Matching CVs based on EDISON Data Science Competencies (CF-DS) and advanced text analysis methods

Mathijs F. Maijer
11043024

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligente
University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Mr. Y. Demchenko
Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 29th, 2018
Data has started to play an important role in modern data driven research and industry; allowing for deeper understanding of different phenomena. Demand for new types of data science specialists who can support all stages of the data development life cycle; from data production and input, to data processing, visualisation and reporting, is growing. This has caused the gap between the supply and demand of data scientists to widen considerably.

There is a strong need to define a methodology and develop tools for effective job candidate CV assessment against job/employer requirements. The EDISON Data Scientist Framework (EDSF) provides a comprehensive multi-dimensional, multi-domain, and multi-component definition of the relevant data science competencies, skills and professional profiles as part of its Data Science Competence Framework (CF-DS) and Data Science Professional Profiles (DSPP). This framework classifies the data science related competencies in five different groups, that each contain six competencies. However, to assess a data scientist correctly using the EDISON Competence Framework is a complex task. Document similarity techniques along with regular expressions were used on the CV of a data scientist to create insights into the competencies of the CV’s creator. The aim of this research was to accurately portray the competencies of data scientists by using the CF-DS and to make this tool available and usable for data scientists and recruiters. The proposed solution has three components, a timeline indicating the career path of the data scientist, a graph showing competency scores based on document similarity, and a graph showing competency scores based on the career path. The best method to get an accurate overview of a data scientist’s competencies at this point in time, is to use all three components of the tool that was developed. The developed CV assessment tool and application provides an effective instrument for recruiters to assess CVs which allows them to decide on the most suitable candidates. It is also a useful tool for job seekers and practitioners to assess their competencies and identify a path for professional development.

Using the proposed solution, it has become possible for data scientists and recruiters to start a basic assessment process, which will in turn enable data scientists and even the data science community itself to develop. The project has delivered a working code that is included in the ongoing EDSF community development project: https://github.com/EDISONcommunity/EDSFapps
List of Figures

1 Example of job and CV matching with the EDISON competencies by Vural . . 6
2 Design of first visualisation tool screen . . . . . . . . . . . . . . . . . . . . . . . 13
3 Design of extracted text and uploaded document visualisation . . . . . . . 14
4 Design of how a career path can be visualised . . . . . . . . . . . . . . . . . . 14
5 Design of a graph showing a CV matched with all CF-DS competencies . . 15
6 Architecture of Angular web application (front end) . . . . . . . . . . . . . . 16
7 Architecture of the back end . . . . . . . . . . . . . . . . . . . . . . . . . . 18
8 Processing steps of CV and competency documents text . . . . . . . . . . . . 20
9 JSON format of competency documents . . . . . . . . . . . . . . . . . . . . . . . 20
10 Relations between data components of career path extraction . . . . . . . 21
11 JSON format of jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
12 Document similarity competence scores of a data scientist with 8 years ex- 
perience . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
13 Career competence scores of a data scientist with 8 years experience . . . 31
14 Career timeline of a data scientist with 8 years experience . . . . . . . . . . 31
15 Document similarity competence scores of a data scientist with no clear job 
names . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
16 Career competence scores of a data scientist with no clear job names . . . 33
17 Career timeline of a data scientist with no clear job names . . . . . . . . . . 33
18 Document similarity competence scores of a data scientist with 15 years ex- 
perience . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
19 Career competence scores of a data scientist with 15 years experience . . . 35
20 Career timeline of a data scientist with 15 years experience . . . . . . . . . 35
21 Document similarity competence scores of a Hadoop developer with 3 years 
experience . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
22 Career competence scores of a Hadoop developer with 3 years experience . . 37
23 Career timeline of a Hadoop developer with 3 years experience . . . . . . 37
24 Document similarity competence scores of a Hadoop developer with 15 years 
experience . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
25 Career competence scores of a Hadoop developer with 15 years experience . 39
26 Career timeline of a Hadoop developer with 15 years experience . . . . . . 39
1 Introduction

Data has started to play an important role in approaches to understand and analyse different phenomena. The concept for approaches that rely on data to gather results, is called data science. Data science is an interdisciplinary field that unifies statistics, data analysis, machine learning and their related methods. Techniques from multiple fields are used and combined, operating on data as their foundation, which allows for new insights to be acquired which were previously unavailable.

Especially in a world that is increasingly generating more data than ever before, data has started to play a vital role. Businesses, politicians, scientists, and other parties are starting to rely on data and the constant increase thereof. In the last three years, more than 90% of the world’s data has been created, showing no signs of decreasing. It has become increasingly complex not only to capture more relevant data, but also to interpret data that was already gathered.

Because of the enormous increase in available data, big data analysis has become very popular and big data experts are very sought after. This has caused the gap between the supply and demand of data scientists to widen considerably. Unfortunately, not only is it difficult to find capable data scientists, it is also difficult to define what a data scientist exactly is.

A project funded by the EU, called the EDISON project, was created to offer more insights into data science as a profession. Arguably the most important part of this project is the EDISON Data Science Framework (EDSF), which is a collection of documents that defines the data science profession. These documents have been developed to guide educators, trainers, employers, managers, and data scientists themselves. This collection of documents collectively breakdown the complexity of the skills and competencies needed to define data science as a professional practice.

An important part of the EDSF is the Data Science Competence Framework (CF-DS). This framework classifies the data science related competencies in five different groups, that each contain six competencies. These different groups define different aspects of the data science profession, like managing competencies, engineering competencies or analysing competencies. There are other EDSF components that map these competencies to knowledge and academic courses.

These thirty competence documents defined by the CF-DS can be used in a multitude of ways. One important usage is to assess how competent a data scientist is. If a data scientist would have a score for every competence, then this collection of scores can be used as an assessment. This assessment can then be used to further train a data scientist in a desired direction after gap analysis, or it can be used to select the best candidate for a job.

Unfortunately, assessing a data scientist by using the CF-DS is a complex task because there is no method that can easily be used to extract competence.

---

1EDISON Data Science Framework
2Data Science Competence Framework
scores from a person, nor is there a metric defining these competence scores. To be able to assess a data scientist in the best way, an entity that best describes a data scientist’s capabilities is required. Something that comes close to this requirement, is the Curriculum Vitae (CV) of a data scientist. This CV should be a collection of the education and professional experience that the data scientist has, which makes it the easiest retrievable, most descriptive document of someone’s capabilities.

But how can a CV be matched with the CF-DS to accurately represent the capabilities of a data scientist? This thesis will aim to find a solution to this problem by looking at different parts of a CV and the EDISON framework. Document similarity techniques along with regular expressions will be used together with the EDISON framework to create overviews that yield insights into the competencies of a data scientist. The following researching questions are addressed in this thesis:

1. How can a CV, which is likely to have no pre-determined structure and is often expressed in a free textual form, be used for assessment based on the competencies defined in the CF-DS.

2. How can the solution be easily used by relevant personnel, like data scientists for individual competence gap analysis, or by recruiters who want to assess a data scientist’s CV.

The thesis is structured as follows: the first part of this thesis will discuss related works: first more information will be given about the EDSF, after that two other works will be discussed that have been used as a foundation for this thesis. After discussing the related works, a theoretical foundation will be provided to describe the theory behind document similarity calculations and the retrieval and location of terms in a CV using regular expressions.

The method will be discussed afterwards, explaining the process of how the eventual results can be collected and which considerations and decisions have been made to create those results. The results will be explained and evaluated after the method has been described.

Finally the conclusion will provide a short summary and some more insights, before the final discussion will take place along with some possibilities for future work on this research topic.
2 Related work

This section will discuss relevant related works, that function as a foundation for this thesis. First the EDISON Data Science Framework will be examined in a more in-depth manner; the framework has more important parts than just the CF-DS that are of use for this thesis.

Next to this, two more works will be discussed that have started the research for matching CVs with the thirty competencies defined by the CF-DS. Spiros Koulouzis developed the EDISON COmpetencies ClassificatiOn (E-CO-2) service, which can be used to match CVs with competence documents through document similarity calculations. A recent paper by Baris Can Vural tried to reproduce the results of the E-CO-2 service using Apache Spark. This thesis uses datasets that were created by these previous works what makes it important to discuss them in-depth.

2.1 EDISON Data Science Framework

As mentioned before, the EDSF is a collection of documents that define the data science professions and a scientific domain. Before this framework, there have been no guidelines to easily break down the data scientist profession, which makes this framework highly relevant.

The framework was developed to support, guide, and accelerate the education process of data science professionals for desired purposes. Different parties can make use of this framework: educators can design and implement programmes to educate the professionals that are needed by research or professional organisations, employers will be able to understand competencies that are relevant for their cause, which will help improve the design and execution of recruitment procedures, and future data science professionals will be able to understand and select educational programs that better suit their own desired career and ambitions.

To effectively describe the data scientist profession and data science field, the EDSF has four main components:

1. Data Science Competence Framework (CF-DS)

The Data Science Competence Framework serves as a foundation for the whole EDISON framework. It provides a basis for all the other components of the framework and was defined in compliance with the European e-Competence Framework.

It classifies the relevant data science competencies into five groups: Data Science Analytics (DSDA), Data Science Engineering (DSENG), Data Management (DSDM), Research Methods and Project Management (DRSM), and Domain related competencies (DSDK).

These groups were chosen after job market study and data science vacancy analysis. Each group contains six competencies that each define a competence that belongs to the group. The document containing all the competencies is showed in appendix.
2. Data Science Professional Profiles (DSPP)

The Data Science Professional Profiles define which profiles a Data Scientist may have. There are twenty-two different profiles which in theory should encompass every data scientist profession. All Data Science Professional Profiles have a certain relevance for the competencies described in the CF-DS. These relevance scores are numbers from a scale of 0–9. A list of the profile names can be seen in appendix 10.2, next to that, the relevance scores that each DSPP has for the competencies from the CF-DS can be seen in appendix 10.3.

3. Data Science Body of Knowledge (DS-BoK)

The Data Science Body of Knowledge is linked to and based on the CF-DS, it is based on overviews and analyses of existing bodies of knowledge that are relevant to intended frameworks for Data Science. The DS-BoK provides a foundation for the definition of the Data Science Model Curriculum (MC-DS).

The DS-BoK defines six groups that contain knowledge areas groups (KGA) that are linked to the competence groups defined in the CF-DS. These KGAs consist of Knowledge Areas (KA) that represent important areas of knowledge for a data scientist, e.g: data analytics. Multiple Knowledge Units (KU) form a KA, these units represent the topics that are important in a knowledge area, e.g: statistics, which are important for the data analytics knowledge area.

4. Data Science Model Curriculum (MC-DS)

The Data Science Model Curriculum is built with the DS-BoK and the CF-DS as its core, it defines Learning Units (LU), which can be seen as courses. These learning units are mapped to the knowledge units of the DS-BoK, which can be used when designing or creating curricula for data scientists.

2.2 EDISON Competencies Classification service

Spiros Koulouzis has developed the EDISON Competencies Classification (ECO-2) service prototype. It is a distributed automated service that performs gap analysis for the data science profession. It calculates the similarity between documents, which can be combined with the EDISON taxonomy to identify gaps between education, the side that 'supplies' data scientists, and the industry, the side that 'demands' data scientists.

This service was created so that students, data analysts, educators and other stakeholders can gain insights into their current status and competencies, which will allow them to take steps in a desired direction. This was achieved by creating a reference dataset that is based upon the CF-DS; thirty vocabulary documents were created that each describe an individual competence of the CF-DS. In contrast to the competency definitions in the CF-DS, the individual documents in the reference dataset contain more relevant keywords.
and sentences associated with the competence that the document references, creating a much larger document.

Finally, a document, like a CV, can be compared to all the competency documents based on how similar the text is. The amount of similarity of the document with a competence document, serves as the score of the document for that competence. This way, a CV can be compared to all thirty documents that describe a competence, yielding thirty scores that indicate how competent someone is based on the CF-DS.

2.3 CV and Job Vacancy Matching

A master’s thesis by Baris Can Vural[7], describes the process he went through to create a tool that matches data scientist CVs and job vacancies (ads). This was inspired by the prototype of the service that Koulouzis developed and Vural aimed to reproduce the same results, except by using a different platform to calculate the competence scores. The dataset that he used to achieve this was also inherited from Koulouzis’ work.

The E-CO-2 service used the Hadoop MapReduce[13] framework, which is one of the most popular big data processing platforms. However, because of its performance penalties, Vural wanted to use the Apache Spark[14] platform to reproduce the results with better performance. Instead of using the same method to calculate document similarity, Vural experimented using multiple document similarity calculation algorithms and visualised his results in a web application written using the Angular framework[15].

The results of a document similarity calculation with the competence documents were visualised in the form of a radar chart, also called spider chart. These were created by calculating the similarity of a job ad with the thirty competence documents and displaying these scores in the chart. Next to the job ad, a CV was also compared to the competence documents and was displayed in the same chart as the job ad. The axes of the spider chart contains the thirty competence scores, while the values that are plotted on the graph are the document similarity scores. An example can be seen in figure 1.

Unfortunately, the implementation of Vural is not very practical in a real-life situation where a data scientist’s competence must be assessed. The implementation made use of the Apache Spark framework, but this framework is meant for big data processing and would be impractical to use in a situation where one CV is uploaded to be assessed, next to that, there was no option to upload a CV document real-time.

By practically assessing the tool that was developed by Vural, a number of issues were revealed that prevented the tool from being used practically. One problem that appears when Vural’s implementation is to be used as a tool for the assessment of a data scientist, is that the competence scores for the CV are calculated by comparing the similarity between two documents. This means that a score of 1 will never appear, because a CV will never be exactly the same as a document explaining a competence. If the visualisation of the competence scores for the CV are shown on a 0 to 1 scale, it will give a
non-realistic portrayal of how competent this person is.

Another problem that occurs is that in a visualisation of document similarity, some important data may not be taken into account without using additional techniques. For example, a CV containing the information that someone has been practising a profession for a decade or longer yields almost the same score as someone’s CV that has the exact same text and structure, with the only difference of having practised that profession for one year instead of a decade. This score is obviously correct for a document similarity measure, but not when a CV should be matched with competence documents to provide insights into how competent a data scientist is. It can be assumed that someone who has practised a profession for a longer time is also more competent at the relevant skills that are required for this profession.

One of the aims of this thesis will be to solve the different problems that Vural’s tool encounters, by using more than a document similarity measure to calculate competence scores of a CV. Instead, the relevant contents of a CV will be extracted, to provide another overview based on the career path described in the CV.

Figure 1: Example of job and CV matching with the EDISON competencies by Vural
3 Theoretical foundation

This section will discuss a theoretical background for relevant methods that have been used in this study: document similarity calculations and regular expressions. These two methods play a major role in how the results of this study were acquired.

Both Koulouzis (section 2.2) and Vural (section 2.3) have used document similarity techniques in a straightforward manner to match CVs with the competence framework of the EDISON project. The presented study has extended the generic methods by additionally using regular expressions to extract relevant terms from a CV which were used to calculate competence scores in a different manner.

3.1 Document similarity

Document similarity is a measure that quantifies how similar two documents with text are. In the scope of this study, the documents that will be compared are written in human language and are the competence documents from the dataset of previous works that will be compared with an uploaded CV. There is no single definition for a similarity measure to achieve this, but two techniques called term frequency-inverse document frequency (TF-IDF) and cosine similarity will be used to make this possible.

3.1.1 TF-IDF measure for featurization

To calculate the similarity between two documents, it is necessary to transform a document into a vector of numbers. Computers cannot calculate with words, they can only operate on numbers, which means that a numeric representation of a document must be made before a computer can perform mathematical operations on this ‘document’.

The process of extracting vectors containing numbers from documents is called featurization. Each vector is called a feature vector. Because the vectors only contain numbers and should represent a document they can be used as input for different algorithms, while the output of these algorithms can then be retraced to the document itself.

One important aspect about using feature vectors of a document to operate on and reach conclusions about the document itself, is how well the features represent the document. If completely arbitrary features are extracted, the results might not lead to valid conclusions about the document itself.

Documents contain terms that have a statistical quality, which can be used when classifying a document in which the terms occur. This study uses TF-IDF as a method for collecting features of a document using these terms. 83% of text-based recommender systems of digital libraries use TF-IDF.

TF-IDF is used for weighing terms based on how important a term is for a document in a collection of documents. The weight of a term describes how important a term is. The values are acquired by multiplying the term value by the inverse document frequency of the term.
frequency (TF) and the inverse document frequency (IDF), hence the name, ‘term frequency–inverse document frequency’.

The term frequency can be calculated by counting how many times a term appears in a document: $TF(t, d)$. This is a key step, because it can be assumed that words that appear often in a document can be deemed important for this document.

Unfortunately, terms that appear often in a document are not always a possible representative of this document. Some common words that occur often in a document may actually hold little meaning; a document often contains stop words like: 'the', 'has', and 'a'. These words would receive large weights because they appear often, but if they would be used as features, they would not represent a document at all. This does not necessarily only happen with stop words; terms like 'data' or 'science' could also often appear in documents about data science, without actually representing the contents of the document. This is especially the case if a distinction should be made between multiple documents that are all about data science.

This means that something that decreases the importance or weight of a term is needed, if this term appears often in every document in the collection. This is what the inverse document frequency measure is for. It shows how much information a word provides. It can be assumed that a term that does not appear often in a collection of documents but appears often in a single document, holds important information about the document it appears in.

The inverse document frequency of a term can be calculated by using the following formula:

$$IDF(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (1)$$

where $|D|$ is the amount of documents in a collection or corpus and $|\{d \in D : t \in d\}|$ is the number of documents in the corpus that contain the term. To avoid dividing by zero, the amount of documents that contain the term are increased by one if there are no documents containing this term. This formula can be understood intuitively by looking at what happens when a term occurs in all of the documents. If a term appears in all of the documents, $|D|$ will be divided by $|D|$, which will result in $log(1)$ which is equal to 0, meaning the term will not be considered important.

Finally, term frequency-inverse document frequency can be calculated by taking the term frequency $TF(t, d)$ and inverse document frequency $IDF(t, D)$, which will result in:

$$TF - IDF(t, d, D) = TF(t, d) \cdot IDF(t, D) \quad (2)$$

which will provide high weights for terms that have a high term frequency in a given document and have a low document frequency. This will filter out the terms that are common across all documents in the collection, while rewarding terms that occur often in a single given document from the collection.
3.1.2 Cosine similarity metric

TF-IDF can be used to extract features from a document as explained in section 3.1.1. These features, which are vectors that represent a document, can then be used to calculate the similarity between two documents. To do this, a similarity metric called cosine similarity can be used. Research has shown that the cosine similarity metric is a solid choice when TF-IDF has been used to extract features. The similarity between two documents can be calculated by measuring the cosine of the angle between two vectors that represent a document. This relies on the assumption that documents that are similar to each other, will also have small divergence between their vectors. The cosine similarity for two vectors: A and B with an angle $\theta$ between them, can be calculated with the following formula:

$$
cosine\ similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{||A||_2||B||_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3)
$$

3.2 Regular expressions

Regular expressions\(^4\), also known as regex or regexp, are sequences of characters that define a search pattern. These searching patterns can then be used by algorithms to find, replace, copy or perform other operations upon the text that is found. These regular expressions are widely used by search engines, text editors or other text analysis software.

Regexes\(^5\) contain characters that either signify a character that should literally be matched, or a metacharacter\(^6\) that says something about the pattern. Combining these literal characters and metacharacters, regular expressions possess a very expressive power to specify a wide range of patterns.

Regular expressions can be used to identify text of a given pattern, or process a number of occurrences of the pattern. This can be specified by the metacharacters, they allow for specification of how precise the text must be matched with the pattern. Metacharacters can denote small ranges from letters a–z, or huge ranges, for example every character. This makes it possible to find very specific things like job names in a text, or the location of date ranges, e.g: 05/2017–09/2018. Some basic regular expression concepts are as follows:

- **Alternative matching**

  Regular expressions often allow for a specification of alternative occurrences. This is done by using a Boolean ‘or’, which is represented by the ‘\(|\)$’ symbol. For example, the regular expression: ’data—date’, matches the word ’data’ or matches the word ’date’.

\(^4\)Regular expressions

\(^5\)A character that has special meaning to a computer program
• Grouping
Regular expressions often contain groups. These groups indicate scopes and precedence of operators. Parentheses are used to indicate a group, for example, the following regex: \texttt{dat(a|e)}, also matches the words ’date’ and ’data’. The difference with the regex from the previous point, is that the Boolean or operator is limited to the final two letters, which will result in the same words being matched because the only difference between the word ’date’ and ’data’ is the final letter.

• Quantification indication
Another commonly used concept in regular expressions are the indications of multiple occurrences of a pattern. This is done by using specific quantifiers after a character or group to indicate how many times this group or character should occur in the pattern. The most common quantifiers are the ’?’ and ’*’ characters. They respectively indicate that ’zero or one’ occurrences must be matched and ’zero or more’. To exactly match a preceding item a specific number of occurrences, a number can be given between the { and } symbols.
4 Design

This thesis aims to match a CV with the CF-DS to accurately represent the capabilities of a data scientist. The aim was also to make this as accessible and easy to use as possible. To solve these problems, a tool with two implementations was created: a document similarity implementation similar to what Koulouzis and Vural did as described respectively in section 2.2 and 2.3, and an implementation that extracts information about a career path from a CV, which is then also used to calculate and visualise competence scores. These implementations are then both used by a visualisation tool that displays the results. This section will provide the design of the proposed solution which is supported by the main use cases analysis and requirements. It will also explain the architectures that have been used to realise this.

4.1 Use case analysis and requirements

By understanding the different aims that someone might have for using a tool, it becomes possible to specify the exact requirements and specifications that are needed to solve a problem correctly and completely. To provide a complete solution that allows data scientists and recruiters to assess someone based on EDISON’s CF-DS after a CV was uploaded, it is necessary to fully consider the use cases and requirements that they might have. The following use cases would be relevant for the scope of this thesis:

1. Gap analysis
   A data scientist might be interested in assessing himself by being able to take note of his current competency levels. The CF-DS defined by EDISON is perfect for data scientists to accurately evaluate themselves and decide on a direction that serves their personal ambition.

2. Assessment for a job vacancy
   To be able to match a CV with the competencies defined in the CF-DS would be very desirable by a recruiter; this would allow them to accurately select the most competent and fitting data scientist for a given job vacancy. A recruiter might also be interested in the career path of a data scientist, because this can be used next to the overview of the competencies.

3. Team assessment
   When a team of data scientists is working together, it might be helpful to assess the total strengths and weaknesses of the team. If the CVs of the data scientists are used to generate graphs that show how competent the team is, it becomes possible to analyse what tasks are most fitting for the team, or what kind of training the team needs.

   By listing the use cases, it becomes possible to create the requirements or specifications. The requirements serve to specify what should be possible to enable the best experience for the user and to create the most optimal solution
for the research questions. By looking at the use cases and research problem, the following requirements should be satisfied:

1. Contains accurate representations showing the scores for the CF-DS competencies of a data scientist
2. Contains an overview of the career path of a data scientist

4.2 Functional design and architecture

To satisfy the requirements specified in section 4.1, two components that form the solution were created. This section describes the functional design and architecture of both components. One component is a visualisation tool, or front end, that handles the user interaction and displays the results in an understandable manner. The second component is the back end of the solution, that handles the creation of the results by operating on the data that is provided by the front end.

4.2.1 Visualisation tool design

The visualisation tool was developed to easily evaluate the results of the back end implementation. It is also the final product produced by this thesis that can be used by recruiters or data scientists for assessment purposes, which means that it should be easy to use and serve results in an understandable manner. To achieve this, the visualisation tool was designed to have multiple components:

- Upload component
  This component is the initial point where a user makes contact with the tool, it was designed to be effective and minimalist. The only function that it has is to allow the user to easily upload a CV and then direct them to relevant information when this has been retrieved. The design for this component is shown in figure 2.

- Extracted text and uploaded document visualisation
  Another component of the tool shows the document that was uploaded by the user to easily allow for visual feedback and avoid making mistakes by having someone upload the wrong document. Next to the uploaded document, the extracted text from the document is shown to portray what text was used exactly for retrieving the results. The design for this component is shown in figure 3.

  On the left, the uploaded document can be seen in an iframe\(^6\), while on the right the text that was extracted from the uploaded document is shown. This is important because not all documents allow all of their text to be extracted, this component makes it possible to understand if the results of other components are feasible.

\(^6\) a nested browsing context that can be used to view PDF documents
• Data scientist career path visualisation

A visualisation of the career path of a data scientist can be valuable for a data scientist or recruiter, especially when the different elements of the career are also provided together with relevant information describing how the other results of the visualisation tool were collected.

In figure 4 the design is shown for a timeline that visualises the career path of a data scientist. It shows the different career elements in an alternating manner in chronological order, where the most recent job is located on top. The profession name is clickable, indicated by the grey background, while the other side shows the span of time that indicates how long the person has practised this profession. If someone clicks on the grey background containing a job name, it expands and should list relevant information for this job of the person. The career path is also visualised in the form of competence relevance scores, the design for this will be explained in the next point.

• Graphs showing how competent a data scientist is, based on the CF-DS.

The final component is the essence of the whole tool, it shows graphs that visualise the scores of someone’s competencies using the CF-DS. The axes contain the thirty different competencies described by the CF-DS. The scores are indicated by the grey dots on the black lines, while the idea behind this chart is that the bigger the area that is covered, the more competent a data scientist is. The design for the chart used in this thesis is shown in figure 5. Inspiration for this chart was gained from the work Vural did, which is explained in section 2.3.

Figure 2: Design of first visualisation tool screen
Figure 3: Design of extracted text and uploaded document visualisation

Figure 4: Design of how a career path can be visualised
4.2.2 Visualisation tool architecture and user interface

To create the visualisation tool that fits the requirements and design, the decision was made to create a web application using the Angular framework\textsuperscript{15}. The reason the front end was created in the form of a web application, is because this enables the creation of specific user interfaces and detailed visual components, without much effort. Creating an easily accessible, usable and good looking design is much harder in other environments that do not depend on HTML or CSS\textsuperscript{22}. To create this web application, the Angular framework was chosen for multiple reasons.

The strongest reason for using the Angular framework, is that it allows to dynamically update the interface with new elements, without having to reload the page. This provides an environment where results can quickly be evaluated without unnecessary waiting. Also, the Angular framework is very modular, which makes it easy for other researchers to continue or improve on this project with relatively little amount of time needed to get accustomed to the implementation.

The architecture of the web application, which serves as the front end, is shown in figure\textsuperscript{6}. The modularity of the application is clearly shown in this figure; all the components have their own files. For every component, three files are used: an HTML file, a CSS file and a TypeScript (TS) file. The HTML

Figure 5: Design of a graph showing a CV matched with all CF-DS competencies
4 DESIGN

describes which elements a page contains and how they are ordered. The CSS file contains the rules for the design of these elements, e.g.: the background colour of the component. The TypeScript file contains all the logic of the component, which handles all the user interaction. The Angular framework can be seen as the wires that connect all the components and files. Finally, the data that is visualised by this application must be retrieved from somewhere, which is the task of the back end.

Figure 6: Architecture of Angular web application (front end)

4.2.3 Back end architecture

To be able to actually extract data from an uploaded CV and to operate on this data to collect the results, a back end component is needed. This component must be able to collect the CV data, transform the data of the CV into desired data and must then be able to serve this data to the front end designed in section 4.2.1. This means the back end should be able to do the following:

1. Receive CV data from front end
2. Transform CV data into desired data
3. Serve the new data to the front end

All the components of the back end were written in the Python programming language[23], this decision was made because Python has considerable support for mathematical operations and has an extensive list of external libraries that can be used for different features. Next to that, Python makes it very easy to construct strong coding statements in relatively few lines of code.
For point 1 and 3 mentioned above, an additional component is needed. To receive and serve data to the front end, a microframework called Flask was used. Flask is a framework that can be fully programmed in Python, it was used because it does not require particular tools or libraries. It has no database abstraction layer, or other components that require additional setup. This way it can be used effectively and efficiently without the necessity to consider unnecessary configuration. A representation state transfer (REST) design was used for the back end architecture, by combining Flask and the Python programming language. This allows the back end to function as an application programming interface (API), which makes it easy to implement the communication between the front end and back end.

In figure 7 the architecture for the back end is shown. The Flask framework is used to call upon the desired main components whenever the front end communicates this. The functionality of the back end that is mentioned in point 2 was split into two main components: The document similarity component that, as the name indicates, calculates the document similarity between the CV and the CF-DS competencies described in the reference dataset that was used from previous works. The other component is the career path component that extracts data from a CV and transforms this into relevant data like the career path timeline or competence relevance scores of the data scientist that has uploaded the CV. The other logic that was not as complex, can be simply handled in the main component that also contains the Flask code. When the back end and front end design and architecture are combined, the implementations to satisfy all the requirements listed in 4.1 should become possible. The next section will discuss the actual implementations that have been used to transform and collect the relevant results.
Figure 7: Architecture of the back end
5 Development

This section will describe the development process that took place to create the results. The logic and implementation of the two main components of the back end described in section 4.2.3 will be explained in multiple steps. First, the dataset that was gathered will be explained, along with the preparation process that the data went through to become usable. After that, the core solutions for the implementations will be discussed, together with the relevant assumptions and decisions that were made to realise the implementation.

5.1 Data gathering and preparation

This section will explain an important step of the development process; the gathering and preparation of the data that was used as a foundation to collect the results. Most initial datasets still require processing before they are usable for the research. The following parts will describe how the data that was used is collected and how the data is processed for both the document similarity component and the career path extraction component.

5.1.1 Competence document similarity

To create a graph showing a data scientist’s competence scores based on the CF-DS using document similarity techniques, two things are needed:

1. The text of the CV document
2. The competence documents

The data used for point 1 was relatively simple to collect. indeed.com was used to collect initial data scientist CVs, next to other sources that also offered CVs. The full text was extracted from these documents and was stored as strings 7 for each different CV. Eventually, a CV will be provided by the front end which will need to be processed.

To retrieve the best results in regards to document similarity, the text that is being compared should go through several stages. The pre-processing of the documents, is necessary to ensure that the document similarity scores will be as accurate as possible. For example, a computer cannot understand that a word with and without a capital letter are in essence the same; this means that a CV document that mentions important words in capital letters, will not be deemed similar as the same document that uses the same words without any capital letters. For example, a computer sees the terms ’Data Scientist’ and ’data scientist’ as two different things. This means that it is necessary to transform the CV and the competence documents into texts that use the same standards so they can be compared.

The process of changing the text of a document to a form in which its meaning can be compared to other documents that have gone through the same steps, is called normalisation. To achieve this, the CV and competency 7Sequences of characters
documents go through the steps as shown in figure 8. First all the punctuation is removed from the text, after this all the letters are transformed to lower case. Finally, the Porter algorithm is used to stem all the words. This is the process of reducing inflected or derived words to their word stem; this will help to compare words that have the same meaning, but appear in different forms.

The other dataset, the competency documents mentioned in point 2, are also required for the document similarity implementation. As mentioned before, the competency dataset was inherited from Vural’s previous work. Vural’s work included a lot of unnecessary data for the scope of this study, hence only the competency data was used. The data was structured in a JavaScript object notation (JSON) in a line format, where each line contains one competency document. The structure of the JSON is shown in figure 9. This structure makes it easy to access the name or the text of a competency, without losing track of the structure.

Figure 8: Processing steps of CV and competency documents text

```json
{
   "competencyName": "DSRM06",
   "competencyText": "Text describing competency DSRM06"
},
{
   "competencyName": "DSRM07",
   "competencyText": "Text describing competency DSRM07"
}
```

Figure 9: JSON format of competency documents

5.1.2 Career path extraction

To extract the career path from a CV and visualise this information in the form of CF-DS competence scores, the following data was prepared:

1. Job names and DSPP classification
2. DSPP competence relevance scores
3. CF-DS competence label list
The job names were collected by collecting lists of possible jobs in an excel file, and were then manually classified into one of the Data Scientist Professional Profiles listed in appendix 10.2. The list has 556 different jobs, where 161 of those jobs were associated with a DSPP. The other jobs were less relevant, non-related data scientist professions. The excel file was loaded into Python and transformed into a dictionary\(^8\) where the job names served as the keys and the DSPP classification as the value.

For point 2 and 3 the excel sheet shown in appendix 10.3 was loaded in, and then also transformed into a dictionary. First a list was created from all the competence labels and secondly all the DSPP profiles were used as keys, and a list of their respective competence relevance scores were used as the values. For all the values that are missing, the letters 'NA' were used to indicate that there was no DSPP classification or competence relevance score. The relations between the data can be seen in figure 10. As can be seen from the figure, jobs in the list of jobs may have a classification to a DSPP. Every DSPP has a list of competence relevance scores for the competencies defined in the CF-DS.

---

\(^8\)A data structure that is used for storing objects. A dictionary has a set of keys and each key has a single associated value. The associated values can be looked up by using the corresponding key.
5.2 Implementation

The following section explains the logic for the main components that produce the final results. An in-depth overview will be given of which techniques were used and which decisions were made to produce the final outcome.

5.2.1 Competence document similarity

One important part of the final solution is the graph that shows someone’s competencies based on the CF-DS. This implementation uses the processed data described in section 5.1.1: the pre-processed CV text and pre-processed competencies text. Formula (4) was used to calculate all the document similarity scores between the CV and competency documents, it is used for each competency document in the list, which means it is executed thirty times in total.

\[ \text{Similarity score}_i \leftarrow \text{sim} (\text{CV}, \text{competency}_i) \]  
\[ (4) \]

In formula (4) the i stands for the current competency document and the sim() function calculates the similarity between the CV document and the competence at index i. This function was implemented by using the functionalities of the scikit-learn library. Their implementation of gathering TF-IDF features was used on all the text documents. What their implementation does in essence, was explained in section 3.1.1. When their implementation is fed the two pre-processed texts, a matrix of TF-IDF features is returned, which can then be used to calculate the cosine similarity. The explanation of using the cosine similarity as a metric to calculate the similarity scores was described in section 3.1.2.

The process of calculating the cosine similarities from the TF-IDF matrix is straightforward. By taking the dot product of the vectors, which is possible because the vectors are already row-normalised, a similarity matrix is acquired. For the comparison of two texts, the similarity matrix is in the form of a 2x2 matrix. By looking at the second value, the cosine similarity, and thus the similarity score, can be retrieved.

Because these cosine similarities indicate the amount of similarity between the CV and competency documents, a score of 1 should never be reached. Someone’s CV will never be exactly the same as a competence document. If the visualisation of the competence scores for the CV are shown on a 0 to 1 scale, it will give a non-realistic portrayal of how competent this person is. To solve this, a score of 0.7 is seen as the maximum score that can be practically assessed at this stage, which means that when a score is 0.7 or higher, it is automatically scaled up to one. The final step is to store all these values; this was done by creating a dictionary that associates the competence similarity scores as values with the labels, e.g: DSDA01, as keys.
5.2.2 Career path extraction

This part will explain how the career path of a CV was extracted and how this information was used to create a graph showing competence relevance scores, along with a timeline. The process of extracting information from a CV that can be used to construct a graph showing competence relevance scores and a career path timeline has multiple steps. The first thing that was done was look for jobs in the CV, then for each job that was found, the following steps were executed:

1. Get the position level for the job
2. Get the amount of experience listed for the job
3. Classify job in a Data Scientist Professional Profile
4. Assign competence relevance scores of DSPP for the job
5. Tag if the job is relevant

The first step is to find all mentions of known positions from the CV. For this step, the list that was prepared from point 1 in section 5.1.2 was used. One by one, each job on the list was used as a regular expression to look up the location of the job in the CV. If the regular expression came up empty, the next job on the list was used as a regular expression. If the regular expression gave results for the job, the steps above are executed for each result. A result here is a start and end position that indicate the location of the mention of the job that was found. Every time a regular expression has multiple results, the pattern was found multiple times in the text. This means a job name can be encountered multiple times in the text of the CV. To deal with this, every time a job with the same name is encountered, the job gets an underscore with a number indicating how many times this job was already encountered. This makes identification possible between multiple mentions of the same job. For each job, a dictionary is created to keep track of all the values that are acquired by following the steps mentioned above. The first value stored is an index value that indicates the order of where the job names were found in the text.

After retrieving all the jobs on the CV that are also on the job list, for each job, the position level is extracted. With position level, the level of the person practising the profession is meant. There are five different profession levels that are often countered, table 1 shows which levels were used as patterns to look for. Different companies use different levels for different tasks, or use different names altogether, hence the table lists multiple names for the same level. These values were acquired after looking through multiple CVs.

For every job, the location of where the job was encountered in the text of the CV was stored. This location was then used to look for the job level near the location of the where the job was mentioned. 100 letters before and after the mention of the job were used as an area to look for a position level match. This choice was made because a position level is often mentioned near the job
name in a CV. The number 100 was chosen because this way there is only a low probability of looking at different unrelated sections, while still having a relatively broad area to search. For each position level name, all the possible names for this level were used as patterns to be found. When a possible name was found near the job mention, it was stored in the dictionary of the job as the position level name shown in the first column. When a position level was encountered for a job mention, the searching process was stopped. If no job levels were encountered for the job, the position level was simply stored as 'NA'.

<table>
<thead>
<tr>
<th>Position level name</th>
<th>Possible name 1</th>
<th>Possible name 2</th>
<th>Possible name 3</th>
<th>Possible name 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>Intern</td>
<td>Junior</td>
<td>Entry</td>
<td>Staff role</td>
</tr>
<tr>
<td>Intermediate</td>
<td>Intermediate</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Senior</td>
<td>Senior</td>
<td>Sr</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Principal</td>
<td>Principal</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lead</td>
<td>Lead</td>
<td>Chief</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Position level names and alternative names

After all the jobs were associated with their possible position levels, the next step is to get the amount of experience that someone has for a job. Experience here means the time that someone has spent practising this job. This was achieved by using regular expressions to look for date patterns. This is a tricky step, because CVs have no clear structure. This means that date ranges can be given in multiple forms, while still meaning the same thing. For example, someone might indicate he has practised his job from 2015 – 2018, while another person might use 01/01/2015 – 01/01/2018. Next to that, some people might use a '−' to indicate a range, while other people might use the word ‘to’. The following problems were taken into consideration:

- Different Indicators for a span, for example: '−' or 'to'
- Different Indications for present year: 'Present', 'Current', '2018'
- Different Abbreviations: 'Jan' vs January
- Different separators: '01-2018' vs '01/2018'

These problems are very important to consider, because if some forms are not taken into account when creating the regular expressions that are used to find the date patterns, it is very likely that the experience for a job will not be found. To split up and deal with this problem, multiple small regular expressions that can deal with different formats were created. The following regular expressions were created to deal with the problems above:

- A regular expression for matching separators: '/' and '−':
  \[(separator) = (/ | |-)\] 

\(^9\)The | symbol means 'or'
• A month regular expression that matches full months or abbreviations of months:

\[(month) = (jan)\mid (feb)\mid (mar)\mid (apr)\mid (may)\mid (jun)\mid (aug)\mid (sep)\mid (oct)\mid (nov)\mid (dec)\mid (january)\mid (february)\mid (march)\mid (april)\mid (may)\mid (june)\mid (july)\mid (august)\mid (september)\mid (october)\mid (november)\mid (december)\mid (01)\mid (02)\mid (03)\mid (04)\mid (05)\mid (06)\mid (07)\mid (08)\mid (09)\mid (10)\mid (11)\mid (12)\]

• A span indicator regular expression that matches: ‘-’, ‘–’, ‘to’:

\[(span\ indicator) = (-)\mid (-)\mid (to)\]

• A regular expression matching a year between 1900 and 2100:

\[(year) = (20\d{2})\mid (\d{2})\]

This regular expression matches a pattern that has ‘19’ or ‘20’ as first two digits and is followed by any two digits. To account for cases just listing ‘14 instead of 2014, any two digits will be matched if the first two digits are not 19 or 20.

• A regular expression matching a day pattern

\[(day) = (0?[1-9])\mid ([12][0-9])\mid (3[01])\]

This regular expression checks for a pattern where a zero is followed by a number between 1 and 9, a pattern where a 1 or 2 is followed by a number between 0 and 9, and finally a pattern where a 3 is followed by a 0 or 1. These three patterns combined can find any day pattern.

Finally, when all regular expressions that can match the different parts of a date indication are combined, the following three formats of date indications can be matched:

1. Full year range (e.g: 2015–2017):
   \[(year)\mid (span\ indicator)\mid (year)\]

2. Month, year range (e.g: jan/2018 to februari/2017):
   \[(month)\mid (separator)\mid (year)\mid (span\ indicator)\mid (month)\mid (separator)\mid (year)\]

3. Full date range (e.g: 05/02/2017 – 06/05/2018):
   \[(day)\mid (separator)\mid (month)\mid (separator)\mid (year)\]
   \[(span\ indicator)\mid (day)\mid (separator)\mid (month)\mid (separator)\mid (year)\]

For every job, a search is done to find any of the date indication patterns above. If a pattern is found, the amount of days from the start date is subtracted from the amount of days from the end date. This number is divided by 365 and rounded down, for simplicity’s sake, to get the number of years someone has practised a profession. This amount is stored in the dictionary of each job, along with the start and end date of the job.

After this, if a job has a DSPP classification, the classification is stored alongside the other values for this job. For every job that has a DSPP classification, the competence relevance scores are also stored in the dictionary. The competence relevance scores can be found in appendix 10.3.
The final step is to tag all the jobs and their dictionaries that can be used for visualisation of a data scientist’s competencies. This part is necessary and important, because it filters out the job names that were matched in a CV, but don’t have enough information to be useful for giving insights into the data scientist’s competencies or career path. For example, job titles or positions occur a lot in a CV, but when a job title is mentioned, it is not always with the intention of listing it as a job that someone lists to show their experience. It might be the case that someone is mentioning the job in a summary, or is talking about what he wants to do at a later stage in his life. A job is deemed relevant if it has a DSPP classification (and thus competence relevance scores) and at least has one of the following two values:

1. Start and end date
2. Profession / job level

At least a DSPP classification and one of the two requirements above is needed for a job occurrence in a CV to be deemed relevant. This is because a DSPP classification is needed for the competence relevance scores, which will be used in the next stage for the visualisation of the data scientist’s competencies based on his career path. Furthermore, the amount of years that someone has practised a profession is also necessary for the next stage, this part can be calculated from having point 1. When a start and end date are not present, a job is still relevant if it has a profession / job level associated to it, because from this the amount of experience in years can be guessed.

When all values are gathered for each job, the final object is a list of jobs with their extracted values. The example structure for two jobs can be found in figure 11. Any values that are missing during the extraction process, become ’NA’, this allows for the structure to be maintained and any missing values to be easily identified.
Figure 11: JSON format of jobs

5.2.3 Front end

The last part of the implementation explains the front end and how the final data from section 5.2.1 and 5.2.2 can be used to visualise the competence scores and a timeline. The easiest component to visualise are the competency scores retrieved by the document similarity implementation. This was done using JavaScript framework for charts, called ChartJS. A spider chart, or radar chart, was created by using an array with the labels of the competence scores and an array with the competence and CV similarity scores. The scores and labels are associated with each other by their orders, e.g: the first label will be associated with the score at the first index. This produces a graph similar to the designed graph shown in figure 5.

To visualise the competence scores based on the career path of a data scientist, an algorithm is needed that can use all the data from the relevant

---

10 A data structure that stores values in a list. The values can be accessed by presenting an index or location of the element
jobs and transform it into thirty competence scores that can be used to create a radar chart. The fields shown in figure 11 will be used to accomplish this.

As mentioned before, the competence relevance scores shown in appendix 10.3 show how relevant a job is for the competencies from the CF-DS. To create the final graph showing the competence scores, two assumptions were made:

- when someone has practised a job for a longer time, he becomes more competent at the relevant competencies that are listed

  The assumption can be made, that a career path is a process of accumulating experience. The competencies that are more relevant for the job of the data scientist, will also be developed more than other competencies that are not related to the job of the data scientist.

- When someone quits a job, the competencies to perform relevant tasks from that job are not lost

  If someone has practised a certain profession for a long amount of time and then quits that job and starts doing a new job, it can be assumed that the skills and competencies that were acquired during the period of the previous job were not lost, or at least can be reacquired easily.

These two assumptions are necessary for the algorithm to transform the data that was extracted from the CV into thirty competence scores. For each competence of the CF-DS, the following formula was used to calculate the final competence score using the career path:

$$\text{Competence}_i \leftarrow \min\left(\sum_{j \in J} c_{ij} \cdot \text{multiplier}_j, 100\right)$$

Where $i$ is the current competence, $j$ is a job, $J$ are all the jobs that were extracted, $c_{ij}$ is the competence relevance score for the current competence and job, and multiplier$_j$ is the multiplier that is used for the job. This formula is used for each competence: it takes the minimum of 100 and the sum of the values calculated using all the relevant jobs for the current competence. The value calculated for each job for a competence, was retrieved by using the competence relevance score that is associated with the current competence and being multiplied by a multiplier. The minimum of the sum and 100 is taken, because the scale that will be used to visualise the values is $0 - 100$. This was done to create an easy to understand visual depiction of someone’s scores, where 100 is perfectly competent and 0 means no competence at all.

The multiplier that is used in formula 5 can be retrieved in two manners:

1. Direct amount of job experience in years acquired by end date of job - start date of job

2. Mapping from position level to experience amount

The most straightforward one is by using the amount of years that someone has practised a job, but if this is not available due to non available date information for the job, it is necessary to infer the amount of time someone has practised a
job, using the profession level. In the step where jobs were marked relevant or not, it was also assured that all relevant jobs have either a value in years for the amount of experience, or a job level indication. To map the position level to inferred amount of experience, table 2 was used. When all the scores for the competencies have been calculated by using all the relevant extracted jobs another graph can be created like figure 5 but this time using a scale from 0 – 100. The final visualisation of the timeline as shown in figure 4, is achieved by simply showing the relevant data from figure 11 in an ordered manner.

<table>
<thead>
<tr>
<th>Position level</th>
<th>Amount of years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>1</td>
</tr>
<tr>
<td>Intermediate</td>
<td>2.5</td>
</tr>
<tr>
<td>Senior</td>
<td>6</td>
</tr>
<tr>
<td>Principal</td>
<td>6</td>
</tr>
<tr>
<td>Lead</td>
<td>8</td>
</tr>
<tr>
<td>NA</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Mapping from position level to experience amount
6 Results

This section will show and analyse the results that were produced after using the implementation described in section 5. To test the final tool that was developed, multiple documents were used to gather results. The following documents were used as a dataset to gather the results that will be shown in this section:

1. CV of a data scientist with 8 years of experience (appendix 10.4)
2. CV of a data scientist with 13 years of experience (appendix 10.5)
3. CV of a data scientist with 15 years of experience (appendix 10.6)
4. CV of a Hadoop developer with 3 years of experience (appendix 10.7)
5. CV of a Hadoop developer with 15 years of experience (appendix 10.8)

The results for the first CV, a data scientist with 8 years experience can be seen in figures 12, 13, and 14. What can be observed is that the competence scores from the career path, figure 13, cover a much larger area than the competence scores showed by the document similarity implementation, figure 12. Next to that, the timeline, figure 14 shows all the jobs that were listed in the CV that can be seen in appendix 10.4.

Figure 12: Document similarity competence scores of a data scientist with 8 years experience
The results for the CV of a data scientist with 13 years of experience can be seen in figures 15, 16, and 17. The CV, that can be seen in appendix 10.5, barely mentions any understandable job or profession names, except for 'data
analyst’. This is reflected in the timeline of the career path seen in figure 17. Furthermore, from both figures 15 and 16 that show the competence scores, it can be observed that almost no area is covered by the competence values.

Figure 15: Document similarity competence scores of a data scientist with no clear job names
Figures 16, 17, and 18 show the results for a CV of a data scientist who has 15 years of experience, that can be seen in appendix [10.6]. What can be observed is that the career path competence scores shown in figure 19 cover a much larger area than the document similarity scores shown in 18. In the timeline shown in 20, two entries of a data scientist can be seen. The CV has three entries of data scientist jobs, but it only mentions a position level for one of them, while none of them have any date indications.
Figure 18: Document similarity competence scores of a data scientist with 15 years experience
The penultimate CV that was used as a test, can be seen in appendix 10.7. The CV mentions two jobs, where one job has an 'intern' level mentioned, which was classified as 'entry' level by the developed tool. Next to that, an irregular date format is used. Figures 21, 22, and 23 show the results for this CV. What can be observed is that both competence graph show relatively little amount of area being covered, but the document similarity graph shown
in figure 21 still has some small peaks. The timeline only shows one job that mentions an 'entry' job level.

Figure 21: Document similarity competence scores of a Hadoop developer with 3 years experience
The final CV that was used can be seen in Figure 10.8. It lists three entries of jobs, where all three have irregular date span formats. Two of the jobs mentioned have a position level. What can be seen from the competence graphs shown in Figure 24 and 25 is that the competence graph based on the career path has almost all the area covered, while the graph computed through document similarity means has only a few small peaks. The timeline shown in Figure 26 shows three job entries, where the first one has a principal level as job level.
Figure 24: Document similarity competence scores of a Hadoop developer with 15 years experience
Figure 25: Career competence scores of a Hadoop developer with 15 years experience

Figure 26: Career timeline of a Hadoop developer with 15 years experience
7 Evaluation

CVs are essentially documents that can appear in multiple formats, while still conveying the same meaning. This makes it difficult to extract desired information from this document, because it is difficult to assess where to look for certain information in a document that has no clear structure.

The aim of this research was to determine the best possible methods of matching a CV document with the competency framework defined by EDISON. Next to that, the aim was to design and create an implementation that can be used by data scientists and recruiters for assessment.

This was done by using regular expressions to extract relevant information from a CV that indicate the career timeline of a data scientist. This data was then used to visualise a timeline and competency graph. Next to that, document similarity techniques were used to produce another competency graph showing how similar a given CV is with the different competencies defined by EDISON. The following sections will analyse the results from the previous section in-depth to answer the research questions. Because there is no dataset that contains CVs that already have competence scores labelled to them manually, all the assessments of the results have to be done empirically.

7.1 Document similarity

Document similarity techniques were used to collect thirty competence scores for different CVs. An implementation was created that allows for the upload of one’s CV to visualise the competence scores in a radar chart. This was also the approach taken by both Koulouzis and Vural, who are mentioned in section 2.2 and 2.3.

Using document similarity as a basis to visualise a data scientist’s competencies has certain disadvantages. From the results it can be seen that the document similarity scores are always on the low side. It seems that the maximum value that can be achieved at this stage is around 0.5, which makes sense, because CVs can have a lot of different formats and it might be that most CVs lack a repeat of keywords before the similarity scores would go up. It might be helpful to visualise the graphs on a basis from 0 – 0.5 instead of 0 to the maximum score of 1, this would allow for a more practical assessment of someone’s competencies.

For a human, an implicit assumption is made that when someone lists he or she has practised a profession for a long time, he has also become more competent. A computer can not infer this automatically, hence for the document similarity scores to portray this information, it is necessary that the relevant keywords are repeated a lot of times.

This also causes that document similarity techniques can not be used well to infer implied knowledge from relative small amount of keywords. For example, someone might mention he has 1 year amount of experience, while someone with almost the exact same CV has a small difference in text that notes 10 years of experience. These documents would get almost the same competence graph
using document similarity as a metric, because the difference between one word makes very little difference. This is caused because it has no contextual meaning for the computer. This can clearly be seen by comparing figure 21 and 24. While one figure is based on the CV of someone with three years of experience and the other figure is based on someone’s CV that mentions 15 years of experience, both the graphs look almost identical. Using this document similarity implementation on its own as a basis for visualising a data scientist’s competencies is best avoided, because too much information from the CV is not taken into consideration.

7.2 Career path

Regular expressions were used to extract information about the career path in a CV. This data was used afterwards to create a timeline that visualised the career path, it was also used to create a visualisation of the competencies of a data scientist, by using the CF-DS and DSPP frameworks from EDISON. Several important assumptions were made to realise this:

1. Relevant information that describes a job can be found near the mention of the job
2. The experience gained during someone’s career is cumulative, meaning acquired competencies will not drop after an amount of time.
3. The competency relevance scores accurately portray which competencies someone with the associated DSPP will acquire.

By relying on the assumptions above, several graphs could be created that seem to accurately portray the competencies of a data scientist. By relying on this implementation, it also becomes possible to differentiate between CVs that have the same structure, but different amounts of experience. This can be inferred from comparing figure 13 and 19. The CV of the data scientist with 8 years of experience is visualised as a competence graph with an area that is relatively covered, but the CV of the data scientist with 15 years of experience is maxed out at some competencies, which is quite believable after 15 years of experience.

Unfortunately, this implementation has one obvious disadvantage, which is that it is completely reliant on data that is extracted from the CV. To gather relevant data, a lot of patterns have to be taken into account, because CVs can appear in all kinds of different forms. Next to this, a lot of data has to be prepared that can be used to look for job names or position levels. For example, the results shown in figure 15 and 16 indicate that the data scientist with 13 years of experience is not competent at all. This is because next to no relevant information could be extracted from the CV; all the job names in the CV are more task descriptions than clear functions. This makes it very difficult to extract, which results in barely any information that can be used to create a career path for the computer.
Another important point to consider, is the third assumption that was made. The eventual scores for each competence that is shown in the figures for the career path, are dependent on: the amount of relevant jobs, the competence relevance scores that the jobs have for the competence, and the multipliers that are used to indicate an amount of experience. The core of this calculation are the competence relevance scores from the DSPP and CF-DS frameworks. These values are predefined, which means it might be the case that people who do not have high scores for certain competencies, still show up as very competent even while this is not the case in real life. This is because the competence relevance values are not defined in accordance with the description of a job that someone has listed.
8 Discussion and future work

This research has shown that by using document similarity techniques and regular expressions, depictions of a data scientist’s competencies can be given that are likely to be close to reality. A difficult part of this research is that the assessment of the results has been done empirically. This means that the validation process can very easily have some small mistakes that have slipped in unnoticed.

Next to that, to get a working implementation, a lot of assumptions have to be made in order to get the most accurate results. This is essentially no problem, but if one of the assumptions turn out not to hold, it becomes a huge problem for the other components of the implementation.

To improve the research and final solution, several more steps can be taken. This project has delivered working code that is public and can be improved upon. The front end component and back end component were both well documented and are free to view on the EDSF community development project page (appendix [10.9]). To improve this research one step that can be taken for example is that the competence relevance scores that a DSPP lists, could be partly based on the description of a job that was classified into the DSPP. Weights for the competence relevance scores can be acquired, to further use the context of a CV and avoid over qualifying a data scientist.

Another approach that can be taken to optimise the solution of the document similarity implementation, is to use the information of the CV to apply weights to different features. This can be done by looking for certain education requirements in the CV and then using this to weigh these terms more, creating higher similarity scores with the competence documents that are closely related to the education.

Finally, if an expert would be able to assess a lot of CVs manually and produce a test set containing a lot of samples of CVs and competence scores, then machine learning could be applied to learn to weigh the different features in the document similarity implementation, or it could be used to find relevant patterns that can be used to extract relevant text more easily from a CV.
9 Conclusion

To be able to assess a data scientist is much desired. The EDISON Data Scientist Framework makes this a possibility because all different layers of the data science field and profession have been clearly specified. Unfortunately, to assess a data scientist correctly using the EDISON Competence Framework is a convoluted task. This is because scores for the different competencies can not simply be extracted from a person.

As a representative of a data scientist’s skills and experience, the CV document was chosen. This entity seems the closest to being a most descriptive document about the creator’s career path, knowledge and experience. Unfortunately, matching a CV with the competence framework is not easy. To do this, document similarity techniques along with regular expressions were used to create insights into the competencies of a data scientist. Looking at the results and the evaluation, it can be inferred that the best method to get an accurate overview of a data scientist’s competencies at this point in time, is to use all three components of the tool that was developed. Every part of the implementation supports each other, giving insights to the user.

The visualisation of the competence scores based on the document similarity implementation, shows which terms the creator of the CV uses the most and which competencies he is likely to have. The career path visualisation of the competencies graph can help with showing the differences in experience and the competencies that a data scientist is likely to have with a career path. The document similarity implementation is unable to do this, but the graph created from the document similarity implementation does show which competency scores likely to be too high or too low. If certain terms are often used in CV, which are projected as high scores for certain competencies, but these competencies are not relevant for the jobs of the career path that the data scientist has, then it can be assumed that the data scientist is more competent than indicated by his job profiles. Next to this, if the relevant information of the CV cannot be completely extracted, the document similarity graph is the closest one can get to a visualisation of competency scores. Finally, the timeline yields an easy to understand overview that visualises what data was used to create the competency graph based on the career path. If very low scores are shown for both graphs, but the timeline shows only one entry of a job while the CV lists multiple jobs, then this result is likely due to an incomplete parsing of the CV, than that it is likely for the data scientist to be incompetent.

This means that by combining all the different elements of the research, a close to accurate representation of a data scientist's competencies can be visualised, given that the information of the CV can be extracted to form a complete career path of the data scientist. This makes it possible for data scientists and recruiters to assess CVs for their own purposes. Next to that, they can infer if the scores are likely to be indicated correctly due to the extra provision of information, for example the timeline showing the used career path.
The presented research has answered the research questions with the proposed solution that improves existing and prior solutions by using new approaches in regards to CV evaluation and a combination of multiple components. The benefits of using a solution that incorporates both document similarity and the context inside a CV are apparent. Both components support each other and make up for the weaknesses of the other component.

To conclude, this research has established that a combination of regular expressions and document similarity techniques can yield sufficient representations of a data scientist’s competencies in most cases. There are still a lot of improvements that can be made, but at this stage, it has become possible for data scientists and recruiters to start a basic assessment process, which will in turn enable data scientists and even the data science community itself to develop.

References


[23] Python programming language. [https://www.python.org/](https://www.python.org/)


[31] ChartJS. [https://www.chartjs.org/](https://www.chartjs.org/)
## 10 Appendix

### 10.1 Data Science Competence Framework (CF-DS)

<table>
<thead>
<tr>
<th>Data Analytics (DSDA)</th>
<th>Data Science Engineering (DSENG)</th>
<th>Data Management (DSDM)</th>
<th>Research Methods and Project Management (DSRM)</th>
<th>Domain related Competences (DSDK): Applied to Business Analytics (DSBA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSDA</strong></td>
<td>Use appropriate data analytics and statistical techniques on available data to discover new relations and deliver insights into research problem or organizational processes and support decision-making.</td>
<td><strong>DSENG</strong></td>
<td>Use engineering principles and modern computer technologies to research, design, implement new data analytics applications; develop experiments, processes, instruments, systems, infrastructures to support data handling during the whole data lifecycle.</td>
<td><strong>DSDM</strong></td>
</tr>
<tr>
<td><strong>DSDA01</strong></td>
<td>Effectively use variety of data analytics techniques, such as Machine Learning (including supervised, unsupervised, semi-supervised learning), Data Mining, Prescriptive and Predictive Analytics, for complex data analysis through the whole data lifecycle</td>
<td><strong>DSENG01</strong></td>
<td>Use engineering principles (general and software) to research, design, develop and implement new instruments and applications for data collection, storage, analysis and visualisation</td>
<td></td>
</tr>
<tr>
<td><strong>DSDA02</strong></td>
<td>Apply designated quantitative techniques, including statistics, time series analysis, optimization, and simulation to deploy appropriate models for analysis and prediction</td>
<td><strong>DSENG02</strong></td>
<td>Develop and apply computational and data driven solutions to domain related problems using wide range of data analytics platforms, with the special focus on Big Data technologies for large datasets and cloud based data analytics platforms</td>
<td><strong>DSDM01</strong></td>
</tr>
<tr>
<td><strong>DSDA03</strong></td>
<td>Identify, extract, and pull together available and pertinent heterogeneous data, including modern data sources such as social media data, open data, governmental data</td>
<td><strong>DSENG03</strong></td>
<td>Develop and prototype specialised data analysis applications, tools and supporting infrastructures for data driven scientific, business or organisational workflow; use distributed, parallel, batch and streaming processing platforms, including online and cloud based solutions for on-demand provisioned and scalable services</td>
<td><strong>DSDM03</strong></td>
</tr>
<tr>
<td><strong>DSDA04</strong></td>
<td><strong>DSENG04</strong></td>
<td>Develop and implement new data analytics applications; develop experiments, processes, instruments, systems, infrastructures to support data handling during the whole data lifecycle</td>
<td><strong>DSDM04</strong></td>
<td>Develop and implement data management strategy, in particular, in a form of data management policy and Data Management Plan (DMP)</td>
</tr>
<tr>
<td>DSBA04</td>
<td>Understand and use different performance and accuracy metrics for model validation in analytics projects, hypothesis testing, and information retrieval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSEM04</td>
<td>Develop, deploy and operate large scale data storage and processing solutions using different distributed and cloud based platforms for storing data (e.g. Data Lakes, Hadoop, HBase, Cassandra, MongoDB, Accumulo, DynamoDB, others)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSBM04</td>
<td>Maintain historical information on data handling, including reference to published data and corresponding data sources (data provenance)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSRM04</td>
<td>Undertake creative work making systematic use of investigation or experimentation, to discover or revise knowledge of reality, and uses this knowledge to devise new applications, contribute to the development of organizational objectives</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSBA05</td>
<td>Analyse opportunity and suggest use of historical data available at organisation for organizational processes optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DSBA05</th>
<th>Develop required data analytics for organizational tasks, integrate data analytics and processing applications into organization workflow and business processes to enable agile decision making</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSEM05</td>
<td>Consistently apply data security mechanisms and controls at each stage of the data processing, including data anonymisation, privacy and IPR protection.</td>
</tr>
<tr>
<td>DSBM05</td>
<td>Ensure data quality, accessibility, interoperability, compliance to standards, and publication (data curation)</td>
</tr>
<tr>
<td>DSRM05</td>
<td>Design experiments which include data collection (passive and active) for hypothesis testing and problem solving</td>
</tr>
<tr>
<td>DSBA05</td>
<td>Analyse customer relations data to optimise/improve interacting with the specific user groups or in the specific business sectors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DSBA06</th>
<th>Visualise results of data analysis, design dashboard and use storytelling methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSEM06</td>
<td>Design, build, operate relational and non-relational databases (SQL and NoSQL), integrate them with the modern Data Warehouse solutions, ensure effective ETL (Extract, Transform, Load), OLTP, OLAP processes for large datasets</td>
</tr>
<tr>
<td>DSBM06</td>
<td>Develop and manage/supervise policies on data protection, privacy, IPR and ethical issues in data management</td>
</tr>
<tr>
<td>DSRM06</td>
<td>Develop and guide data driven projects, including project planning, experiment design, data collection and handling</td>
</tr>
<tr>
<td>DSBA06</td>
<td>Analyze multiple data sources for marketing purposes; identify effective marketing actions</td>
</tr>
</tbody>
</table>
### 10.2 Data Scientist Professional Profiles

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>TITLE</th>
<th>ALTERNATIVE TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP01</td>
<td>Data Science (group) Manager</td>
<td>Data analytics department manager</td>
</tr>
<tr>
<td>DSP02</td>
<td>Data science infrastructure manager</td>
<td>Big data infrastructure manager</td>
</tr>
<tr>
<td>DSP03</td>
<td>Research infrastructure manager</td>
<td>Research infrastructure data storage facilities manager</td>
</tr>
<tr>
<td>DSP04</td>
<td>Data scientist</td>
<td>Data analyst</td>
</tr>
<tr>
<td>DSP05</td>
<td>Data science researcher</td>
<td>Data analyst</td>
</tr>
<tr>
<td>DSP06</td>
<td>Data science architect</td>
<td>System architect, applications architect</td>
</tr>
<tr>
<td>DSP07</td>
<td>Data science application programmer / engineer</td>
<td>Scientific programmer</td>
</tr>
<tr>
<td>DSP08</td>
<td>Data analyst</td>
<td></td>
</tr>
<tr>
<td>DSP09</td>
<td>Business analyst</td>
<td>Business development manager</td>
</tr>
<tr>
<td>DSP10</td>
<td>Data stewards</td>
<td></td>
</tr>
<tr>
<td>DSP11</td>
<td>Digital data curator</td>
<td>Digital data curator, digital archivist, digital librarian</td>
</tr>
<tr>
<td>DSP12</td>
<td>Data librarian</td>
<td>Digital data curator</td>
</tr>
<tr>
<td>DSP13</td>
<td>Data archivist</td>
<td>Digital archivist</td>
</tr>
<tr>
<td>DSP14</td>
<td>Large scale (cloud) database designer</td>
<td>Large scale (cloud) database developer</td>
</tr>
<tr>
<td>DSP15</td>
<td>Large scale (cloud) database admin</td>
<td></td>
</tr>
<tr>
<td>DSP16</td>
<td>Scientific database administrator</td>
<td>Large scale (cloud) database admin</td>
</tr>
<tr>
<td>DSP17</td>
<td>Big data facilities operator</td>
<td></td>
</tr>
<tr>
<td>DSP18</td>
<td>Large scale (cloud) data storage operator</td>
<td>Large scale (cloud) data storage operators</td>
</tr>
<tr>
<td>DSP19</td>
<td>Scientific database operator</td>
<td>Large scale (cloud) data storage operators</td>
</tr>
<tr>
<td>DSP20</td>
<td>Data entry/access worker</td>
<td>Data entry desk/terminal worker</td>
</tr>
<tr>
<td>DSP21</td>
<td>Data entry field workers</td>
<td></td>
</tr>
<tr>
<td>DSP22</td>
<td>User support data services</td>
<td></td>
</tr>
</tbody>
</table>

### 10.3 Data Scientist Professional Profile competence relevance scores

| COMP1        | COMP2 | COMP3 | COMP4 | COMP5 | COMP6 | COMP7 | COMP8 | COMP9 | COMP10 | COMP11 | COMP12 | COMP13 | COMP14 | COMP15 | COMP16 | COMP17 | COMP18 | COMP19 | COMP20 | COMP21 | COMP22 | COMP23 | COMP24 | COMP25 | COMP26 | COMP27 | COMP28 | COMP29 | COMP30 | COMP31 | COMP32 | COMP33 | COMP34 | COMP35 | COMP36 | COMP37 | COMP38 | COMP39 | COMP40 | COMP41 | COMP42 | COMP43 | COMP44 | COMP45 | COMP46 | COMP47 | COMP48 | COMP49 | COMP50 | COMP51 | COMP52 | COMP53 | COMP54 | COMP55 | COMP56 | COMP57 | COMP58 | COMP59 | COMP60 | COMP61 | COMP62 | COMP63 | COMP64 | COMP65 | COMP66 | COMP67 | COMP68 | COMP69 | COMP70 | COMP71 | COMP72 | COMP73 | COMP74 | COMP75 | COMP76 | COMP77 | COMP78 | COMP79 | COMP80 | COMP81 | COMP82 | COMP83 | COMP84 | COMP85 | COMP86 | COMP87 | COMP88 | COMP89 | COMP90 | COMP91 | COMP92 | COMP93 | COMP94 | COMP95 | COMP96 | COMP97 | COMP98 | COMP99 | COMP100 |
10.4 CV of a data scientist with 8 years experience

https://www.livecareer.com/build-resume/select-resume

Data Scientist

Jason Caldwell
1141 Briarcliffe Road, Portland, OR 11111
T: 555-485-5897
E: jasoncaldwell@anymail

Professional Summary

Experienced and driven data scientist with eight years of experience in the industry. Strong background in computer programming language, and knowledge of various types of databases. Solid skills in mathematics, statistics and algorithms. Commitment to providing support and essential information about trends to companies in a variety of industries.

Work Experience

Data Scientist
July 2014 – present
• Research and transform information from raw data into an easily understood analysis that identifies trends and insights for the organization.
• Pinpoint a set of variables to evaluate and work with when deciding on the range of analysis and scope of information sought.
• Use a variety of sources inside and outside of the company to collect, aggregate and analyze data, using corresponding Big Data platforms and NoSQL databases Hbase, MongoDB, Azure DocumentDB
• Work with business oriented Big Data Analytics platforms, Hadoop tools, Tableau visualisation tools

Data Scientist
August 2011 – July 2014
• Interpreted information from a series of database investigations to make predictions and recommendations for a company’s scope of work.
• Worked with large datasets storing and processing with NoSQL databases HBase, MongoDB, Microsoft SQL.
• Discussed results of database analysis with various members of management in an organization, and led staff members to realize the significance of the data.
• Discovered industry trends based on data collection methods and analysis strategies, and used the information to help the company make production and product adjustments to increase efficiency by 12 percent.

Data Scientist
November 2008 – August 2011
• Ensured accurate and consistent statistical analysis by meticulously going through the data and validating results.
• Developed company guidelines and best practices based on information learned through an analysis of consumer behavior data.
• Determined additional means of organization improvement with employee engagement by using data collected by staff surveys.

Education and Training
10.5 CV of a data scientist with 13 years experience

CV for Data Scientist

HE ZHANG 张 鹤
Data Scientist, PhD in Machine Learning
Address: Innopoli 2, FI-02150, Espoo, Finland
Tel: +358-505188888 Email: klarke4001@gmail.com
Born: 19.08.1981, Changchun, P. R. China

KEY COMPETENCIES AND STRENGTHS
• Over 7 years research and working experience in Machine Learning and Data Mining field.
• Strong data analytical and programming skills especially with Matlab and Python.
• Excellent English writing and oral presentation skills.
• Strong team-work spirit with experience of working in highly international environments for years.
• Native Mandarin speaker with Permanent Finnish Resident and Working Permit.

WORKING & RESEARCH EXPERIENCE
2014 - 2015 Data Analyst at Verto Analytics Inc. (Area: data analytics and image recognition)
I am working on versatile projects at Verto Analytics Inc. - a Finnish Pioneer in Digital Media Research and Measurement Industry.
My responsibilities include: 1) developing and implementing machine learning algorithms for mobile-end App image recognition; 2) collaborating with marketing professionals for writing market insights reports; 3) data quality assurance, data cleaning and curation, data visualisation, and data production.

2011 - 2014 Nonnegative Learning for Data Clustering (Area: algorithms and optimisations)
I designed several Machine learning algorithms using matrix factorisation models to better detect groups or clusters in various data sets. The algorithms can be directly applied for, e.g., Recommendation Systems and Market Segmentation. I published the results in 6 scientific journals and papers.

2011 - 2014 Understanding the Emotional Impacts of Images (Area: image processing)
I developed several image processing methods to predict emotional impacts of artistic images. The methods can improve the performance of Affective Image Classification and Retrieval systems. I published the results in 4 scientific journals and papers.

2008 - 2010 PinView - A Proactive Personal Information Navigator (Area: multimedia retrieval)
I developed a Gaze-and-Speech-enhanced Content-Based Image Retrieval system that can infer the user’s search interests based on his or her feedbacks such as eye tracking data. I also implemented a client-side browser extension using JavaScript and managed to publish the results in 2 scientific conferences.

2005 - 2007 Research Assistant in the Multimedia Laboratory, Jilin University, China
I developed matrix transformation techniques for colour image and video compression.

CV: He Zhang, +358-50-5188888, klarke4001@gmail.com
10.6 CV of a data scientist with 15 years experience

Data Science - 15 Year Resume Sample

Ebony Moore
- Austin, TX
- (123) 456-7891
- emoore@email.com

SUMMARY

Senior Data Scientist with PhD in computer science and 15+ years of hands-on experience leveraging machine learning models and data mining to uncover insights and drive $1M+ in business growth.

EDUCATION

- Longford Tech
  - Current
  - PhD in Computer Science

EXPERIENCE

Cloud Clearwater
Senior Data Scientist
- Current
- Developed multivariate Gaussian anomaly detection algorithm in Python to identify suspicious patterns in network traffic
- Applied decision tree analysis, using R to predict whether an email is spam
- Led 4-person team on a data mining project to predict sales in the retail domain
- Conducted cluster analysis to generate segmented profiles of customers

Tradelot
Data Scientist
- Created machine learning models with Python and scikit-learn to predict energy usage of commercial buildings with 98% accuracy
- Reduced third-party reporting costs by $100K+ per year with machine learning models
- Developed an algorithm in R that automated financial forecasting

River Tech
Data Scientist
- Doubled campaign response rates with predictive models in R
- Used random forest algorithm to help identify loyal customers and predict the likelihood of customers buying a recommended product
- Created customized reports in Tableau for data visualization

SKILLS

- Python, Hadoop, R, SQL
10.7 CV of a Hadoop developer with 3 years experience

Hadoop Developer - 3 Year Resume Sample

Ivy Haddington
• Alpharetta, GA
• (123) 456-7891
• ihaddington@email.com

SUMMARY
Hortonworks-certified Hadoop Developer with 3+ years of experience installing, configuring, and leveraging the Hadoop ecosystem to glean meaningful insights from semi-structured and unstructured data.

EDUCATION
LONGFORD TECH
Aug ’10 - May ’14
• Bachelor of Science in Computer Engineering

EXPERIENCE
RETAIL OCEAN
Hadoop Developer
Sep ’14 - Current
• Implemented Hadoop data pipeline to identify customer behavioral patterns, improving UX on e-commerce website
• Develop MapReduce jobs in Java for log analysis, analytics, and data cleaning
• Perform big data processing using Hadoop, MapReduce, Sqoop, Oozie, and Impala
• Import data from MySQL to HDFS, using Sqoop to load data
• Developed and designed a 10-node Hadoop cluster for sample data analysis
• Regularly tune performance of Hive and Pig queries to improve data processing and retrieving
• Run Hadoop streaming jobs to process terabytes of XML data
• Create visualizations and reports for the business intelligence team, using Tableau

CRANE & JENKINS
Hadoop Developer Intern
Mar ’14 - May ’14
• Analyzed datasets using Pig, Hive, MapReduce, and Sqoop to recommend business improvements
• Setup, installed, and monitored 3-node enterprise Hadoop cluster on Ubuntu Linux
• Analyzed and interpreted transaction behaviors and clickstream data with Hadoop and HDP to predict what customers might buy in the future

SKILLS
• Hadoop big data ecosystems (MapReduce, HDFS, HBase, Zookeeper, Hive, Pig, Sqoop, Cassandra, Oozie, Talend)
10.8 CV of a Hadoop developer with 15 years experience

Hadoop Developer - 15 Year Resume Sample

Cody Fredrickson
- Boston, MA
- (123) 456-7891
- cfredrickson@email.com

SUMMARY
Principal Hadoop Developer with 15+ years of experience building scalable, distributed data solutions with 80TB+ of data and driving business improvements with innovative Hadoop and BI tools.

EDUCATION
GREEN VALLEY STATE
Aug '98
May '02
Master of Science in Computer Science

EXPERIENCE
RIVER TECH
Principal Hadoop Developer
Jul '15 - Current
- Leverage Hadoop and HDP to analyze massive amounts of clickstream data and identify the most efficient path for customers making an online purchase
- Analyze Hadoop clusters using big data analytic tools including Pig, Hive, and MapReduce
- