A metadata collector for the guests on "Desert Island Discs"

A progress report submitted to The University of Manchester for the degree of MSc in Advanced Computer Science, in the Faculty of Engineering and Physical Sciences

May, 2012

Author: BALUSESCU, Alina Elena
Supervisor: Dr. Robert Stevens

School of Computer Science
The University of Manchester
Table of Contents

Abstract......................................................................................................................................................3

Chapter 1. Introduction..............................................................................................................................4
  1.1 Project objectives..................................................................................................................................4
  1.2 Report overview....................................................................................................................................4

Chapter 2. Background..............................................................................................................................6
  2.1 “Desert Island Discs” archive................................................................................................................6
  2.2 What is Metadata?..................................................................................................................................6
    2.2.1 Metadata types...................................................................................................................................6
    2.2.2 Metadata functions...............................................................................................................................7
    2.2.3 Metadata structure ..............................................................................................................................7
    2.2.4 Previous studies using metadata......................................................................................................7
  2.3 Collecting metadata for people............................................................................................................7
    2.3.1 DBpedia..........................................................................................................................................7
    2.3.2 SPARQL query language....................................................................................................................8
  2.4 Metadata for music records...................................................................................................................8
    2.4.1 MusicBrainz versus FreeDB................................................................................................................9
  2.5 Machine learning..................................................................................................................................10
    2.5.1 Supervised learning. Classifiers.........................................................................................................10
  2.6 Background summary...........................................................................................................................11

Chapter 3. Research Method......................................................................................................................12
  3.1 Research Methodology.........................................................................................................................12
    3.1.1 Data analysis and data cleansing......................................................................................................13
    3.1.2 Project evaluation strategy..............................................................................................................13
  3.2 Development tools..................................................................................................................................13
  3.3 Project Plan..........................................................................................................................................14
    3.3.1 Gantt chart.....................................................................................................................................15
    3.3.2 Risk analysis..................................................................................................................................15

Chapter 4. Project progress.......................................................................................................................16
  4.1 The list of guests.....................................................................................................................................16
    4.1.1 Checking names...............................................................................................................................16
    4.1.2 Checking links..................................................................................................................................16
  4.2 Collecting metadata about the guests..................................................................................................18
    4.2.1 Metadata of interest..........................................................................................................................18
    4.2.2 Querying DBpedia for metadata......................................................................................................20
  4.3 Collecting the choices and setting up the database.............................................................................21
    4.3.1 Classes........................................................................................................................................21
  4.4 Results Summary..................................................................................................................................22

Chapter 5. Summary and future work......................................................................................................23

List of References.......................................................................................................................................24

Appendix A – Project plan Gantt Chart....................................................................................................25
Abstract

This project seeks to collect metadata about the guests on the BBC radio 4 show, “Desert Island Discs” and to study whether the choices made on the show can be predicted based on this metadata. A person's choice may be influenced by many things, such as personal background, occupation, friends and social circles, life events or year of birth. The question discussed in this project is whether the information publicly available about a person can be enough to predict the music choices he or she will make on the show. This report presents the approach used in answering this question and it highlights the progress made thus far in achieving the goal set. The study starts with defining metadata, finding places where metadata is stored on the Web and information on how it can be accessed. Also, machine learning algorithms need to be identified in order to be used in the predictive model later on the project. The report continues with the proposed methodology and the final or partial results found. The first step of the method requires to pull out a list of all castaways from the BBC archive. Results in this step show some anomalies in a small percentage of the data that will be discard. Furthermore, some erroneous data in the BBC programme archive is highlighted. Analysing the results in more depth, it can be observed that a small percentage of the guests cannot be found on Wikipedia, the place chosen to collect metadata from. The method continues with step two, making a list with metadata of interest for the castaways, list that will be prioritised afterwards based on information availability. The next step is to design a database to store the metadata about castaways and their choices made on the show. The second and third step are in progress whilst the first one had been successfully completed. Further work in this dissertation project consists in following the rest of the steps from the method: collecting metadata about castaways based on the list made; linking each guest to his or her choices made on the programme; collecting information about music records and adding it to the database; and finally, using machine learning tools to make predictions about music choices for future guests on the show.
Chapter 1. Introduction

This project aims to create a collection of metadata for the guests of “Desert Island Discs” and to study to what extent this data can influence the choices made on the show. The final goal of the project is to build a model that will use the collection of metadata in predicting the choices for future guests. In this study, the prediction of music records will be prioritised, although the prediction of book and luxury item choices could also be investigated.

“Desert Island Discs” is a BBC radio programme where, in each episode, a guest must choose eight records, one book and a luxury item to take with them on a desert island. The show is structured as an interview, so that after picking each record, the guest would motivate his or her choice. At the end of the episode, the castaway must choose only one out of the eight records, as the “Castaway's favourite”. The programme is very well known in the United Kingdom and famous people from various industries appeared on the show so far. The idea of the show prompts the fans to think about what their own list of eight records would be. This choice may be influenced by many things, such as personal background, field of work, friends and social circles or year of birth. The question that arises at this point is whether we can predict these choices based on the metadata publicly available about the guests (for example on their personal Wikipedia page).

Metadata, widely known as “data about data” [25], can describe a wide range of entities, from persons to places, music, diseases or location. Many systems make use of metadata in their applications in order to tailor them to the user needs. For example, Amazon uses a recommendation system based on attention metadata [23] to suggest further items to the user, in a box called “Customers Who Bought This Item Also Bought”. A similar system is used by IMDb, to make movie suggestions, in a box called “People who liked this also liked …”.

Other systems use metadata to predict future behaviour. There is a large body of research into the analysis of different metadata and finding patterns that could be used in prediction algorithms. Particular interest was spotted in the study of socially generated data and how this data can be used in predicting things like social connections, movie ratings or what music do we listen to whilst performing an activity.

This project seeks to use metadata available about the guests on “Desert Island Discs” to predict music choices for future guests. For a better understanding of how this prediction could be made, we observe the following scenario:

“The BBC programme website announces the next guest on the show: John Smith. We go and collect metadata about this person and put in the software his date of birth, place of birth, occupation and other background information available about him. If John Smith was born in Edinburgh in 1946, works in a Theatre and is an alumni of the University of Manchester, the software would predict that he likes Musical Theatre songs, Irish traditional music and pop-rock songs from the early 70s”.

As observed from this scenario, the prediction of the exact artist and song is beyond the scope of this study, but might be investigated in a future research.

The successful completion of this project might rise other interesting questions: could we predict other things as well by analysing a person's metadata? Could the time and place where we were born, the place where we work, or our entourage, in general, influence the choices we make in life?

In the Computer Science field, these questions are interesting to investigate because they reveal the importance of metadata and the applicability of machine learning algorithms in solving real-life problems.

1.1 Project objectives

The main goal of the project is to predict music choices for future guests on “Desert Island Discs”. To ensure the project’s final goal is reached, it was divided into a set of objectives. Objectives are the fundamental part of project management and from them derive tasks and milestones.

---

The list of identified objectives is as follows:

- Pulling out a list of castaways from the “Desert Island Discs” archive and checking whether they can be found on Wikipedia.
- Deciding what metadata may influence the guests on choosing the eight records and collecting it from DBpedia using SPARQL queries.
- Gathering metadata about the music choices made thus far by past guests.
- Writing a set of queries on the data collected, to present statistical information about the show and its guests.
- Building a model that will use the collection of metadata in order to predict music choices for future guests on the show.
- Testing the results of every investigation and experiment made with the data set, to ensure data quality and to take further decisions accordingly.
- Creating an error management plan, to identify possible risks and back-up solutions.

