# Project: Flight Delay and Cancellation Predictor

(Arpeet Kale, Saurabh Mehta and Jaime Montoya)

## 1. Introduction

**1.1. Problem description:** The uncertainty experienced by travelers about the probability of experiencing flight delays or cancellations produces an undesirable condition in which travelers do not know what to expect regarding the punctuality of their flights, representing a difficult challenge for the aviation systems (Xiong & Hansen, 2013, p. 64).

**1.2. Motivation:** Being stuck in an airport is a frustrating experience for travelers. In the past, it was possible to argue that lack of information and limited technology were the main constraints to be able to discover and analyze patterns in flight delays and cancellations. However, nowadays the information is available and data mining technologies have matured to facilitate the analysis of data that can be used to predict flight delays and cancellations.

**1.3. Report organization:** This report has been organized in seven main sections: Introduction, Data exploration, Methodology, Logic of the Problem, Conclusions, Appendix, and References. Within each of these sections, the development of this report has been organized systematically to allow readers to follow a sequential analysis of the project from its introduction and motivations, to the challenges, findings, and conclusions resulting from the research, development, and implementation of the project.

**2. Data exploration:** Some projects have to overcome the problem of lack of information. The Flight Delay and Cancellation project posed the opposite challenge: nearly 120 million records in total, which is an excessive amount of information. For this reason, exploring and understanding the data was one of the first priorities in the development of the project. It was crucial to study and understand the main characteristics of the data obtained from http://stat-computing.org/dataexpo/2009/the-data.html ("Airline on-time performance", 2009), such as the number of records available per year, total number of attributes and their corresponding data types, missing values and outliers. The importance of data exploration for this project is paramount because it provides the required information to choose appropriate preprocessing and data analysis techniques (Kumar, Steinbach & Tan, 2014, p. 97).

## 3. Methodology

**3.1. Data preprocessing:** Originally, the data was available as individual files corresponding to each year from 1987 to 2008, totalling nearly 120 million records. Depending on the research questions that were being answered, different preprocessing techniques were used, including sampling, feature subset selection, discretization, binarization, and dimensionality reduction.

## 3.2. Mining the data

### 3.2.1. Technique number one to build model: Naïve Bayes

**3.2.1.1. Summary:** The conditional independency between the attributes was the assumption that made this data mining technique ideal for this study. For example, knowing the departure time of a flight does not reveal the carrier because the attributes are conditionally independent.

**3.2.1.2. Parametrization:** The target attribute expected by the Naïve Bayes algorithm has to be nominal. After completing the preprocessing step of converting the target attributes ArrDelay and Cancelled from numeric to nominal, the default values of the parameters were used on Weka with a 10 fold cross-validation and 66% percentage split.

**3.2.1.3. Conclusions:** Initially, the Naïve Bayes algorithm was disabled on Weka when the ArrDelay and Cancelled attributes had a numeric type. After converting these attributes from numeric to nominal, it was possible to use Naïve Bayes for analyzing the data. This was because Naïve Bayes is a classification and not a regression algorithm. The performance of this algorithm to predict cancellations was outstanding. However, the accuracy was not better than the results obtained using J48 and SVM when the target attribute was the arrival delay. Naïve Bayes took the lowest time to build the model compared to the other two algorithms.

**3.2.1.4. Model output:** Please refer to Tables 6.1.1 and 6.1.2 in the Appendix to see the accuracy and time taken to build model, respectively.

### 3.2.2. Technique number two to build model : J48

**3.2.2.1. Summary:** ArrDelay and Cancellation are the dependent variables that have to be predicted. All the other attributes are independent variables used in the internal nodes of the decision tree with the purpose of finding the final value or classification that will predict if a flight will be cancelled or delayed. One of the main advantages of this data mining technique was the fact that the results were easy to understand, interpret and visualize because decision trees provide a visual representation that facilitates the analysis of the results provided by the classification model.

**3.2.2.2. Parametrization:** The parameter "unpruned" was set to False because it was convenient to perform pruning as an strategy to remove nodes that do not provide additional information. Another benefit of this configuration was reducing the size of the learning tree while maintaining the predictive accuracy. minNumObj was set to 2, which means that at least two instances must exist in the training data before creating a new leaf in the decision tree. This represents the minimum number of instances per leaf. The confidenceFactr was set to 0.25 because 25% works reasonably well in most cases (Lazakidou, 2010, p. 233)

**3.2.2.3. Conclusions:** The time taken to build the model was efficient because it always took less than one second for all of the test options using training and testing data. The accuracy was excellent for the Cancelled attribute, even though the performance was significantly lower (around 18%) when the target attribute was arrival delay.

