | Most of remarks are from the form but some are data entry notes and are usually in parentheses. |
| AOS | Time aircraft was out of service in hours. If unknown, it is blank. |
| COST_REPAIRS | Estimated cost of repairs of replacement in dollars (USD) |
| COST_OTHER | Estimated other costs, other than those in previous field in dollars (USD). May include loss of revenue, hotel expenses due to flight cancellation, costs of fuel dumped, etc. |
| COST_REPAIRS_INFL_ADJ | Costs adjusted for inflation |
| COST_OTHER_INFL_ADJ | Other cost adjusted for inflation |
| REPORTED_NAME | Name(s) of person(s) filing report |
| REPORTED_TITLE | Title(s) of person(s) filing report |
| REPORTED_DATE | Date report was written |
| SOURCE | Type of report. Note: for multiple types of reports this will be indicated as Multiple. See "Comments" field for details |
| PERSON | Only one selection allowed. For multiple reports, see field "Reported Title" |
| NR_INJURIES | Number of people injured |
| NR_FATALITIES | Number of human fatalities |
| LUPDATE | Last time record was updated |
| TRANSFER | Unused field at this time |
| INDICATED_DAMAGE | Indicates whether or not aircraft was damaged |

3.2.1.2 Flight data

| Column name | Explanation of Column Name and Codes |
|-------------|--------------------------------------|
| Year | Year |
| Quarter | Quarter (1-4) |
| Month | Month |
| DayofMonth | Day of Month |
| DayOfWeek | Day of Week |
| FlightDate | Flight Date (yyyymmdd) |

7

| Column name | Explanation of Column Name and Codes |
|----------------------|--|
| UniqueCarrier | Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years. |
| AirlineID | An identification number assigned by US DOT to identify a unique airline (carrier). A unique airline (carrier) is defined as one holding and reporting under the same DOT certificate regardless of its Code, Name, or holding company/corporation. |
| Carrier | Code assigned by IATA and commonly used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the Unique Carrier Code. |
| TailNum | Tail Number |
| FlightNum | Flight Number |
| OriginAirportID | Origin Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused. |
| OriginAirportSeqID | Origin Airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time. |
| OriginCityMarketID | Origin Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market. |
| Origin | Origin Airport |
| OriginCityName | Origin Airport, City Name |
| OriginState | Origin Airport, State Code |
| OriginStateFips | Origin Airport, State Fips |
| OriginStateName | Origin Airport, State Name |
| OriginWac | Origin Airport, World Area Code |
| DestAirportID | Destination Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused. |
| DestAirportSeqID | Destination Airport, Airport Sequence ID. An identification number assigned by US DOT to identify a unique airport at a given point of time. Airport attributes, such as airport name or coordinates, may change over time. |
| DestCityMarketID | Destination Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market. |
| Dest | Destination Airport |
| DestCityName | Destination Airport, City Name |
| DestState | Destination Airport, State Code |
| DestStateFips | Destination Airport, State Fips |
| DestStateName | Destination Airport, State Name |
| DestWac | Destination Airport, World Area Code |
| CRSDepTime | CRS Departure Time (local time: hhmm) |
| DepTime | Actual Departure Time (local time: hhmm) |
| DepDelay | Difference in minutes between scheduled and actual departure time. Early departures show negative numbers. |
| DepDelayMinutes | Difference in minutes between scheduled and actual departure time. Early departures set to 0. |
| DepDel15 | Departure Delay Indicator, 15 Minutes or More (1=Yes) |
| DepartureDelayGroups | Departure Delay intervals, every (15 minutes from <=15 to >180) |
| DepTimeBlk | CRS Departure Time Block, Hourly Intervals |
| TaxiOut | Taxi Out Time, in Minutes |
| WheelsOff | Wheels Off Time (local time: hhmm) |

8

| Column name | Explanation of Column Name and Codes |
|------------------|--|
| Div2LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code2 |
| Div2WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code2 |
| Div2TailNum | Aircraft Tail Number for Diverted Airport Code2 |
| Div3Airport | Diverted Airport Code3 |
| Div3AirportID | Airport ID of Diverted Airport 3. Airport ID is a Unique Key for an Airport |
| Div3AirportSeqID | Airport Sequence ID of Diverted Airport 3. Unique Key for Time Specific Information for an Airport |
| Div3WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code3 |
| Div3TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code3 |
| Div3LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code3 |
| Div3WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code3 |
| Div3TailNum | Aircraft Tail Number for Diverted Airport Code3 |
| Div4Airport | Diverted Airport Code4 |
| Div4AirportID | Airport ID of Diverted Airport 4. Airport ID is a Unique Key for an Airport |
| Div4AirportSeqID | Airport Sequence ID of Diverted Airport 4. Unique Key for Time Specific Information for an Airport |
| Div4WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code4 |
| Div4TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code4 |
| Div4LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code4 |
| Div4WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code4 |
| Div4TailNum | Aircraft Tail Number for Diverted Airport Code4 |
| Div5Airport | Diverted Airport Code5 |
| Div5AirportID | Airport ID of Diverted Airport 5. Airport ID is a Unique Key for an Airport |
| Div5AirportSeqID | Airport Sequence ID of Diverted Airport 5. Unique Key for Time Specific Information for an Airport |
| Div5WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code5 |
| Div5TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code5 |
| Div5LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code5 |
| Div5WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code5 |
| Div5TailNum | Aircraft Tail Number for Diverted Airport Code5 |

3.3 Explore Data

3.3.1 Data Exploration Report

This report will be part of the following documents:

- Preliminary Report
- Final Paper

3.4 Verify Data Quality

3.4.1 Data Quality Report

This report will be part of the following documents:

- Preliminary Report
- Final Paper

| Column name | Explanation of Column Name and Codes |
|----------------------|--|
| WheelsOn | Wheels On Time (local time: hhmm) |
| TaxiIn | Taxi In Time, in Minutes |
| CRSArrTime | CRS Arrival Time (local time: hhmm) |
| ArrTime | Actual Arrival Time (local time: hhmm) |
| ArrDelay | Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers. |
| ArrDelayMinutes | Difference in minutes between scheduled and actual arrival time. Early arrivals set to 0. |
| ArrDel15 | Arrival Delay Indicator, 15 Minutes or More (1=Yes) |
| ArrivalDelayGroups | Arrival Delay intervals, every (15-minutes from <=15 to >180) |
| ArrTimeBlk | CRS Arrival Time Block, Hourly Intervals |
| Cancelled | Cancelled Flight Indicator (1=Yes) |
| CancellationCode | Specifies The Reason For Cancellation |
| Diverted | Diverted Flight Indicator (1=Yes) |
| CRSElapsedTime | CRS Elapsed Time of Flight, in Minutes |
| ActualElapsedTime | Elapsed Time of Flight, in Minutes |
| AirTime | Flight Time, in Minutes |
| Flights | Number of Flights |
| Distance | Distance between airports (miles) |
| DistanceGroup | Distance Intervals, every 250 Miles, for Flight Segment |
| CarrierDelay | Carrier Delay, in Minutes |
| WeatherDelay | Weather Delay, in Minutes |
| NASDelay | National Air System Delay, in Minutes |
| SecurityDelay | Security Delay, in Minutes |
| LateAircraftDelay | Late Aircraft Delay, in Minutes |
| FirstDepTime | First Gate Departure Time at Origin Airport |
| TotalAddGTime | Total Ground Time Away from Gate for Gate Return or Cancelled Flight |
| LongestAddGTime | Longest Time Away from Gate for Gate Return or Cancelled Flight |
| DivAirportLandings | Number of Diverted Airport Landings |
| DivReachedDest | Diverted Flight Reaching Scheduled Destination Indicator (1=Yes) |
| DivActualElapsedTime | Elapsed Time of Diverted Flight Reaching Scheduled Destination, in Minutes. The ActualElapsedTime column remains NULL for all diverted flights. |
| DivArrDelay | Difference in minutes between scheduled and actual arrival time for a diverted flight reaching scheduled destination. The ArrDelay column remains NULL for all diverted flights. |
| DivDistance | Distance between scheduled destination and final diverted airport (miles). Value will be 0 for diverted flight reaching scheduled destination. |
| Div1Airport | Diverted Airport Code1 |
| Div1AirportID | Airport ID of Diverted Airport 1. Airport ID is a Unique Key for an Airport |
| Div1AirportSeqID | Airport Sequence ID of Diverted Airport 1. Unique Key for Time Specific Information for an Airport |
| Div1WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code1 |
| Div1TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code1 |
| Div1LongestGTime | Longest Ground Time Away from Gate at Diverted Airport Code1 |
| Div1WheelsOff | Wheels Off Time (local time: hhmm) at Diverted Airport Code1 |
| Div1TailNum | Aircraft Tail Number for Diverted Airport Code1 |
| Div2Airport | Diverted Airport Code2 |
| Div2AirportID | Airport ID of Diverted Airport 2. Airport ID is a Unique Key for an Airport |
| Div2AirportSeqID | Airport Sequence ID of Diverted Airport 2. Unique Key for Time Specific Information for an Airport |
| Div2WheelsOn | Wheels On Time (local time: hhmm) at Diverted Airport Code2 |
| Div2TotalGTime | Total Ground Time Away from Gate at Diverted Airport Code2 |

9

4 Contributors

Student: Gábor Horváth
Mentor: Gergely Daróczi

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Business understanding | 2 |
| 2.1 | Determine Business Objectives | 2 |
| 2.1.1 | Business Objectives | 2 |
| 2.1.2 | Business Success Criteria | 2 |
| 2.2 | Assess Situation | 2 |
| 2.2.1 | Inventory of resources | 2 |
| 2.2.2 | Requirements, Assumptions, and Constraints | 2 |
| 2.2.3 | Risks and Contingencies | 2 |
| 2.2.4 | Terminology | 2 |
| 2.2.5 | Costs and Benefits | 3 |
| 2.3 | Determine Data Mining Goals | 3 |
| 2.3.1 | Data Mining Goals | 3 |
| 2.3.2 | Data Mining Success Criteria | 3 |
| 2.4 | Produce Project Plan | 3 |
| 2.4.1 | Project Plan | 3 |
| 2.4.2 | Initial Assessment of Tools and Techniques | 3 |
| 3 | Data Understanding | 5 |
| 3.1 | Collect Initial Data | 5 |
| 3.1.1 | Initial Data Collection Report | 5 |
| 3.2 | Describe Data | 5 |
| 3.2.1 | Data Description Report | 5 |
| 3.3 | Explore Data | 10 |
| 3.3.1 | Data Exploration Report | 10 |
| 3.4 | Verify Data Quality | 10 |
| 3.4.1 | Data Quality Report | 10 |
| 4 | Contributors | 11 |
| | References | 13 |

References

Shearer, Colin. 2000. "The Crisp-Dm Model - the New Blueprint for Data Mining." Journal of Data Warehousing 5 (4): 13-22.

13 Appendix 4 - Estimate of Resource Needs

The following pages contain the Estimate of Resource Needs, which is the third deliverable described as the “Students estimates the resource needs of the project. Who needs to be involved? What time do they need to devote to the project? Any new software or data needs to purchased?” in the final project requirements.

Estimate of resource needs of the Final Paper

for the CEU MSc in Business Analytics program

Gábor Horváth

2017



- Federal Aviation Administration: [Wildlife Strike Database](#)
- United States Department of Transportation: [Bureau of Transportation Statistics](#)

Note: In case data enrichment would be required for the successful risk modelling, additional data sources might be used as well. These possible additional data sources will be listed in the Final Paper.

3 Contributors

Student: Gábor Horváth
Mentor: Gergely Daróczi

1 Human resource needs

1.1 Stakeholders & people to involve

As this final paper is a pet project, there is no actual business management behind the requirements, therefore no business stakeholders are identified and involved. The completion of the project requires feedback and guidance from the mentor (Gergely Daróczi), but no other person (or role) needs to be involved.

1.2 Dedication for the project

There are no additional time dedication requirements identified above the requirements stated by CEU in the Final Project description document.

1.3 Training requirements

As stated earlier no additional organized / official training requirements are required above the trainings received during the courses in the program. There are tools and techniques used to fulfill the project, which have not been described in the program at CEU. There are several useful user manuals available on the webpages of the creators of the tools, which would enable the use of these tools and resources for any student who have been part of the program.

2 Software and data resource needs

2.1 Tools & resources used

As stated earlier, fulfilling the completion need for the project the following tools are planned to be used:

- Programming language:
 - R: <https://www.r-project.org/>
- IDE for the programming language:
 - RStudio: <https://www.rstudio.com/>
- Documentation is created using:
 - knitr: <https://yihui.name/knitr/>
 - MiKTeX: <https://miktex.org/>
 - ReporteRs: <https://cran.r-project.org/web/packages/ReporteRs/index.html>
- Data visualization:
 - ggplot2: <http://ggplot2.org/>
- Data manipulation:
 - access2csv: <https://github.com/AccelerationNet/access2csv>
 - dplyr: <https://cran.r-project.org/web/packages/dplyr/index.html>
- Project plan / task management:
 - Buckets: <https://www.buckets.co/>
- Source code repository:
 - GitHub: <https://github.com/>

Note: The list above do not contain the list of all the tools and packages used to create the project, but the full list will be provided in the source code.

2.2 Data sources

As stated earlier, the project will use the following data provided by multiple US government agencies:

14 Appendix 5 - Source code

The following pages contain the source code of the project.

```
# #'
# #' \code{wildLifeStrikeDataSet} based on the configuration
# #' items checks if the wildlife strike data set file has been:
# #' - downloaded
# #' - uncompressed
# #' - included tables extracted
# #' if not, then execute these tasks.
# #'
# #' @examples
# #' wildLifeStrikeDataSet()
# #'
# wildLifeStrikeDataSet <- function() {
#   #setting the download parameters
#   URL <- getWDData()
#   destfile <- paste(getDataDir(), "wildlife.zip", sep = "/")
#
#   method="auto"
#
#   #if the file exists then do not download again
#   if (file.exists(destfile) != TRUE)
#   {
#     download.file(URL, destfile, method)
#   } else
#   {
#     message("File exists no download required.")
#   }
#
#   destdir <- getDataDir()
#
#   #unzip the file
#   unzip(destfile, exdir = destdir)
#
#   csvfile <- paste(destdir,
#                     "/STRIKE_REPORTS (1990-1999).csv",
#                     sep="")
#
#   if (file.exists(csvfile) != TRUE)
#   {
#     setwd(getDataDir())
#     system(paste("java -jar ",
#                   getDataDir(),
#                   "/access2csv.jar ",
#                   getDataDir(),
#                   "/wildlife.accdb",
#                   sep = ""))
#     setwd(getMainDir())
#   } else
#   {
#     message("File exists no extract required.")
#   }
# }
```

```

#
# }
# #'
# #' \code{onTimeFlightPerformanceDataSet} based on the configuration
# #' items checks if the commercial flight data set files have been:
# #' - downloaded
# #' - uncompressed
# #' if not, then execute these tasks.
# #'
# #' @examples
# #' onTimeFlightPerformanceDataSet()
# #'
# onTimeFlightPerformanceDataSet <- function() {
#
#   method="auto"
#   dataDir <- getDataDir()
#   startYear <- getStartYear()
#   endYear <- getEndYear()
#   startMonth <- getStartMonth()
#   endMonth <- getEndMonth()
#
#   for (i in startYear:endYear){
#     for (j in startMonth:endMonth){
#
#       variableName <- paste("On_Time_On_Time_Performance_",
#                             i,
#                             "_",
#                             j,
#                             sep = "")
#
#       sourceFile <- paste(variableName, ".zip", sep = "")
#       URL <- paste(getFData(), sourceFile, sep = "")
#       destinationFile <- paste(dataDir, "/", sourceFile, sep = "")
#
#       #if the file exists then do not download again
#       if (file.exists(destinationFile) != TRUE)
#       {
#         message("Downloading ", sourceFile)
#         download.file(URL, destinationFile, method)
#         Sys.sleep(0.1)
#       } else
#       {
#         message(sourceFile,
#                 " file exists, no download is required.")
#       }
#
#       zippedFileName <- sourceFile
#       zippedFile <- destinationFile
#       unzippedFileName <- paste(variableName,
#                                 ".csv",
#                                 sep = "")
#       unzippedFile <- paste(dataDir, "/", unzippedFileName, sep = "")
#

```

```

#         #if the file exists then do not unzip it again
#         if (file.exists(unzippedFile) != TRUE)
#         {
#             message("Unzipping ", zippedFileName)
#             unzip(zippedFile,
#                 overwrite = FALSE,
#                 exdir = dataDir) #No overwrite
#             #Clear warnings
#             assign("last.warning", NULL, envir = baseenv())
#         } else
#         {
#             message(unzippedFileName,
#                 " file exists, no unzip is required.")
#         }
#
#     } #end of "for (j in startMonth:endMonth)"
# } #end of "for (i in startYear:endYear)"
#
# }

# #'
# #' \code{wildLifeStrikeDataSetSplitByYear} splits the strike data
# #' into RDS files by year, so that the data files would be aligned
# #' across the different data sets
# #'
# #' @examples
# #' wildLifeStrikeDataSetSplitByYear()
# #'
# wildLifeStrikeDataSetSplitByYear <- function() {
#
#     dataDir <- getDataDir()
#     startYear <- getStartYear()
#     endYear <- getEndYear()
#
#     for (i in startYear:endYear){
#         RDSFileName <- paste(i,
#                               "_Animal_Strikes.rds",
#                               sep = "")
#
#         RDSFile <- paste(dataDir,
#                          "/",
#                          RDSFileName,
#                          sep = "")
#
#         if (file.exists(RDSFile) != TRUE){
#
#             if (exists("sr_1990_1999") != TRUE){
#
#                 message("Reading sr_1990_1999")
#
#                 variableName <- "sr_1990_1999"
#
#                 assign(variableName,
#                     data.table(

```

```
#         read.csv(  
#         paste(  
#             dataDir,  
#             "/STRIKE_REPORTS (1990-1999).csv",  
#             sep=""),  
#         header = FALSE)),  
#         envir = .GlobalEnv)  
#  
# names(sr_1990_1999) <- c("INDEX_NR",  
#                           "OPID",  
#                           "OPERATOR",  
#                           "ATYPE",  
#                           "AMA",  
#                           "AMO",  
#                           "EMA",  
#                           "EMO",  
#                           "AC_CLASS",  
#                           "AC_MASS",  
#                           "NUM_ENGS",  
#                           "TYPE_ENG",  
#                           "ENG_1_POS",  
#                           "ENG_2_POS",  
#                           "ENG_3_POS",  
#                           "ENG_4_POS",  
#                           "REG",  
#                           "FLT",  
#                           "REMAINS_COLLECTED",  
#                           "REMAINS_SENT",  
#                           "INCIDENT_DATE",  
#                           "INCIDENT_MONTH",  
#                           "INCIDENT_YEAR",  
#                           "TIME_OF_DAY",  
#                           "TIME",  
#                           "AIRPORT_ID",  
#                           "AIRPORT",  
#                           "STATE",  
#                           "FAAREGION",  
#                           "ENROUTE",  
#                           "RUNWAY",  
#                           "LOCATION",  
#                           "HEIGHT",  
#                           "SPEED",  
#                           "DISTANCE",  
#                           "PHASE_OF_FLT",  
#                           "DAMAGE",  
#                           "STR_RAD",  
#                           "DAM_RAD",  
#                           "STR_WINDSHLD",  
#                           "DAM_WINDSHLD",  
#                           "STR_NOSE",  
#                           "DAM_NOSE",  
#                           "STR_ENG1",  
#                           "DAM_ENG1",  
#                           "STR_ENG2",
```

```
# "DAM_ENG2",
# "STR_ENG3",
# "DAM_ENG3",
# "STR_ENG4",
# "DAM_ENG4",
# "INGESTED",
# "STR_PROP",
# "DAM_PROP",
# "STR_WING_ROT",
# "DAM_WING_ROT",
# "STR_FUSE",
# "DAM_FUSE",
# "STR_LG",
# "DAM_LG",
# "STR_TAIL",
# "DAM_TAIL",
# "STR_LGHTS",
# "DAM_LGHTS",
# "STR_OTHER",
# "DAM_OTHER",
# "OTHER_SPECIFY",
# "EFFECT",
# "EFFECT_OTHER",
# "SKY",
# "PRECIP",
# "SPECIES_ID",
# "SPECIES",
# "BIRDS_SEEN",
# "BIRDS_STRUCK",
# "SIZE",
# "WARNED",
# "COMMENTS",
# "REMARKS",
# "AOS",
# "COST_REPAIRS",
# "COST_OTHER",
# "COST_REPAIRS_INFL_ADJ",
# "COST_OTHER_INFL_ADJ",
# "REPORTED_NAME",
# "REPORTED_TITLE",
# "REPORTED_DATE",
# "SOURCE",
# "PERSON",
# "NR_INJURIES",
# "NR_FATALITIES",
# "LUPDATE",
# "TRANSFER",
# "INDICATED_DAMAGE")
#
# }
#
# if (exists("sr_2000_2009") != TRUE) {
#
#     message("Reading sr_2000_2009")
#
# }
```

```
#      variableName <- "sr_2000_2009"
#
#      assign(variableName,
#             data.table(
#               read.csv(
#                 paste(
#                   dataDir,
#                   "/STRIKE_REPORTS (2000-2009).csv",
#                   sep=""),
#                 header = FALSE)),
#             envir = .GlobalEnv)
#
#      names(sr_2000_2009) <- c("INDEX_NR",
#                                "OPID",
#                                "OPERATOR",
#                                "ATYPE",
#                                "AMA",
#                                "AMO",
#                                "EMA",
#                                "EMO",
#                                "AC_CLASS",
#                                "AC_MASS",
#                                "NUM_ENGS",
#                                "TYPE_ENG",
#                                "ENG_1_POS",
#                                "ENG_2_POS",
#                                "ENG_3_POS",
#                                "ENG_4_POS",
#                                "REG",
#                                "FLT",
#                                "REMAINS_COLLECTED",
#                                "REMAINS_SENT",
#                                "INCIDENT_DATE",
#                                "INCIDENT_MONTH",
#                                "INCIDENT_YEAR",
#                                "TIME_OF_DAY",
#                                "TIME",
#                                "AIRPORT_ID",
#                                "AIRPORT",
#                                "STATE",
#                                "FAAREGION",
#                                "ENROUTE",
#                                "RUNWAY",
#                                "LOCATION",
#                                "HEIGHT",
#                                "SPEED",
#                                "DISTANCE",
#                                "PHASE_OF_FLT",
#                                "DAMAGE",
#                                "STR_RAD",
#                                "DAM_RAD",
#                                "STR_WINDSHLD",
#                                "DAM_WINDSHLD",
#                                "STR_NOSE",
```

```

# "DAM_NOSE",
# "STR_ENG1",
# "DAM_ENG1",
# "STR_ENG2",
# "DAM_ENG2",
# "STR_ENG3",
# "DAM_ENG3",
# "STR_ENG4",
# "DAM_ENG4",
# "INGESTED",
# "STR_PROP",
# "DAM_PROP",
# "STR_WING_ROT",
# "DAM_WING_ROT",
# "STR_FUSE",
# "DAM_FUSE",
# "STR_LG",
# "DAM_LG",
# "STR_TAIL",
# "DAM_TAIL",
# "STR_LGHTS",
# "DAM_LGHTS",
# "STR_OTHER",
# "DAM_OTHER",
# "OTHER_SPECIFY",
# "EFFECT",
# "EFFECT_OTHER",
# "SKY",
# "PRECIP",
# "SPECIES_ID",
# "SPECIES",
# "BIRDS_SEEN",
# "BIRDS_STRUCK",
# "SIZE",
# "WARNED",
# "COMMENTS",
# "REMARKS",
# "AOS",
# "COST_REPAIRS",
# "COST_OTHER",
# "COST_REPAIRS_INFL_ADJ",
# "COST_OTHER_INFL_ADJ",
# "REPORTED_NAME",
# "REPORTED_TITLE",
# "REPORTED_DATE",
# "SOURCE",
# "PERSON",
# "NR_INJURIES",
# "NR_FATALITIES",
# "LUPDATE",
# "TRANSFER",
# "INDICATED_DAMAGE")
#
# }
#

```

```
# if (exists("sr_2010_Current") != TRUE){
#
#   message("Reading sr_2010_Current")
#
#   variableName <- "sr_2010_Current"
#
#   assign(variableName,
#           data.table(
#             read.csv(
#               paste(
#                 dataDir,
#                 "/STRIKE_REPORTS (2010-Current).csv",
#                 sep=""),
#               header = FALSE)),
#           envir = .GlobalEnv)
#
#   names(sr_2010_Current) <- c("INDEX_NR",
#                                "OPID",
#                                "OPERATOR",
#                                "ATYPE",
#                                "AMA",
#                                "AMO",
#                                "EMA",
#                                "EMO",
#                                "AC_CLASS",
#                                "AC_MASS",
#                                "NUM_ENGS",
#                                "TYPE_ENG",
#                                "ENG_1_POS",
#                                "ENG_2_POS",
#                                "ENG_3_POS",
#                                "ENG_4_POS",
#                                "REG",
#                                "FLT",
#                                "REMAINS_COLLECTED",
#                                "REMAINS_SENT",
#                                "INCIDENT_DATE",
#                                "INCIDENT_MONTH",
#                                "INCIDENT_YEAR",
#                                "TIME_OF_DAY",
#                                "TIME",
#                                "AIRPORT_ID",
#                                "AIRPORT",
#                                "STATE",
#                                "FAAREGION",
#                                "ENROUTE",
#                                "RUNWAY",
#                                "LOCATION",
#                                "HEIGHT",
#                                "SPEED",
#                                "DISTANCE",
#                                "PHASE_OF_FLT",
#                                "DAMAGE",
#                                "STR_RAD",
```

```
# "DAM_RAD",
# "STR_WINDSHLD",
# "DAM_WINDSHLD",
# "STR_NOSE",
# "DAM_NOSE",
# "STR_ENG1",
# "DAM_ENG1",
# "STR_ENG2",
# "DAM_ENG2",
# "STR_ENG3",
# "DAM_ENG3",
# "STR_ENG4",
# "DAM_ENG4",
# "INGESTED",
# "STR_PROP",
# "DAM_PROP",
# "STR_WING_ROT",
# "DAM_WING_ROT",
# "STR_FUSE",
# "DAM_FUSE",
# "STR_LG",
# "DAM_LG",
# "STR_TAIL",
# "DAM_TAIL",
# "STR_LGHTS",
# "DAM_LGHTS",
# "STR_OTHER",
# "DAM_OTHER",
# "OTHER_SPECIFY",
# "EFFECT",
# "EFFECT_OTHER",
# "SKY",
# "PRECIP",
# "SPECIES_ID",
# "SPECIES",
# "BIRDS_SEEN",
# "BIRDS_STRUCK",
# "SIZE",
# "WARNED",
# "COMMENTS",
# "REMARKS",
# "AOS",
# "COST_REPAIRS",
# "COST_OTHER",
# "COST_REPAIRS_INFL_ADJ",
# "COST_OTHER_INFL_ADJ",
# "REPORTED_NAME",
# "REPORTED_TITLE",
# "REPORTED_DATE",
# "SOURCE",
# "PERSON",
# "NR_INJURIES",
# "NR_FATALITIES",
# "LUPDATE",
```

```

#                                     "TRANSFER",
#                                     "INDICATED_DAMAGE")
#
#     }
#
#     #STRIKE_REPORTS_BASH --> contains only military data, not required
#
#
#     if (i >= 1990 && i <= 1999) {
#         dataOfWholeYear <- sr_1990_1999[INCIDENT_YEAR == i]
#     }
#     else if (i >= 2000 && i <= 2009) {
#         dataOfWholeYear <- sr_2000_2009[INCIDENT_YEAR == i]
#     }
#     else if (i >= 2010 && i <= 2019) {
#         dataOfWholeYear <- sr_2010_Current[INCIDENT_YEAR == i]
#     }
#
#     saveRDS(dataOfWholeYear, file = RDSFile)
#     message(RDSFileName, " created.")
#
#     #free up memory
#     rm(dataOfWholeYear)
#     gc()
#
#     }
#     else {
#         message(RDSFileName,
#                 " exists, no further action is required.")
#     }
#
# } #end of "for (i in startYear:endYear)"
#
# }
#
# #'
# #' \code{onTimeFlightPerformanceDataSetMergeByYear} merges the flight data
# #' into RDS files by year, so that working with the data would not consume
# #' all the memory of the running environment
# #'
# #' @examples
# #' onTimeFlightPerformanceDataSetMergeByYear()
# #'
# onTimeFlightPerformanceDataSetMergeByYear <- function() {
#     dataDir <- getDataDir()
#     startYear <- getStartYear()
#     endYear <- getEndYear()
#     startMonth <- getStartMonth()
#     endMonth <- getEndMonth()
#
#     for (i in startYear:endYear){
#
#         RDSFileName <- paste(i,
#                               "_On_Time_On_Time_Performance.rds",

```

```

#           sep = "")
#
# RDSFile <- paste(dataDir,
#                 "/",
#                 RDSFileName,
#                 sep = "")
#
# #Create the RDS files only if they do not exist yet
# if (file.exists(RDSFile) != TRUE){
#
#   for (j in startMonth:endMonth){
#
#     variableName <- paste("On_Time_On_Time_Performance_",
#                           i,
#                           "_",
#                           j,
#                           sep = "")
#
#     unzippedFileName <- paste(variableName,
#                               ".csv",
#                               sep = "")
#
#     unzippedFile <- paste(dataDir,
#                           "/",
#                           unzippedFileName,
#                           sep = "")
#
#     assign(variableName,
#            data.table(read.csv(unzippedFile,
#                               header = TRUE)))
#
#     if (j == startMonth){
#       dataOfWholeYear <- get(variableName)
#       rm(list = ls(pattern = "On_Time_On_Time_Performance*"))
#       gc()
#     }
#     else {
#       dataOfWholeYear <- rbindlist(list(dataOfWholeYear,
#                                         get(variableName)))
#       rm(list = ls(pattern = "On_Time_On_Time_Performance*"))
#       gc()
#     }
#
#   } #end of "for (j in startMonth:endMonth)"
#
#   saveRDS(dataOfWholeYear, file = RDSFile)
#   message(RDSFileName," created.")
#
#   #free up memory
#   rm(dataOfWholeYear)
#   gc()
#
# }
# else {

```

```

#       message(RDSFileName,
#               " exists, no further action is required.")
#   }
#
# } #end of "for (i in startYear:endYear)"
#
# }

# #'
# #' \code{loadLibraries} checks if the required libraries are
# #' - installed and
# #' - loaded
# #' if not, the it installs (if required) and loads them.
# #'
# #' @examples
# #' loadLibraries()
# #'
# loadLibraries <- function() {
#   if (!require(installr)) {install.packages("installr"); require(installr)}
#   if (!require(RODBC)) {install.packages("RODBC"); require(RODBC)}
#   if (!require(knitr)) {install.packages("knitr"); require(knitr)}
#   if (!require(data.table)) {install.packages("data.table"); require(data.table)}
#   if (!require(dplyr)) {install.packages("dplyr"); require(dplyr)}
#   if (!require(dtplyr)) {install.packages("dtplyr"); require(dtplyr)}
#   if (!require(ggplot2)) {install.packages("ggplot2"); require(ggplot2)}
#   if (!require(ReporteRs)) {install.packages("ReporteRs"); require(ReporteRs)}
#   if (!require(yaml)) {install.packages("yaml"); require(yaml)}
#
#   #update R
#   updateR(TRUE)
#
#   #update MiKTeX packages
#   system("mpm --update --quiet")
#
#   #require(grid)
#   #require(lattice)
#   #require(ggplot2movies)
#   #require(latticeExtra)
# }
#
#
# #'
# #' \code{versionDetails} provides details about the running environment
# #'
# #' @return text with the versions of R, RStudio, and used packages
# #'
# #' @examples
# #' versionDetails()
# #'
# versionDetails <- function() {
#
#   cat(paste(
#     "R Studio version 1.0.143\n\n",
#     version$version.string, " ", version$`svn rev`, "\n\n",

```

```

#   "Package versions:\n",
#   "- RODBC version ", packageVersion("RODBC"), "\n",
#   "- knitr version ", packageVersion("knitr"), "\n",
#   "- data.table version ", packageVersion("data.table"), "\n",
#   "- dplyr version ", packageVersion("dplyr"), "\n",
#   "- dtplyr version ", packageVersion("dtplyr"), "\n",
#   "- ReporteRs version ", packageVersion("ReporteRs"), "\n",
#   "- ReporteRsjars version ", packageVersion("ReporteRsjars"), "\n",
#   "- installr version ", packageVersion("installr"), "\n",
#   "- stringr version ", packageVersion("stringr"), "\n",
#   "- ggplot2 version ", packageVersion("ggplot2"), "\n",
#   "- yaml version ", packageVersion("yaml"), "\n\n",
#   "Base package versions:\n",
#   "- stats version ", packageVersion("stats"), "\n",
#   "- graphics version ", packageVersion("graphics"), "\n",
#   "- grDevices version ", packageVersion("grDevices"), "\n",
#   "- utils version ", packageVersion("utils"), "\n",
#   "- datasets version ", packageVersion("datasets"), "\n",
#   "- methods version ", packageVersion("methods"), "\n",
#   "- base version ", packageVersion("base"), sep="")
# }
#
#
# #'
# #' \code{versionDetailsMiKTeX} provides details about the running
# #' environment
# #'
# #' @return text with the versions of MiKTeX
# #'
# #' @examples
# #' versionDetailsMiKTeX()
# #'
# versionDetailsMiKTeX <- function() {
#   cat(system("mpm --version", intern = TRUE), sep = '\n')
# }
#
#
# #'
# #' \code{versionDetailsMiKTeXPackages} provides details about the
# #' running environment
# #'
# #' @return text with the versions of the installed MiKTeX packages
# #'
# #' @examples
# #' versionDetailsMiKTeXPackages()
# #'
# versionDetailsMiKTeXPackages <- function() {
#   cat(system("mpm --list", intern = TRUE), sep = '\n')
# }
#
#
# #'
# #' \code{readConfigFile} reads the YAML config file into a global
# #' environment variable

```

```
# #'
# #' @param a boolean
# #' The YAML config file is in the working directory or in the parent
# #' directory (i.e. one directory above)
# #'
# #' @examples
# #' readConfigFile(TRUE)
# #'
# readConfigFile <- function(a) {
#   vName <- "config"
#   if (a == TRUE){
#     assign(vName, yaml.load_file("91-Config.yaml"), envir = .GlobalEnv)
#   }
#   else {
#     assign(vName, yaml.load_file("../91-Config.yaml"), envir = .GlobalEnv)
#   }
# }
#
#
# #'
# #' \code{getMainDir} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the maindir configuration item
# #'
# #' @examples
# #' getMainDir()
# #'
# getMainDir <- function() {
#   return(config$directories$maindir)
# }
#
#
# #'
# #' \code{getBackupDir} provides the value of the specific
# #' configuration item and creates the directory if it does
# #' not exist
# #'
# #' @return the value of the backupdir configuration item
# #'
# #' @examples
# #' getBackupDir()
# #'
# getBackupDir <- function() {
#   backupdir <- config$directories$backupdir
#   subdir <- Sys.Date()
#   returnvalue <- file.path(backupdir, subdir)
#
#   if (!file.exists(returnvalue)){
#     dir.create(returnvalue)
#     dir.create(file.path(returnvalue, "Documents"))
#   }
#   return(returnvalue)
# }
```

```
#
#
# #'
# #' \code{getDocDir} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the documents configuration item
# #'
# #' @examples
# #' getDocDir()
# #'
# getDocDir <- function() {
#   return(config$directories$documents)
# }
#
#
# #'
# #' \code{getDocInputDir} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the documentinput configuration item
# #'
# #' @examples
# #' getDocInputDir()
# #'
# getDocInputDir <- function() {
#   return(config$directories$documentinput)
# }
#
#
# #'
# #' \code{getDocOutputDir} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the documentoutput configuration item
# #'
# #' @examples
# #' getDocOutputDir()
# #'
# getDocOutputDir <- function() {
#   return(config$directories$documentoutput)
# }
#
#
# #'
# #' \code{getDataDir} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the datasets configuration item
# #'
# #' @examples
# #' getDataDir()
# #'
# getDataDir <- function() {
```

```
#   return(config$directories$datasets)
# }
#
#
# #'
# #' \code{getStartYear} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the startyear configuration item
# #'
# #' @examples
# #' getStartYear()
# #'
# getStartYear <- function() {
#   return(config$years$startyear)
# }
#
#
# #'
# #' \code{getEndYear} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the endyear configuration item
# #'
# #' @examples
# #' getEndYear()
# #'
# getEndYear <- function() {
#   return(config$years$endyear)
# }
#
#
# #'
# #' \code{getStartMonth} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the startmonth configuration item
# #'
# #' @examples
# #' getStartMonth()
# #'
# getStartMonth <- function() {
#   return(config$months$startmonth)
# }
#
#
# #'
# #' \code{getEndMonth} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the endmonth configuration item
# #'
# #' @examples
# #' getEndMonth()
```

```
# #'
# getEndMonth <- function() {
#   return(config$months$endmonth)
# }
#
#
# #'
# #' \code{backupFiles} makes a copy of the most important
# #' files to a safe location set by the YAML configuration
# #' file
# #'
# #' @examples
# #' backupFiles()
# #'
# backupFiles <- function() {
#   #Main directory files
#   filesMain <- list.files(getMainDir(), full.names = TRUE)
#   file.copy(filesMain, getBackupDir(), overwrite = TRUE)
#   #Documents folder
#   filesDocuments <- list.files(getDocDir(), full.names = TRUE)
#   file.copy(filesDocuments,
#             file.path(getBackupDir(),
#                       "Documents"),
#             overwrite = TRUE)
# }
#
#
# #'
# #' \code{getWDData} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the wildlife configuration item
# #'
# #' @examples
# #' getWDData()
# #'
# getWDData <- function() {
#   return(config$sources$wildlife)
# }
#
#
# #'
# #' \code{getFData} provides the value of the specific
# #' configuration item
# #'
# #' @return the value of the flightdata configuration item
# #'
# #' @examples
# #' getFData()
# #'
# getFData <- function() {
#   return(config$sources$flightdata)
# }
#
```

```
#
# #'
# #' \code{removeDataSetVariables} removes the data set variables
# #' from the memory and calls the garbage collection to free up
# #' memory - currently disabled
# #'
# #' @examples
# #' removeDataSetVariables()
# #'
# removeDataSetVariables <- function() {
#   # rm(list = ls(pattern = "On_Time_On_Time_Performance*",
#   #               envir = .GlobalEnv),
#   #   #   envir = .GlobalEnv)
#   # rm(list = ls(pattern = "sr_*",
#   #               envir = .GlobalEnv),
#   #   #   envir = .GlobalEnv)
#   # gc()
# }
#
#
# #'
# #' \code{loadSourceCodeFunctions} makes the functions created
# #' in different R files available for further use and process
# #' management
# #'
# #' @examples
# #' loadSourceCodeFunctions()
# #'
# loadSourceCodeFunctions <- function() {
#   source("01-WildLiveStrikeDataSetDataPreparation.R")
#   source("02-OnTimeFlightPerformanceDataSetDataPreparation.R")
#   source("03-WildLifeStrikeDataSetSplitByYear.R")
#   source("04-OnTimeFlightPerformanceDataSetMergeByYear.R")
# }
```

References

Shearer, Colin. 2000. "The Crisp-Dm Model - the New Blueprint for Data Mining." *Journal of Data Warehousing* 5 (4): 13–22.