1.2 Report overview

The remaining of this report is structured as follows:

Chapter 2. Background

This chapter offers information to situate the project in a wider context. At first, a short history of the BBC radio programme “Desert Island Discs” is presented. Next, the term of “metadata” is introduced. Types of metadata, metadata schemas and how metadata is used are also discussed in the section. This project uses metadata about people and music. This section also presents ways of collecting metadata about people from DBpedia, and investigates metadata schemas for music. Finally, some fundamentals of Machine Learning algorithms are presented as a base for the predictive model.

Chapter 3. Research Method

This chapter presents the methodology used to answer the research question addressed in Introduction. It describes the steps of the method, the process of data cleansing and the development tools needed for project completion. Also, the section presents the project plan and a risk analysis.

Chapter 4. Project progress

This chapter presents the progress made so far in this study. Final results were obtained in the first step of the method and partial results have been found in the second and third steps. Results are analysed and design decisions are also explained in this section.
Chapter 2.  Background

2.1 “Desert Island Discs” archive

“Desert Island Discs” is a long-running BBC radio programme, first aired on the 27th of January 1942 and still running now on BBC Radio4. During almost 70 years of broadcasting, the programme has had four presenters and over 2800 episodes. In each episode of “Desert Island Discs” one famous person is cast away on a desert island and is asked to pick eight music records to take with them. At the end of the episode, they must choose only one record to save from the waves, the “castaway's favourite”. They are given the Bible and the Complete Works of Shakespeare and are allowed to pick a book of their choice and a luxury item that cannot be used to escape the island. The format of the show started with the guests only choosing music records. The book and luxury item options were added 9 years later, in 1951 [2].

Due to copyright problems with the family of Roy Plomely, the inventor of the show and its first presenter until his death in 1985, the BBC archive of the programme was not available on-line until 2011, when the BBC reached a settlement with the family. However, since 2009, each episode has been available for seven days on the BBC iPlayer for catch-up [12]. The Castaway archive for all the guests from 1942 to current date is now available on the BBC Radio4 website and allows the fans to rediscover all the vintage episodes.

The process of building up the archive shows that the data associated with the programme is enormous, bringing together over 22,000 pieces of information consisting in record choices, books and luxury items [2]. The programme is structured as an interview where the guests relate their musical choices to key events in their life or people that inspired them up to that moment in time. Thus, the data must be handled in the context of the recording. Apart from the data generated by the choices of the guests, personal information about the castaway was recorded, for example, occupation and gender. After gathering all the data, a BBC team checked everything to assure there were no duplicates, missing information or data linked incorrectly. The cross-checking was made by constantly consulting the “Desert Island Lists” book by Roy Plomley [28] and archive copies of Radio Times. However, taking into consideration the volume of information processed, data cleansing is an on-going process. A castaway or an episode can be searched by castaway name, book choice and author, musical choice or luxury item. Episodes can also be sorted by date of broadcast or by castaway name, and browsing by categories like gender, occupation of the guest or by presenter is also available.

Thus far, the data available on the BBC radio 4 website can lead to statistics [2] about the most favourite musical records, the most requested artists or the most popular occupation of a guest, while we have no way of stating the average age of the guests.

This project will use the data available in the BBC “Desert Island Discs” archive as a starting point in achieving the defined objectives. Further details about this step of the study are explained in Chapter 3. Research Method of this report.

2.2 What is Metadata?

The final goal of the project, as described in Chapter 1. is to create a collection of metadata about the castaways from “Desert Island Discs” and build a predictive model for the music choices of future guests. Having defined what “Desert Island Discs” is about, the format of the show and the initial archive available on the BBC website, we need to introduce and discuss the concept of “metadata”.

Metadata is widely defined as “data about data” [25]. The term is used to describe information about resources [13]. The full understanding of the concept requires familiarity with metadata types, purpose or function, and structure.

2.2.1 Metadata types

There are three main types of metadata [25]:

1. **Descriptive metadata**, mostly used for discovery and identification purposes. This type of metadata allows resource identification based on relevant information and distinguish between similar and dissimilar resources. Some elements that classify as descriptive metadata for a music record may be title, singer, release date or label.

2. **Structural metadata**, indicates how compound objects are put together, for example, the order of the
tracks on the album, for a music record.

3. **Administrative metadata** provides information to help manage a resource. Details about file security and access rights, file type and other technical information, together with intellectual property rights are only some examples of administrative metadata [25].

### 2.2.2 Metadata functions

Metadata can be used in resource identification, organizing electronic resources, digital identification, enhancing interoperability or for preservation purposes [25], and these functions vary according to the type.

Descriptive metadata helps in discovering new, relevant information about resources. Therefore, giving the final goal of this project, that of building a model that can predict music choices for new guests on the “Desert Island Discs” show, it is justified to use descriptive metadata throughout the project.

### 2.2.3 Metadata structure

The structure of metadata is described in metadata schemas. A schema usually defines the names of the elements and their semantics. Specific content rules, representation rules for content, syntax rules or allowed values for elements may also be present inside a schema, but are not mandatory [25].

Metadata schemas depend on the system that uses the data and may be user-designed or standardized. The most common standard for metadata is the Dublin Core [5]. HTML Web pages use Dublin Core metadata for embedding important information, collecting it into Web indexes and then use it in search engines. An expanded list of the current standards available for metadata can be found in [19].

### 2.2.4 Previous studies using metadata

The process of creating metadata may differ according to the field of study, tools available, the resources it describes and the reason why it is created. Some categories of tools for creating metadata are Templates, Markup tools, Extraction tools and Conversion tools [25].

A large body of research illustrates several methods of creating metadata and how important metadata is in analysing patterns and generating new information. Some research projects need to create new metadata based on user behaviour, as it is the case in [23], where the authors record user logs to create attention metadata. However, in the majority of the studies, metadata is created with various extraction tools, from data already available online.

Previous studies show a particular inclination in analysing data about user preferences and finding patterns that could predict future choices, interests or social connections. For instance, in [14] the authors analyse the connection between music genre or artist and the usage suggested by user reviews in the field “Great Music to Play While”. Further examples describe the use of metadata in rendering personalized videos [18], in predicting movie ratings [36] or in determining the opening weekend for a movie [16]. Social networks are another source of valuable metadata and from this field we identify the study of predicting social connections based on similar interests [32] and the analysis whether socially generated metadata could be influenced by user reputation and content analysis of comments [17].

### 2.3 Collecting metadata for people

There are many databases that store metadata on Web in various formats, such as RDF [39], XML [6], CSV or in SQL databases. This section presents an important source of metadata available on Web, DBpedia [4] – an RDF format of the Wikipedia encyclopedia, that will be used to gather metadata about the castaways. To retrieve metadata from DBpedia, SPARQL queries [40] will be used.

### 2.3.1 DBpedia

DBpedia is a database that keeps structured information from Wikipedia in an RDF format. Wikipedia is one of the most consulted sources of information, constantly updated and maintained by contributors. DBpedia allows users to use query this information and link other data sets from the Web to Wikipedia data.