**3.2.2.4. Model output:** Please refer to Tables 6.1.1 and 6.1.2 in the Appendix to see the accuracy and time taken to build model, respectively.

**3.2.3. Technique number three to build model :** Support Vector Machine

**3.2.3.1. Summary:** Considering that 21 attributes were used in the analysis, the curse of dimensionality was a potential problem. This type of high-dimensional data made SVM an excellent algorithm for this project. In addition, SVM is a memory-efficient algorithm, making it ideal for a project of this magnitude where the original dataset before preprocessing contained 29 attributes and nearly 120 million records.

**3.2.3.2. Parametrization:** The default values epsilon = 1.0E-12 and toleranceParameter = 0.001 were not changed, as explicitly stated in the Weka documentation for the SMO classifier.

**3.2.3.3. Conclusions:** This algorithm was extremely slow on training data regardless the test options used, remarkably for the arrival delay target attribute. Nonetheless, it was fast on testing data. In terms of accuracy, SVM performed great for the Cancelled target attribute, and acceptable (around 80%) for arrival delays.

**3.2.3.4. Model output:** Please refer to Tables 6.1.1 and 6.1.2 in the Appendix to see the accuracy and time taken to build model, respectively.

## 3.3. Models performance

**3.3.1. Performance measures:** A crucial performance measure used in this project was the accuracy or correctly classified instances observed from the classifier output after applying the algorithms Naïve Bayes, J48 and Support Vector Machine. This information was analyzed for the training and testing data sets for the purpose of evaluating the performance of each algorithm. Additionally, the time taken to build the model was another important consideration especially because a large amount of data was available for this project, which was the reason for sampling as one of the pre-processing steps. Accordingly, the total number of instances provided to run each algorithm (10,401 instances) was also considered as one of the measures. The confusion matrix was used in the process of obtaining the accuracy as part of the performance measures applied in this analysis.

**3.3.2. Meaning of performance measures:** The number of correctly classified instances revealed by the confusion matrix is the accuracy, expressed as the percentage of instances that were classified correctly. The time taken to build the model is automatically calculated by Weka, expressed in seconds. The number of instances evaluated was another important performance measure because some algorithms crashed due to an excessive number of instances that Weka could not handle. For that reason, the number of instances had to be reduced. The confusion matrix allows to visualize the performance of the algorithms in a simple table, before using more sophisticated visualization techniques.

**3.3.3. Motivation for choosing performance measures:** Even though there are countless strategies to measure the performance of an algorithm, the decision was to focus on the crucial measures for classification algorithms, which include the accuracy or number of correctly classified instances, the time taken to build the model, and the number of instances that the algorithm had to evaluate. This information provides an initial comparison and criteria to evaluate the performance of the algorithms in order to determine which one is the best for the analysis of the prediction of flight delays and cancellations.

**3.3.4. How performance measures were estimated:** The approach was to use all of the test options available on Weka (training set, supplied test set, cross-validation and percentage split) and then compare the performance produced by each of these test options. The results did not reveal any significant difference or pattern that could distinguish a specific test option as better or worse than the others.

**3.3.5. Performance of different models:** SVM was extremely slow compared to Naïve Bayes and J48. Overall, SVM produced the best results on the arrival delay target attribute. The three algorithms performed well on the Cancelled target attribute. The accuracy and time taken to build the model are shown in Tables 6.1.1 and 6.1.2.

## 4. Logic of the problem

**4.1. Experiences applying the Logic of the Problem tool:** This tool was fundamental resource to give meaning and direction to the project, from its the stage it was conceived to its implementation, feedback, and future work recommendations. Starting with the statement of the problem was important because as in any research endeavor, a significant problem must be found whose solution is worth the time and effort spent on it. During the development of the project, the Eight Elements of Thought served as guidelines in the reasoning to discover the solution to the problem.

**4.2. Pros and cons using this analytical thinking procedure:** Three relevant pros of using this tool are: 1) Systematic, strategic, and meaningful approach to discover an efficient solution to almost any problem. 2) Test the validity of the arguments against biased opinions and wrong assumptions. 3) Applying the Eight Elements serves as a filter to refine the quality of the conclusions or solution to the problem. Two cons experienced while using this tool were: 1) The Concepts Element seemed to focus on definitions. However, it also included theories, axioms, laws, principles and models. This can represent a vast amount of material that would be convenient to analyze as individual elements rather than trying to make everything fit under the same Concepts Element or category. 2) The iterative, incremental and evolutionary approach that usually characterizes problems and their potential solutions, creates the necessity of a changelog to analyze the progress of the work performed and use the feedback to fix problems or challenges encountered. The tool is missing this element that would benefit the analysis of a problem to discover optimal solutions.

**4.3. Recommendations to add, change or delete steps:** The Concepts Element should be divided into more elements or categories in order to analyze the definitions separately from the theories, axioms and models because each of these components include crucial and sufficient information that would be convenient to analyze independently instead of trying to merge everything into a single category. An element called "Retrospective Analysis" should be included as a changelog that can be used to study the evolution of the solution to the problem with  the purpose of learning from past errors or analyzing feedback to optimize and discover intelligent solutions to the problem.

## 5. Conclusions

### 5.1. Comparison of results

The results suggest that flight cancellations follow patterns that produce a significantly higher accuracy than those observed while analyzing flight delays. In other words, it is easier to predict flight cancellations than it is to predict flight delays. Including additional attributes that reveal why delays occur, as it happens in the case of cancellations with the attribute CancellationCode that reveals the reason for cancellation (A=carrier,B=weather,C=NAS,D=security), can help to improve accuracy when analyzing flight delays.

**5.1.1. Patterns confirmed:** Flights cancellations and delays tend to increase in the summer and around holidays.

**5.1.2. Patterns discovered:** Flying in the early hours of the morning or late at night is ideal to avoid flight delays and cancellations. Also, the largest and most popular carriers usually experience more cancellations.

**5.1.3. Models that work better:** SVM took the longest time to build the model but it produced the best accuracy for arrival delays, which was the most difficult target attribute to classify correctly or predict.

**5.2. Summary of the problem and conclusions drawn from the built model:** The challenge was to find patterns that help to predict flight delays and cancellations. The models revealed that it is easier to predict flight delays than it is to determine whether or not a flight will be cancelled. More attributes such as reasons for delay should be included in the dataset to understand why flight delays happen.

**5.3. Next steps if the project is continued:** Focusing on flight delays would be beneficial considering the significantly lower accuracy obtained for the arrival delay target attribute. This should include not only information about whether or not a flight will be delayed, but also details that explain how many minutes late a flight is likely to arrive and the reason for the delay.

## 6. Appendix

### 6.1. Model outputs

| Algorithm | Accuracy on training data | | | Accuracy on testing data |
|---|---|---|---|---|
| | Training set | 10-Fold Cross-Validation | Percentage-Split (66%) | |
| Naïve Bayes | Cancelled: 98.4905% | Cancelled: 98.3559% | Cancelled: 98.5011% | Cancelled: 100% |
| | Delayed: 75.6177% | Delayed: 74.2332% | Delayed: 72.9072% | Delayed: 78.3654% |
| J48 | Cancelled: 98.3175% | Cancelled: 98.3175% | Cancelled: 98.4729% | Cancelled: 96.6346% |
| | Delayed: 82.1940% | Delayed: 80.7230% | Delayed: 78.9876% | Delayed: 75.9615% |
| Support Vector Machine | Cancelled: 98.3463% | Cancelled: 98.2982% | Cancelled: 98.4729% | Cancelled: 100% |
| | Delayed: 82.2805% | Delayed: 81.2614% | Delayed: 80.3450% | Delayed: 88.4615% |

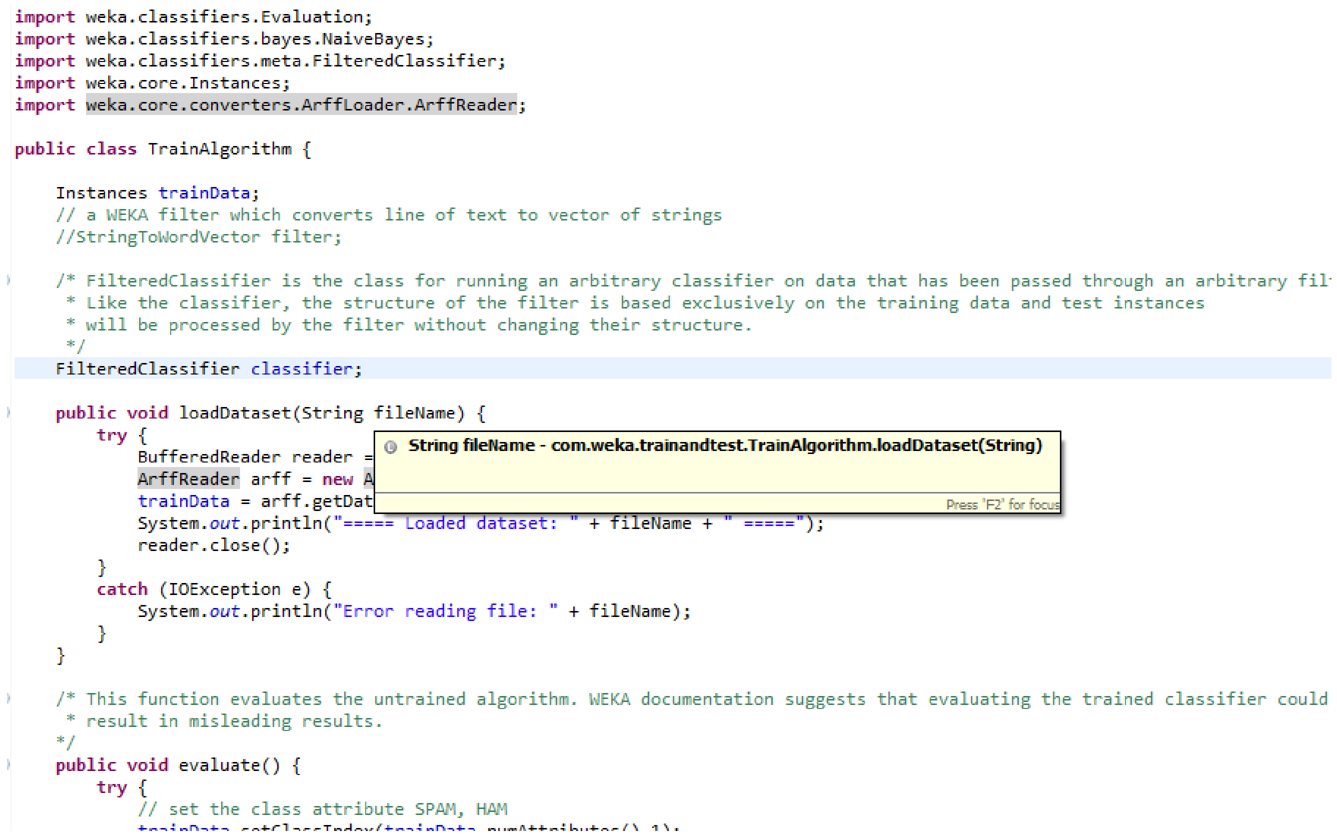
Table 6.1.1. Accuracy results using the data mining techniques Naïve Bayes, J48 and Support Vector Machine.

| Algorithm | Time taken to build model on training data (expressed in seconds) | | | Time taken to build model on testing data (expressed in seconds) |
|---|---|---|---|---|
| | Training set | 10-Fold Cross-Validation | Percentage-Split (66%) | |
| Naïve Bayes | Cancelled: 0.03 | Cancelled: 0.02 | Cancelled: 0.03 | Cancelled: 0.0 |
| | Delayed: 0.06 | Delayed: 0.02 | Delayed: 0.02 | Delayed: 0.03 |
| J48 | Cancelled: 0.1 | Cancelled: 0.23 | Cancelled: 0.21 | Cancelled: 0.0 |
| | Delayed: 0.28 | Delayed: 0.55 | Delayed: 0.37 | Delayed: 0.04 |
| Support Vector Machine | Cancelled: 13.33 | Cancelled: 13.71 | Cancelled: 13.41 | Cancelled: 0.02 |
| | Delayed: 290.21 | Delayed: 540.98 | Delayed: 546.14 | Delayed: 0.17 |

Table 6.1.2. Time taken to build model using data mining techniques Naïve Bayes, J48 and Support Vector Machine.

## 6.2. Additional work for extra credit

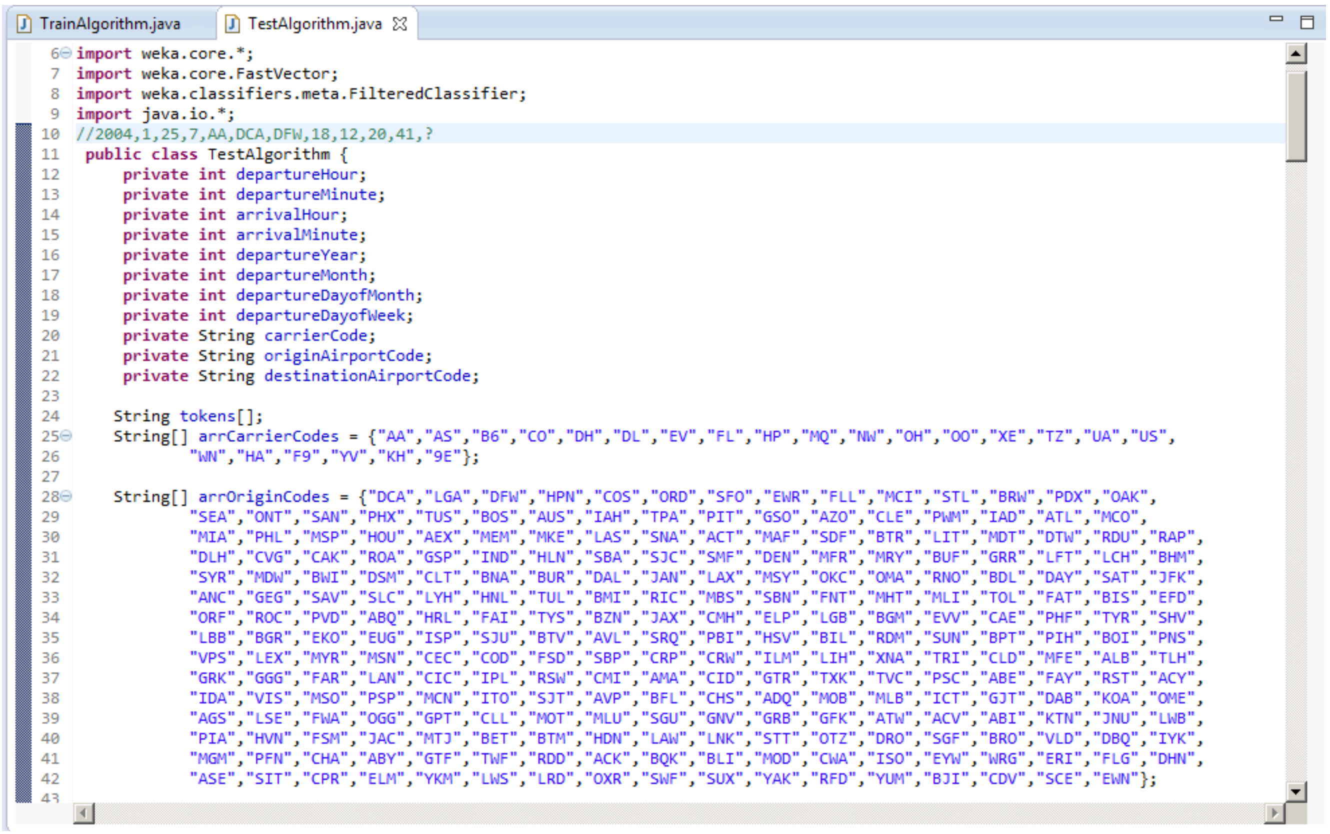
- **Java implementation of algorithms using Weka libraries:** These open source libraries were used by importing the classifier packages and using them on Java to train the Naïve Bayes algorithm on the dataset, which was a representative sample of five years of data.



```
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayes;
import weka.classifiers.meta.FilteredClassifier;
import weka.core.Instances;
import weka.core.converters.ArffLoader.ArffReader;

public class TrainAlgorithm {

    Instances trainData;
    // a WEKA filter which converts line of text to vector of strings
    //StringToWordVector filter;

    /* FilteredClassifier is the class for running an arbitrary classifier on data that has been passed through an arbitrary fil
     * Like the classifier, the structure of the filter is based exclusively on the training data and test instances
     * will be processed by the filter without changing their structure.
     */
    FilteredClassifier classifier;

    public void loadDataset(String fileName) {
        try {
            BufferedReader reader =
            ArffReader arff = new A
            trainData = arff.getDat
            System.out.println("===== Loaded dataset: " + fileName + " =====");
            reader.close();
        }
        catch (IOException e) {
            System.out.println("Error reading file: " + fileName);
        }
    }

    /* This function evaluates the untrained algorithm. WEKA documentation suggests that evaluating the trained classifier could
     * result in misleading results.
    */
    public void evaluate() {
        try {
            // set the class attribute SPAM, HAM
```

String fileName - com.weka.trainandtest.TrainAlgorithm.loadDataset(String)
Press 'F2' for focus

Figure 6.2.1. Java implementation of Weka open source classifier for the training algorithm.



```java
import weka.core.*;
import weka.core.FastVector;
import weka.classifiers.meta.FilteredClassifier;
import java.io.*;
//2004,1,25,7,AA,DCA,DFW,18,12,20,41,?
 public class TestAlgorithm {
     private int departureHour;
     private int departureMinute;
     private int arrivalHour;
     private int arrivalMinute;
     private int departureYear;
     private int departureMonth;
     private int departureDayofMonth;
     private int departureDayofWeek;
     private String carrierCode;
     private String originAirportCode;
     private String destinationAirportCode;

    String tokens[];
    String[] arrCarrierCodes = {"AA","AS","B6","CO","DH","DL","EV","FL","HP","MQ","NW","OH","OO","XE","TZ","UA","US",
            "WN","HA","F9","YV","KH","9E"};

    String[] arrOriginCodes = {"DCA","LGA","DFW","HPN","COS","ORD","SFO","EWR","FLL","MCI","STL","BRW","PDX","OAK",
            "SEA","ONT","SAN","PHX","TUS","BOS","AUS","IAH","TPA","PIT","GSO","AZO","CLE","PWM","IAD","ATL","MCO",
            "MIA","PHL","MSP","HOU","AEX","MEM","MKE","LAS","SNA","ACT","MAF","SDF","BTR","LIT","MDT","DTW","RDU","RAP",
            "DLH","CVG","CAK","ROA","GSP","IND","HLN","SBA","SJC","SMF","DEN","MFR","MRY","BUF","GRR","LFT","LCH","BHM",
            "SYR","MDW","BWI","DSM","CLT","BNA","BUR","DAL","JAN","LAX","MSY","OKC","OMA","RNO","BDL","DAY","SAT","JFK",
            "ANC","GEG","SAV","SLC","LYH","HNL","TUL","BMI","RIC","MBS","SBN","FNT","MHT","MLI","TOL","FAT","BIS","EFD",
            "ORF","ROC","PVD","ABQ","HRL","FAI","TYS","BZN","JAX","CMH","ELP","LGB","BGM","EVV","CAE","PHF","TYR","SHV",
            "LBB","BGR","EKO","EUG","ISP","SJU","BTV","AVL","SRQ","PBI","HSV","BIL","RDM","SUN","BPT","PIH","BOI","PNS",
            "VPS","LEX","MYR","MSN","CEC","COD","FSD","SBP","CRP","CRW","ILM","LIH","XNA","TRI","CLD","MFE","ALB","TLH",
            "GRK","GGG","FAR","LAN","CIC","IPL","RSW","CMI","AMA","CID","GTR","TXK","TVC","PSC","ABE","FAY","RST","ACY",
            "IDA","VIS","MSO","PSP","MCN","ITO","SJT","AVP","BFL","CHS","ADQ","MOB","MLB","ICT","GJT","DAB","KOA","OME",
            "AGS","LSE","FWA","OGG","GPT","CLL","MOT","MLU","SGU","GNV","GRB","GFK","ATW","ACV","ABI","KTN","JNU","LWB",
            "PIA","HVN","FSM","JAC","MTJ","BET","BTM","HDN","LAW","LNK","STT","OTZ","DRO","SGF","BRO","VLD","DBQ","IYK",
            "MGM","PFN","CHA","ABY","GTF","TWF","RDD","ACK","BQK","BLI","MOD","CWA","ISO","EYW","WRG","ERI","FLG","DHN",
            "ASE","SIT","CPR","ELM","YKM","LWS","LRD","OXR","SWF","SUX","YAK","RFD","YUM","BJI","CDV","SCE","EWN"};
```

Figure 6.2.2. Java implementation of Weka open source classifier for the testing algorithm.

- **Android application to predict flight delays:** After obtaining the model by training the Naïve Bayes algorithm using Java, an Android application was created and released on the Google Play Store for free, available at https://play.google.com/store/apps/details?id=couk.jenxsol.parallaxscrollviewdemo1

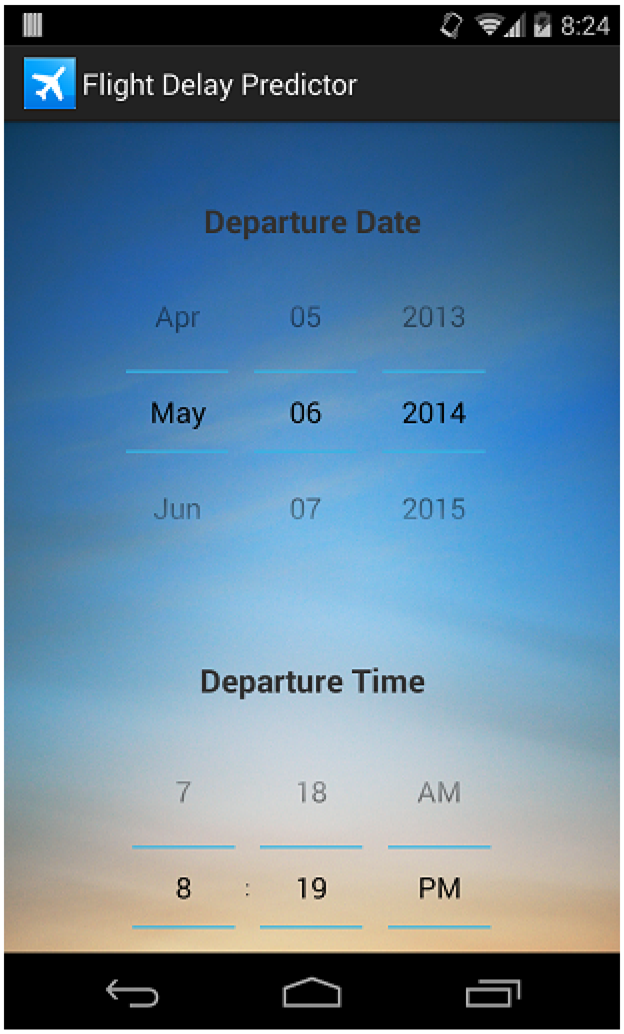
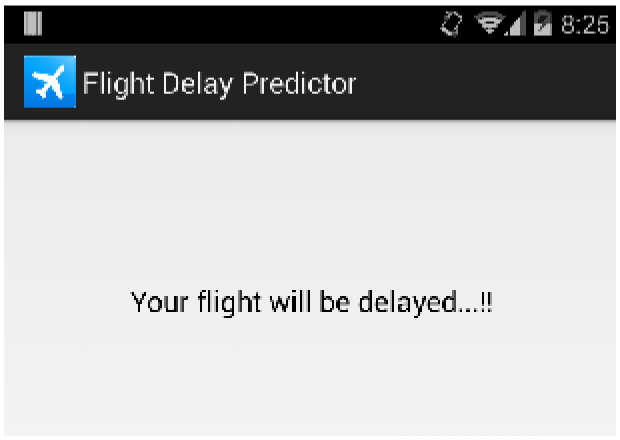


Figure 6.2.3. Android application for flight delays and cancellations.



Your flight will be delayed...!!

Figure 6.2.4. Prediction retrieved from the Android application.

- **Implementation of KnowledgeFlow:** This is a graphical front end to WEKA's core algorithm, used to present a data-flow for analyzing the flight delays and cancellations dataset.

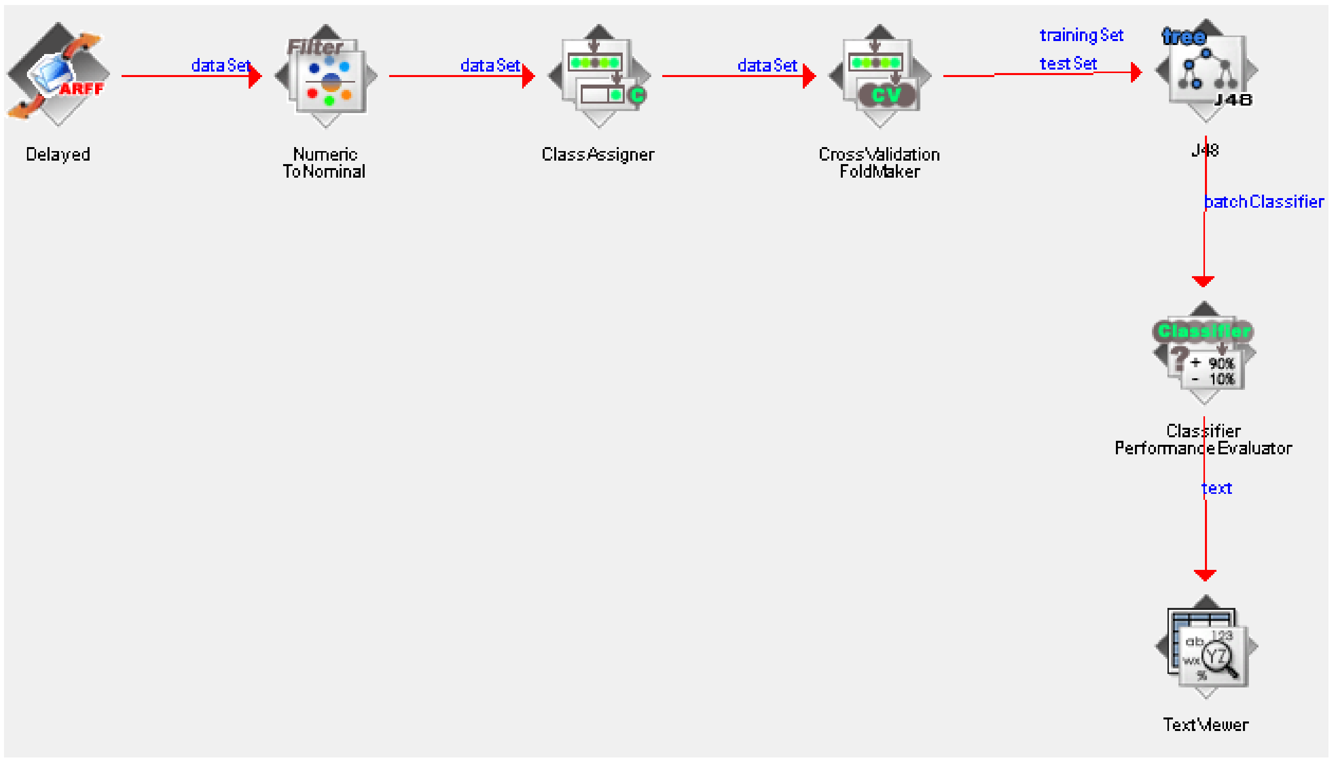


Figure 6.2.5. KnowledgeFlow developed on Weka.

- **Dynamic visualization of results using Tableau Software:** An interactive interface oriented to show the results of the analysis in a graphical environment was developed using Tableau Software. The interactive graphs are a visual representation that reveal the patterns found throughout this project, and the answers to the research questions generated at the beginning of this project.

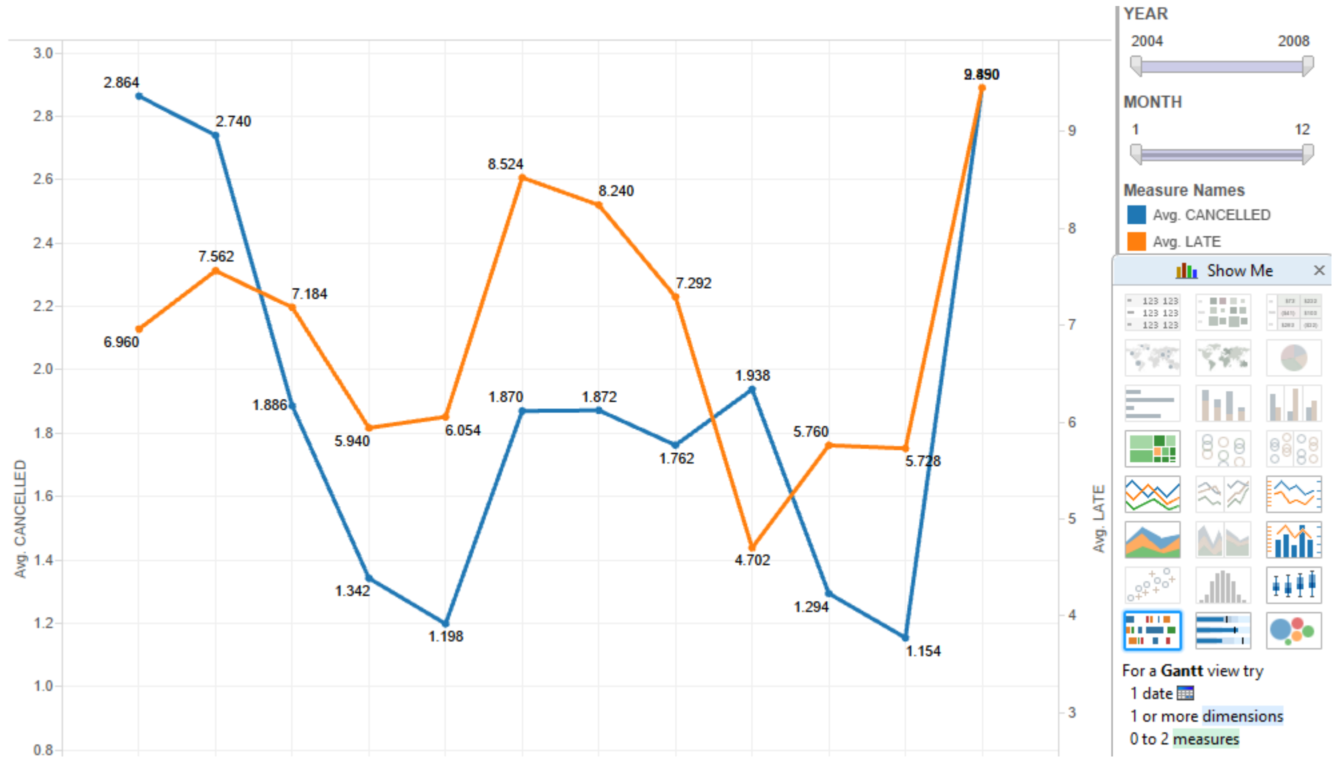


Figure 6.2.6. Interactive visualization of results using Tableau Software.

## 7. References

Airline on-time performance. (2009). Retrieved February 22, 2014, from http://stat-computing.org/dataexpo/2009/

Lazakidou, A. A. (2010). Web-based applications in healthcare and biomedicine. *Annals of Information Systems*, *7*.

Tan, P., Steinbach, M., & Kumar, V. *Introduction to data mining*. Indian Subcontinent Version. Pearson
Education, 2014. Print.

Xiong, J., & Hansen, M. (2013). Modelling airline flight cancellation decisions. *Transportation Research*, *56*, 64-80. http://dx.doi.org/10.1016/ j.tre.2013.05.003