---


The DBpedia knowledge database describes over 3.64 million things, out of which 1.83 are classified into a consistent ontology, including 416,000 people, 526,000 places, 106,000 music albums, 60,000 films and 169,000 organizations. To describe all this things, DBpedia uses over 6,200,000 million links to external RDF datasets, 740,000 Wikipedia categories, 2,900,000 YAGO7 categories and over 9 million links to images and external websites [9].

The DBpedia project uses an information extraction framework to map the data from Wikipedia. The current extraction framework divides into two modules: the Core module, which contains the components of the framework and the Dump extraction Module, which contains the DBpedia dump extraction application. The components of the Core module are the following packages: Source, WikiParser, Extractor and Destination. The extraction application starts from a source of Media Wiki pages, over which the Source package provides an abstraction. After that, the WikiParser designates a parser to transform the source pages into a Abstract Syntax Tree (AST). Next, an extractor is mapping each node of the tree to a graph of RDF statements, over which the Destination package provides an abstraction. Currently, DBpedia consists of 1 billion RDF triples and is set to realize regular dump extractions [9].

The DBpedia knowledge base can be accessed over the Web through four different mechanisms: Linked Data, SPARQL endpoint, RDF dumps and Lookup index [4].

2.3.2 SPARQL query language

SPARQL (Simple Protocol and RDF Query Language) is a specific query language for the Semantic Web, and follows a similar structure as the SQL query language. The World Wide Web Consortium (W3C) [37] explains how SPARQL queries work: “SPARQL queries are based on (triple) patterns. RDF can be seen as a set of relationships among resources (i.e., RDF triples); SPARQL queries provide one or more patterns against such relationships. These triple patterns are similar to RDF triples, except that one or more of the constituent resource references are variables. A SPARQL engine would returns the resources for all triples that match these patterns”.

Query forms

The W3C recommendation for SPARQL [40] identifies four query forms:

- **SELECT** – “returns all, or a subset of, the variables bound in a query pattern match”
- **CONSTRUCT** - “returns an RDF graph constructed by substituting variables in a set of triple templates”
- **ASK** - “returns a boolean indicating whether a query pattern matches or not”
- **DESCRIBE** - “returns an RDF graph that describes the resources found”

Query example

The SPARQL queries used in this project to retrieve metadata from DBpedia are SELECT queries. An example of a SELECT query is the following:

```sparql
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
    ?person foaf:name ?name .
    FILTER (LANG(?description) = 'en') .
} ORDER BY ?name
```

The first three lines indicate the namespaces8 used in the query and their prefixes. These namespaces will be used to find the appropriate RDF graphs to be matched. In SPARQL queries, variables are preceded by a “?” or

---

“$" symbol. The FILTER statement allows the developer to set further constrains on the results.

The example provided searches for the name, date of birth, place of birth and description of all the Alumni of The University of Manchester. The resource that matches the criteria will be kept in the ?person variable and the ?name, ?birth, ?birthPlace, ?description variables will store the values found.

2.4 Metadata for music records

To achieve the project's goal we also need to collect metadata about the music. This section discusses what music metadata schemas would be useful to use in this study and why.

The increased use of digitized music influenced music related applications to use more specific metadata. Music metadata can describe different things, from the length of the song, title, genre and artist, to rhythm patterns, pitch and timbre. There are many music standards used in industry, some following the standard metadata formats, like the Dublin Core, and some evolving out of applications' design. At the moment, there is no music metadata standard to describe all the requirements possible. However, a choice can be made based on the application domain and metadata fields required [8].

A previous study [8] presents a thorough analysis of the expressiveness of various music metadata standards and their relation to application domain and categories of metadata fields. The application domains they consider in the study cover the cycle of music from creation to consumption and transactions:

- Music library/encyclopedia
- Personal collection management
- Commerce and transactions
- Music editing/producing
- Music playback
- Music recommendation
- Music retrieval
- Musical notation

For metadata standards, the authors made their choice based on industry standards and selected eight relevant schemas: ID3\[^9\], FreeDB [10], MusicBrainz [21], Dublin Core, Music Vocabulary\[^10\], Music Ontology [29] and MPEG-7\[^11\].

Due to the extremely large number of metadata fields, the authors clustered them into the following: Musical info, Classifiers, Performance, Versioning, Descriptor, Rights & Ownership, Playback rendition, Lyrics, Grouping & referencing, Identifiers, Record-info, Instrumentation & arrangement, Sound & carrier, Event, Time-modelling, Musical notation, Attention-metadata & usage, Publishing, Compositions, Production and Meta-metadata [8].

The algorithm described in [8] compares the application domains, metadata field clusters, and music metadata standards, each pair at a time. Next, the authors normalize the results and compute the decision table. Analysis on the latter shows that it is possible to keep any kind of information in a MPEG-7 format since it scores well in all application domains. Moreover, same results highlight the suitability of some formats to a specific application domain. This is the case of FreeDB and MusicBrainz, that scored low overall, but appeared to be the best choice for music retrieval applications. Because collecting metadata for the music choices made by “Desert Island Discs” castaways falls into music retrieval application domain, based on [8], the most appropriate music formats are FreeDB and MusicBrainz.

2.4.1 MusicBrainz versus FreeDB

Both standards originate from the same idea, that users can upload information about their CDs into the database and share it with other users. The idea was first implemented in 1996 in the form of a free database, CDDDB (Internet Compact Disc Database), but because there was no moderation system, the database became filled with typos and duplicates [33]. Furthermore, at one point CDDDB became commercial and that created the need of another free system with somewhat the same functionality. FreeDB is a community version of the

---


commercial CDDB and is based on its last free version. MusicBrainz offers the same basic functionality as FreeDB, only in a more structured format and with a moderation system, that allows users to correct information in the database. The main difference between FreeDB and MusicBrainz is that the latter is a Semantic Web Service [33] (an RDF-based Web Service). Having an RDF format, MusicBrainz gives URIs to the major items in the database (such as Track, Artist, Label, Release and so) so that they can be referred to in other applications. Also, querying the database becomes a simpler process through a RDF-based application interface.

With respect to the metadata available in MusicBrainz, there are six core entities in their database, that correspond to the Resources on the Web Service: Artist, Label, Recording, Release, Release-group and Work. Instances of the entity “Work” are usually songs. Apart from the core entities, there are three non-core entities: Rating, Tag and Collection. Music genre is not an available Resource in MusicBrainz, although works of Classical music, Opera, Musical Theatre and Live Bootlegs are uploaded following a different style than the rest of the songs [22].

On the other hand, FreeDB offers information about music genres, even if it is in a free-text form. They classify music in 11 genres because most programs rely their functionality on this exact list [10]: blues, classical, country, data (ISO966012 and other data CDs), folk, jazz, newage, reggae, rock, soundtrack, misc (others that do not fit in the above categories).

Based on the arguments presented above, this project will use the music metadata available on MusicBrainz in order to complete the metadata collection about the “Desert Island Discs” guests with their music choices. However, music genre is an important criteria in classifying music records, thus further investigation might be needed in gathering this parameter from FreeDB.

2.5 Machine learning

Knowing that metadata about music and people is the hypothesis of predicting the choices for the guests on “Desert Island Discs”, we need to discuss the algorithms that can make these predictions. This final step of the project focuses in analysing some existing Machine learning algorithms and using them to meet the goal set.

As described in Chapter 1, a prediction result will look like this: a new guest, born in London in 1960 and working in Academia will pick a mixture of classical records and rock music from the 80s or some 1960's folk music and pop songs from the 1990s. The prediction of the exact artist and record is beyond the scope of this study and is proposed as future research.

“Machine learning is programming computers to optimize a performance criterion using example data or past experience” [1]. In other words, Machine learning is the science that allows computers to learn how to perform based on data or past experience.

Alpaydin [1] classifies Machine Learning algorithms in:

- **Learning associations** – One application that uses Learning associations is “finding associations between products bought by customers: If people who buy X typically also buy Y, and if there is a customer who buys X and does not buy Y, he or she is a potential Y customer”.

- **Supervised learning** – Supervised learning algorithms use datasets of “training examples” to create a rule that maps input data to a desired output. Once the rule is established, new predictions can be made for future data. Classification and Regression are supervised learning algorithms.

- **Unsupervised learning** – In Unsupervised learning only input data is provided. The purpose is to find regularities in that data and predict the output.

- **Reinforcement learning** – Some systems have a sequence of actions as output. In this case, reinforcement learning algorithms learn from past experience and generate sequences that lead to the desired outcome. For example, in a game of chess a reinforcement learning algorithm will be able to generate a successful game strategy.

Supervised learning algorithms can be applied to achieve project's goal. The choice is justified by the metadata collected: we have the input data, which is metadata about the guests and we also have the output data, the choices made by past castaways. Thus, we can use this collection in a supervised learning algorithm.

2.5.1 Supervised learning. Classifiers

A supervised learning algorithm uses datasets of “training examples” to infer a function that will map input data to the a desired output. The “training examples” are named “training data” and the inferred function is called a classifier. A classifier must be tested on another set of data called “testing data” to measure its performance.

Handling the initial data set

In supervised learning, the initial data set must be split between training data and testing data. The difference between training data and testing data is illustrated in [7] where Brown describes it as the difference between driving lessons and driving test: one will learn how to drive on some familiar roads, but the driving test will be on unseen roads, thus skills must be generalized. Therefore, we first train the classifier on some training data and afterwards we test the classifier on another set of data, the testing data, to measure its performance.

There are many ways to split the initial set of data. For large datasets, as is the case for this project (we analyse the choices of 2602 guests, as described in Section 4.1.1), is best to use the K-fold cross-validation technique. In K-fold cross-validation the initial data set is divided randomly into K equal parts. First, the classifier trains on the first part and is tested on the rest of K-1. Next, the classifier trains on the second part and is tested on the rest of K-1 and so on until each part is used as a training set. This will ensure that the classifier is not fine-tuned for the data and is able to generalize well for new values (is not over-fitting the data).

Algorithms and measuring performance

There are many algorithms that use supervised learning and they all perform differently based on the input data and the required outcome. In this project I will compare a list of algorithms based on their performance on the collection of metadata about the castaways and their music choices. The algorithms that I plan to analyse are: Random Forests [6], Logistic Regression [7], SMO [27] together with Naive Bayes, Bagging, AdaBoost, Ensemble Selection and SVM, described in [1]. The algorithm that performs best will be chosen for making the prediction.

There are four possible cases when making a prediction: “For a positive example, if the prediction is also positive, this is a true positive (TP); if our prediction is negative for a positive example, this is a false negative (FN). For a negative example, if the prediction is also negative, we have a true negative (TN), and we have a false positive (FP) if we predict a negative example as positive” [1].

Some performance measurements based on the four possible cases described above are [1]:

- **Error** = (FP+FN) / N
- **Accuracy** = (TP+TN) / N = 1 - error
- **Precision** = TP / P’, where P’ is the total number of retrieved records
- **Recall** = TP / P, where P is the total number of relevant records.
- **F-measure** = \[ \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

In this study, I will compare the performance of the algorithms presented above, based on these performance measurements and decide which algorithm will be used in the final prediction.

2.6 Background summary

This chapter has set the project into a wider context. A short history of “Desert Island Discs” has been provided and also a description of the programme's archive. The concept of metadata was introduced and databases that hold metadata about people and music have been analysed. For this project, I decided to use DBpedia and SPARQL queries in collecting metadata about people, and MusicBrainz and FreeDB for gathering metadata about music.

Also, we took a look at some basic concepts of Machine learning and decided what will be used in the final step of the project. Thus, I will compare eight algorithms based on how their perform on the final collection of metadata (about castaways and their music choices) and pick one to use in the final prediction of music choices for future guests on the show.

Having set the background of this study, in the next chapter I will describe a research methodology proposed for successfully meeting the objectives of this project.
Chapter 3. Research Method

3.1 Research Methodology

As described in Chapter 1, this project seeks to answer the question whether the information publicly available about a person can influence their choices on the “Desert Island Discs” programme. In this study, the priority is to predict the music choices of a castaway. However, book and luxury item choices could also be predicted based on the same metadata.

The method proposed to answer this question follows the project objectives defined in Section 1.1. These steps are also reflected in the project plan described in Appendix A and are as follows:

**Step 1. The list of guests.**

The first step of the project is to develop a web-scraper script to collect information about the guests of “Desert Island Discs”, from the BBC radio 4 website. For each episode of the programme, the website presents the date of broadcast, name of the castaway, occupation at the moment of recording, gender and a link to their personal Wikipedia page. If the link is not provided on the BBC website, I will look for the guests directly on Wikipedia and test if they have a personal page.

The deliverable associated to this step is a list of all the castaways, their personal link of Wikipedia (if available) and metadata collected from the BBC archive: gender, date of broadcast for the episode and occupation.

**Step 2. Collecting metadata about people.**

Having the list of guests, the next step is to consider what information about them might be useful in predicting their music choices. This step consists in finding the appropriate list of metadata parameters that might influence the choice of person. Afterwards this metadata will be collected from DBpedia using SPARQL queries (see Sections 2.3.1 and 2.3.2).

The expected deliverable for this step is a list with all the castaways and the metadata that describes them.

**Step 3. Collecting the choices and setting up the database**

Given the format of the show, where each castaway chooses only eight records, one book and one luxury item, it is convenient to keep the data in a relational database. After completing Step 2, all the information collected will be stored in the database. Next, a web-scraper will be used to gather the music choices made by each guest on the show. Afterwards, every individual can be linked to the episode when he or she was invited as a castaway and to their musical choices.

The final result of this step is a MySQL database that will describe the guests of the show, what choices they made (the eight records, the book and the luxury item picked) and offers information about the music records.

**Step 4. Collecting metadata about music**

In the next step of the method I will collect metadata about music choices and add it to the database. I will gather most of the parameters (such as Artist, Recording, Work or Release) from MusicBrainz and will study the possibility of merging this data with the information from FreeDB, in order to store the music genre parameter as well (see Section 2.4.1).

The deliverable associated with this step is a complete database that can describe the guests of the show, what choices they made (the eight records, the book and the luxury item picked) and offers information about the music records.

**Step 5. Query the data**

In this phase I will write a set of queries for the database created in the previous steps. The purpose is to extract statistical information about the “Desert Island Discs” programme based on the information collected. After this step, I will be able to answer questions such as what is the average age of a castaway or where did most of the guests study.

**Step 6. Develop a predictive model.**

The final step of the method proposed will cover the process of developing a predictive model for music choices of future guests on the show. Tasks associated with this step include normalizing the collection of
metadata in order to clearly define the parameters to be used in the algorithm and using the collection for training and testing a classifier.

Giving the characteristics of the data set, mostly nominal values, the classification must be made with algorithms that perform well for this kind of data. Thus, I made a list of eight algorithms that are known to give good results for nominal values (see Section 2.5.1). I will use a machine learning software, RapidMiner (see subsection Machine Learning Software), to compare their performance on the collection of metadata gathered in the previous steps and choose one to use for the final prediction.

I will use 10-fold cross-validations for splitting the data into testing data and training data and will measure the performance taking into account Accuracy, Error, Precision, Recall and F-measure values (see Section 2.5.1).

### 3.1.1 Data analysis and data cleansing

Because the success of this project depends a lot on the quality of data (For example, if the classifier uses incorrect data, the predictions will be inaccurate), data must be under constant evaluation. Therefore, each of the steps from the research method described above include a phase of results analysis and data cleansing. Data analysis can be made in different ways, depending on the type of data, the volume of data and if the system is safety-critical or not.

This project manipulates a large volume of nominal data and is not safety-critical. Therefore, I propose the following three methods to validate data:

- Automatic testing for illegal characters or empty fields.
- Parity checks – when appropriate.
- Having a manual survey on a sample of 1% of the data and analysing the results.

To the extend of my knowledge, there are not many ways to validate nominal data, at the semantic level. Thus, I will use data validation at character level and check for illegal characters or empty fields, as proposed in the first method. The second method can be used to validate some one-to-many relationships from the database, for instance, that each guest has exactly eight records assigned per episode or that a single book was chosen. The third method is justified by the need of semantic validation that will be done manually. The percentage used in the survey is sensible giving the volume of data and the fact that the system is not safety-critical.

Data cleansing will be done manually after analysing the results produces by the surveys. Some erroneous results may be produced by code that does not cover all the possible exceptions. In this case, the code can be adjusted accordingly and new results are to be generated. The new results are analysed, thus this process may be recurrent. Other results may be generated in a correct way, but be incorrect in the source database. For example, one script may return the Wikipedia link of a castaway correctly from the BBC archive, but if the link is wrong in the archive this leads to erroneous data in the final collection of metadata.

Whatever the source of inaccurate data might be, through data analysis and data cleansing I ensure the classifier will use high quality data to predict the music choices of future guests on the show.

### 3.1.2 Project evaluation strategy

Evaluation can be done by first collecting metadata about the next guest on the show announced on the BBC page, next predicting his or her music choices and finally listen to the episode and check whether the predictions were correct.

This evaluation strategy can be applied due to the “Desert Island Discs” format. Each year, the programme runs about 40 episodes (sometimes 41 or 42) [2]. As of May 2012, the list of guests collected for this study numbers 13 guests out of the 2012 season (see Section 4.1 for more details about the list of guests). This presents the opportunity to use the episodes that will broadcast in the period May 2012 – August 2012 for project evaluation.

### 3.2 Development tools

So far, I have described the research methodology that I will use to complete this project and defined the way in which it will be evaluated as successful. Next, I will present the tools needed to meet the objectives of this project:
RubyMine

RubyMine is “a Ruby and Rails IDE with the full stack of essential developer tools, all tightly integrated into a convenient and smart development environment” [15]. RubyMine will be used in the process of writing the web-scraper scripts. I selected Ruby for this step because it is a flexible programming language and offers a wide range of standard libraries used for scraping web pages. In this project I will use the “net/http”[31] and “nokogiri”[26] libraries to parse the web pages and collect information.

Eclipse Classic

Eclipse is “an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle” [34]. Eclipse offers various packages to suit the needs for different developers, for example Java, Java EE, PHP, C/C++ or Javascript Developers.

In this project I will use Eclipse IDE for Java Developers as an environment to write the code needed to extract metadata from DBpedia, using SPARQL queries (see Section 2.3.2).

SPARQL endpoint

A public SPARQL endpoint is a point (usually a web page) where the user can ask SPARQL queries against a database. A list of currently available SPARQL endpoints can be found at [38]. In this project I will use the DBpedia public SPARQL endpoint for constructing initial queries against the database. Later on, these queries will be sent through a specialized query engine, offered by Jena API.

Jena API

Jena is a Java API for developing Semantic Web Applications [35]. The API is used to enable Java developers to write code that uses RDF data. It also includes “a query engine compliant with the latest SPARQL specification” [35]. Jena can be easily integrated in the Eclipse environment as an external library. I will use Jena in this project to be able to write SPARQL queries against DBpedia.

Machine Learning Software

There are two major open-source systems for data-mining and machine learning: WEKA [20] and RapidMiner [30]. The Waikato Environment for Knowledge Analysis (WEKA) is a software that “aims to provide a comprehensive collection of machine learning algorithms and data preprocessing tools” [11]. WEKA supports its own data format ARFF, CSV, LibSVM files and C4.5 files [11]. RapidMiner is the “world-leading open-source system for data mining” [30]. The product's data sheet [30] specifies that the WEKA machine learning library is fully integrated in the system. Furthermore, RapidMiner has access to a wide range of data sources including Excel, MySQL, Oracle, IBM and many more data formats.

Having the WEKA machine learning library fully integrated in RapidMiner and the metadata collection stored in a MySQL database, I decided to use RapidMiner over WEKA, for building the predictive model.

3.3 Project Plan

A successful project requires good planning. A proposed methodology, as that presented in Section 3.1, is not enough to ensure the project will be completed on time and with the desired results. Throughout the project I will use a Gantt chart to ensure proper time management and also to keep up with the progress made. Risk management will ensure that there will be few to no impediments in finishing the tasks from the project plan.

Weekly planning will be used to evaluate progress and impediments on a regular basis. Also, this will break the activities from the Gantt chart into smaller tasks, thus reflecting delays that might appear in the initial plan. The Gantt chart is prone to change in time, to adapt to changes identified through weekly planning.

3.3.1 Gantt chart

A Gantt chart contains the activities set for a project and their deliverables, in the form of “tasks” and “milestones”. The Gantt chart for this project, as of May 2012, can be found in Appendix A. The tasks in the chart reflect the research method proposed for this project. Because the time intervals between “Data analysis and data cleansing” phases are relatively short, these phases can be represented on the chart as one ongoing task. The list of milestones consists of the major deliverables expected at the end of a set of tasks.
List of milestones

Milestone 1. Preliminary report (due March 8th, 2012)
Milestone 2. List of guests complete (due March 15th, 2012)
Milestone 3. Progress report (due May 11th, 2012)
Milestone 4. A database of metadata about people (due June 7th, 2012)
Milestone 5. Music metadata collected (due July 1st, 2012)
Milestone 6. Predictive model complete (due August 1st, 2012)

As reflected in the list above, other important tasks and events will occur throughout the period set for the development of the project. These refer to writing the project documentation and must be planned accordingly.

3.3.2 Risk analysis

There are some impediments that may appear throughout this project that can produce delays in the project plan. Some things that could go wrong as well as a proposed alternative solution, are:

• **Metadata of interest not found on DBpedia.** - This was identified in the beginning of the project as a potential problem that may appear in the first step of the method proposed. The recommended solution was to query other databases for the information needed. The problem that actually appeared in step 1 was that not all metadata found on DBpedia was structured as expected, in attribute, value pairs (for more details, see Section 4.1). This will lead to small adjustments in the database design and the SPARQL queries written to extract metadata from DBpedia.

• **Problems merging the information from MusicBrainz with music genre from FreeDB.** This problem may arise in step 4 due to incompatibility between MusicBrainz and FreeDB in the way they store information (Free-text for FreeDB and RDF format for MusicBrainz). One solution is to study the structure of others databases that store music genre and try to collect the genre parameter from there. A starting point in finding the appropriate database may be the list of SPARQL endpoints available at the moment [38].

• **The classifier cannot predict eight records for one person.** If this is the case and the classifier cannot predict eight music choices for a future guest, the problem addressed will be simplified to predict only one track per guest, the “castaway's favourite”.
Chapter 4. Project progress

This chapter presents the progress made so far in this dissertation project. This chapter follows the methodology described in Section 3.1 and presents final or partial results. At the moment of writing (May 2012), step 1 is complete and steps 2 and 3 are in progress. Thus, the progress can be reflected only in these three steps: (1) the list of guests, (2) collecting metadata about guests and (3) collecting the choices and setting up the database.

4.1 The list of guests

The initial step of the project is to develop a web-scraper script to collect information about the guests of “Desert Island Discs”, from the BBC radio 4 website. For each episode of the programme, the website presents the date of broadcast, name of the castaway, occupation at the moment of recording, gender and a link to their personal Wikipedia page.

Thus far, the BBC archive contains 2891 episodes of “Desert Island Discs” from 1942 to May 2012. The first step is to pull out a list of guests and check whether they are on Wikipedia or not. Also, because some celebrities appeared more than once on the show, the web-scraper script must distinguish between different guests and must count if a person appears more than once.

Results have been analysed separately, first for the list of names and second for their Wikipedia page link.

4.1.1 Checking names

The first run of the script returned a list with 2630 different guests out of which:

- 229 people appeared twice
- 13 people appeared three times
- 2 people, Arthur Askey and David Attenborough, were on the show four times.

Data analysis and data cleansing

Data was validated, first by automatically checking for empty values or illegal characters and second by manually looking at a sample of 30 guests (1.14%).

Checking for empty values.

The test for empty values found 1 empty row, whilst the rest of 2629 were correctly identified names. A cross-check with the BBC archive, in search of an answer for the error found, proved that the script was accurate and only for that specific episode, the name of the guest (Bryan Forbes) was not displayed on the episode's page. I corrected the entry manually, adding the right name and data.

Checking for illegal characters.

This search highlighted a percentage of 1.06% abnormal values (28 entries in the list of names). The abnormal values correspond to 29 episodes in which there were pair guests, such as “Duke and Duchess of Devonshire”, “John and Roy Boulting” or “Frank Muir and Denis Norden”. One of this pairs, “Elsie and Doris Waters”, appeared twice on the programme.

Considering that the music choices made by these pairs are a common agreement between them and cannot be linked separately to one person or the other, I decided to discard these values. Also, being only 1.06% of the total number of guests, discarding these values will not affect much the learning set for the music choice prediction. Thus, the total number of guests for which I will collect metadata is now 2602.

Survey of names

Manually looking at 30 random names on the list (1.15%), I found no errors in the data. All the entries analysed were correctly identified names of castaways.

4.1.2 Checking links

The algorithm in finding the Wikipedia link on the BBC archive is as follows: first, the script identifies the page of the episode in which a person is guest. Next, the page is parsed and the script searches for the “Related
links” section on the page. The section contains links related to the castaway, such as a Wikipedia page or personal website. Finally, the script collects the information that contains the string “Wikipedia” from the “Related links” section.

**Data analysis and data cleansing**

The script returned the following results:

- a list of 2377 guests with a Wikipedia page
- a list of 225 guests without a Wikipedia page (I discarded the 28 pair entries, as described in Section 4.1.1)

**Survey on Wikipedia links**

Manually checking a sample of 30 random links from the first list (1.26% of 2377), I found one link pointing to a Wikipedia “disambiguation” page. Checking the BBC website, I found that the error was in their archive: they linked a guest (Anthony Smith) to a wrong page.

I also checked 30 guests from the second list (13.33% of 225) to ensure that I could not find them on Wikipedia. The results show that 5 persons, out of the 30 tested, have their personal page on Wikipedia even if that is not stated on the BBC archive.

These results highlight an exceptional case that was not treated by the script: finding the people that are on Wikipedia but not in the BBC archive.

**Experiment**

One solution to the problem found in the first run on the script, could be to construct the Wikipedia link directly in the script and then search if the link exists. Thus, I realized an experiment to see whether this solution would find some or even all the guests that were found missing.

Wikipedia links for people are formed from the standard string “http://en.wikipedia.org/wiki/” concatenated with the name of the person, first letters capitalized, separated by underscore. However, further analysing the second list of guests (the 225 persons without a Wikipedia page), I noticed that some had honorific titles in their name, as it appears on the BBC page. As most of the time searching Wikipedia for a person and including a honorific title will point to a “disambiguation” page, I made the assumption that striping the title out of the name might return better results.

The most titles found in the list of 225 people are: “Dame”, “Sir”, “Baron”, “Baroness”, “Reverend”, “Professor”, “Dr”, “Rt Hon ”, “Rt. Hon.”, “MP”. Before constructing the link, the script will check for these titles and remove them from the name, if found.

After running the script again, I got the following results:

- a list of 2377 links from the episodes pages (as in the previous version of the script)
- a list of 79 guests found on Wikipedia
- a list of 146 guests not found on Wikipedia.

**Analysis on new results**

As the first list return exactly the same results and have already been sampled and tested in the first version of the script, I did not test them again.

The second and third list contain a reasonable number of people to be manually checked. All the 146 persons from the third list were correctly identified as not being on Wikipedia. The second list, however, was not that accurate. The list was cross-checked with the BBC archive to ensure that every person found was the wanted castaway. Results show that from a list 79 links:

- **36 of them (45.56%)** were correct links, pointing to the same person invited on the BBC show.
- **2 links** were as well correct, but because the castaway was strongly related to an organization, he or she does not have their own page; the link redirects to the organization (for example, a Foundation or a Circus).
- **10 of the links (12.65%)** were pointing to a person with the same name as the castaway but different occupation. The correct links existed on Wikipedia but contained either the middle name of the guest or the occupation inside the link. (For example http://en.wikipedia.org/wiki/Anthony_Smith_(explorer)).
- **31 of them (39.24%)**, because the name was very common, they pointed to a Wikipedia “disambiguation”
Interpreting results

One question that rises in this point may be why 179 guests do not have a Wikipedia page, although they are or were, celebrities invited to such a famous show. To answer this question, I picked a sample of 30 people (16.75% out of 179) and looked at the date of broadcast for the episodes they were invited in and as well, analysed their occupation.

Grouped by date of broadcast, the 30 people divide into eight decades: 2 persons in the 1940s, 6 in the 1950s, 11 in the 1960s, 4 in the 1970s, 2 in the 1980s, 3 in the 1990s, 1 in 2000s and 1 in 2010s. This suggests that more than two thirds of the people missing on Wikipedia were guests before 1970, so one explanation could be that some of them might have died.

Analysing the occupation of the 30 guests sampled, I found 11 people working in Entertainment, with jobs like actor, singer or dancer; 9 people working in Media: broadcasters, newsreaders, journalists or chairmen; one civil servant, one Royal photographer and 8 other jobs unrelated to each other, such as equestrian, auctioneer or wine expert. As to how these occupation are scattered in the eight decades presented above, I found that the people working in Media were all guests before 1965, most of the people working in Entertainment were guests between 1965 and 1980 and most of the other guests appeared more recently on the show, after 1980.

These results might be explained, in my opinion, by the Historical events happening in Britain at that time [5]: World War Two (1939 – 1945), War with Argentina (1982) or the Cold War (where Britain played a key role). Also, “Britain was a founder member of North Atlantic Treaty Organization (NATO) in 1949 and sent a contingent of troops to take part in the Korean War (1950 – 1953)” [5]. These events might explain why some of the people working in Media and Entertainment before 1970 do not appear on Wikipedia: maybe they were enjoying short-lived fame or they were famous only in the United Kingdom and they got forgotten over time.

Most recent guests on the show, from late 90s onwards that do not appear on Wikipedia, might be explained by the option of suggesting a castaway, offered by BBC to its audience. Thus, people can write to the BBC team and suggest a specific castaway to appear on the programme [2]. This might lead to guests that have some impact in their field, but are not very famous in general.

Even though metadata about 179 castaways is not available on Wikipedia, they will still be added into the database and linked to their music, book and luxury item choices. They might not affect the classifier in predicting music choices for future guests based on metadata, but they might influence the results of step 5 from the method proposed, Query the data (for details of the research methodology, see Section 3.1).

4.2 Collecting metadata about the guests

4.2.1 Metadata of interest

I created an initial list of metadata about people that could have an impact over the choices made by them on the show: date of birth, place of birth, age at the moment of the recording, sex, occupation, education, social class, hobbies, important people in their life, national background, culture, ethnicity, religious beliefs and political vision.

Whilst some fields on the list, like date of birth, place of birth, age at the moment of recording, sex and occupation are straight-forward and seem easy to collect, gathering other metadata might be more difficult. Fields such as hobbies, education level, schools they went to, the social class they are part of and other people they influenced or were influenced by throughout their life, might be important in setting the personal background of each guest. However, finding this information in a structured format and normalizing it so that it would fit a metadata schema, could be challenging.

Further aspects that I consider to influence personal decisions are related to national background, culture, ethnicity, religious beliefs and political vision. However, exploitation of a person's race, ethnicity, national origin, sexual orientation, political belief, religion, or mental or physical disability, violates the code of ethics
[24], hence I will not consider them in this study.

**Prioritizing the list of metadata**

Due to the challenges that might appear in the collecting process, the list of metadata must be prioritized. An important factor in prioritizing the list is information availability. Because the metadata set collected from the BBC radio 4 web page will be completed with information from DBpedia, a small analysis was made to check data availability.

**Survey of metadata availability on DBpedia**

I used a sample of 30 random guests (1.15% out of 2602) and manually checked their DBpedia pages. The results show that sometimes, some values appear more than once on the same page and that usually, the set of properties available differ from page to page.

For example, the occupation of a person may be described on DBpedia through several properties: dbpedia-owl:profession, dbpprop:occupation, dbpprop:shortDescription, dbpedia-owl:background or dc:description (as shown in Table 1), thus replicating data. One explanation for data repeating itself is that DBpedia uses an RDF format and some properties have different namespaces\(^{13}\) even though they describe the same thing.

| Metadata | Sample 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| dbpedia-owl:birthDate | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbpedia-owl:birthPlace | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbpedia-owl:stateOfOrigin | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:country | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:nationality | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbpedia-owl:residence | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:knownFor | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:industry | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:adopted | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:profession | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:occupation | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:shortDescription | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbpedia-owl:background | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dc:description | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbprop:influenced | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dcterms:subject | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Education_AsCategory | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| BirthDate_AsCategory | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| dbpedia-owl:birthDate | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| PeopleFrom_AsCategory | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

**Table 1: Survey of metadata availability on DBpedia**

I also noticed that even though the set of properties differs from person to person, the property dcterms:subject appears on every page and shows the Categories in which each person is included. DBpedia does not state explicitly the school and university where a person studied, but it does include them in a Category with a suggestive name such as “Alumni_of_the_University_of_Leicester”. The same happens for dbpedia-owl:birthDate, where every person gets included into a Category named “YearOfBirth_births” even if the dbpedia-owl:birthDate value is missing. The last four rows in Table 1 check the appearance of Education information, Birth Date, Occupation and Location, in the form of values of the dcterms:subject property and not as RDF properties.

**Statistics on metadata availability**

Table 2 presents the same results of the test, in the form of statistical information. Thus, we observe the followings:

- dcterms:subject property appears for all guests
- BirthDate_AsCategory appears in 96.67% of cases
- Occupation_AsCategory appears in 93.33% of cases
- dbpedia-owl:birthDate appears in 86.67% of cases
- PeopleFrom_AsCategory appears in 76.67% of cases
- Education_AsCategory appears in 56.67% of cases

• dbpedia-owl:birthPlace appears in 50% of cases
• dbpprop:occupation appears in 33.33% of cases
• dbpprop:shortDescription appears in 33.33% of cases
• dc:description appears in 30% of cases
• and under 10% for the rest.

<table>
<thead>
<tr>
<th>Property</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia-owl:birthDate</td>
<td>86.67</td>
</tr>
<tr>
<td>dbpedia-owl:birthPlace</td>
<td>50.00</td>
</tr>
<tr>
<td>dbpedia-owl:stateOfOrigin</td>
<td>3.33</td>
</tr>
<tr>
<td>dbpprop:country</td>
<td>3.33</td>
</tr>
<tr>
<td>dbpprop:nationality</td>
<td>13.33</td>
</tr>
<tr>
<td>dbpedia-owl:residence</td>
<td>6.67</td>
</tr>
<tr>
<td>dbpprop:knownFor</td>
<td>6.67</td>
</tr>
<tr>
<td>dbpprop:industry</td>
<td>3.33</td>
</tr>
<tr>
<td>dbpprop:adopted</td>
<td>3.33</td>
</tr>
<tr>
<td>dbpedia-owl:profession</td>
<td>3.33</td>
</tr>
<tr>
<td>dbpprop:occupation</td>
<td>33.33</td>
</tr>
<tr>
<td>dbpprop:shortDescription</td>
<td>30.00</td>
</tr>
<tr>
<td>dbpedia-owl:background</td>
<td>6.67</td>
</tr>
<tr>
<td>dc:description</td>
<td>33.33</td>
</tr>
<tr>
<td>dbpprop:influenced</td>
<td>3.33</td>
</tr>
<tr>
<td>dcterms:subject</td>
<td>100.00</td>
</tr>
<tr>
<td>Education_AsCategory</td>
<td>56.67</td>
</tr>
<tr>
<td>BirthDate_AsCategory</td>
<td>96.67</td>
</tr>
<tr>
<td>Occupation_AsCategory</td>
<td>93.33</td>
</tr>
<tr>
<td>PeopleFrom_AsCategory</td>
<td>76.67</td>
</tr>
</tbody>
</table>

Table 2: Survey of metadata availability on DBpedia. Statistical information

The analysis of these results shows that it is sensible to discard the properties that replicate information, keeping the property that has the higher percentage of appearance. It is also sensible to discard the properties that scored under 10%, because the rate of appearance is too small to be used in a prediction algorithm. Therefore, the remained properties available to collect information from, are: BirthDate_AsCategory, Occupation_AsCategory, PeopleFrom_AsCategory and Education_AsCategory.

The final list of metadata consists of age at the moment of recording, sex and categories describing Birth, Education, Occupation and Location.

4.2.2 Querying DBpedia for metadata

This section describes the progress made in the querying phase of Step 2. At the moment of writing, some progress has been made in learning how to write SPARQL queries (for more detail, see Section 2.3.2). I used the SPARQL endpoint available on DBpedia to construct and test queries that will return the metadata of interest for a given person.

To avoid the challenges in dealing with different people with the same name or other issues identified in Step 1, getting the list of names (Section 4.1), I will query DBpedia using SPARQL queries that have as a unique identifier the DBpedia link of the resource. The individual link for every resource in DBpedia can be automatically constructed from the Wikipedia link, by replacing the “http://en.wikipedia.org/wiki/” with “http://dbpedia.org/resource/”.
SPARQL query

The query built to return metadata of interest for a person is:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?name ?birthDate ?classifiedIn ?description ?person WHERE {
  <http://dbpedia.org/resource/Debbie_Harry> foaf:name ?name
  ?person foaf:name ?name
  ?person dcterms:subject ?classifiedIn
  ?person dbo:birthDate ?birthDate
  ?person rdfs:comment ?description
  FILTER (LANG(?description) = 'en'))
```

This query will return the name, date of birth, categories in which she got classified and a description in English, for Debbie Harry. Integrating this query model into a Java program will return a collection of metadata for the entire list of castaways.

4.3 Collecting the choices and setting up the database

This section describes the progress made in Step 3. of the methodology proposed. This step can be divided in two separate tasks: (1) designing and setting up the database and (2) collecting the music choices made by the guests. This section will describe only the process of designing and setting up the database, as the second task is still awaiting.

The format of the radio programme allows the data to be stored in a relational database. Illustration 1 presents the UML (Unified Modelling Language) class diagram [3] that describes how classes communicate inside the project. The database design follows the class diagram, where each class defines a table and the class attributes define the columns of the table.

4.3.1 Classes

Castaway

The class describes a person invited to the show. The attributes stored are “name”, “wikiLink” for the personal Wikipedia link, “yearOfBirth” and “occupation”, “school”, “university” and “location”, as found on the DBpedia page. In the database the required attributes are the unique identifier (class Id) and “name”. This class is associated to the Episode class because one Castaway appearsOn() one or more Episodes of the show.

Episode

The class describes one episode of the “Desert Island Discs” show and stores the following attributes: “age of castaway”, “date of broadcast”, “occupationOfGuest” at time of broadcast (as it appears in the BBC archive). This class is associated with the classes Castaway, Book and LuxuryItem through Foreign Keys. Also, each episode is associated with eight RecordChoices.

Book

The class describes the book choice made by a castaway on one episode.

Luxury Item

The class describes what luxury item was picked by a castaway on one episode.

Record

This class represents a music record picked by a guest in one episode of the show. Attributes like “artist” and “title” will be completed with information from the BBC archive. The rest of the attributes, “genre”, “label”, “recording” and “release” will be completed with metadata collected from MusicBrainz and FreeDB.

RecordChoice

This class appeared as a need to represent the music choices made on one episode. The Primary Key of the table is composed by the “EpisodeID” and “RecordID”. The “choiceOnShow” attribute represents the order in which the castaway picked the music records and can take values from 1 to 8. The “favourite” attribute is true if
the record was picked as the castaway's favourite and false if not.

4.4 Results Summary

This section summarizes the modelling choices made so far and also outlines the results obtained.

List of guests

In the process of analysing the list of names I took the decision to discard data from 29 episodes of “Desert Island Discs”, because the episodes had paired guests. The 29 episodes correspond to 28 entries in the list of names, thus the final number of guests to be analysed and used in the metadata collection is 2602 guests.

After analysing the presence on Wikipedia for the 2602 castaways, I found that 179 of them were not on Wikipedia and 2423 had their own Wikipedia page. Out of the 2423 people, 2377 had their personal link on the BBC archive and for 46 of them the BBC archive did not mention their Wikipedia page.

I took the decision to keep the 179 guests in the collection because they might affect the queries made later in the project even though they might not affect the final predictions.

Metadata of interest

The initial list of metadata proposed has been filtered and passed through a process of prioritization. The results show that the final list of metadata that will be collected about people is:

- age at the moment of recording
- sex
- Birth year
- Education
- Occupation
- Location

Illustration 1: UML class diagram
Chapter 5. Summary and future work

This report presents the progress made into the dissertation project that studies whether personal information about a guest on the radio programme “Desert Island Discs” can determine their music choices on the show. The project is based on the hypothesis that the choices of past castaways and personal metadata about future castaways are enough to make a prediction of the music choices of the latter.

Thus far, a set of objectives has been defined for this study and background research has been done for understanding the wider context of the project. Basic concepts used in the project, such as metadata, DBpedia, SPARQL, MusicBrainz and classifier, have been defined and explained.

A research method has been proposed to meet the objectives set for the project. Also, a short analysis has been made over the necessary development tools that will be used to complete the project. An evaluation opportunity has been identified and will be used to check the predictions' accuracy.

The progress on the project can be observed in the first three steps of the suggested method. The first step of the method has been successfully completed and based on the result's analysis, further development decisions have been made. The second and third steps are still in progress, thus only partial results have been obtained.

In conclusion, this report shows that the method proposed is giving good results, thus is a suitable approach in achieving the goal of the project. Further work include completing the rest of the steps mentioned in the methodology and building the predictive model.

The success of this project implies that we will be able to make correct predictions of music choices based on metadata about a person. If generalized, this shows the importance of time (date of birth), space (place of birth or residence) and entourage (education and field of work) in the decisions people make. From a Computer Science point of view, it would be interesting to investigate whether other choices, such as books, luxury items, paintings or places to visit, might be predicted as well by applying machine learning algorithms over metadata. This would prove that metadata can be used with machine learning algorithms in answering real-life questions.
List of References


# Appendix A – Project plan Gantt Chart

## Gantt Chart for "A metadata collector for the guests of Desert Island Discs"

<table>
<thead>
<tr>
<th>Id</th>
<th>Task</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Background research</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Write script to pull out the list of guests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Prepare preliminary report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Milestone 1 – Due March 8th, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Milestone 2 – Due March 15th, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Design the database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Gather metadata about guests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Query DBpedia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Prepare Progress Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Review Progress Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Milestone 3 – Due May 11th, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Add metadata about people to database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Semester 2 Exams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Milestone 4 – Due June 7th, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Collect music choices for every guest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Gather metadata about the music choices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Milestone 5 – Due July 1st, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Write a set of queries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Create a predictive model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Milestone 6 – Due August 1st, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Data cleansing (orgorg)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Prepare documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Review documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Submit dissertation report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Milestone Legend
- **Milestone 1**: Preliminary report
- **Milestone 2**: List of guests complete
- **Milestone 3**: Progress report
- **Milestone 4**: A database of metadata about people
- **Milestone 5**: Music metadata collected
- **Milestone 6**: Predictive model complete

### Task colour Legend
- **Green**: finished task
- **Yellow**: task in progress
- **Red**: task delayed
- **Grey**: task awaiting

**Author**: BALUSESCU, Alina Elena, 8123228  
**Supervisor**: Dr. Robert Stevens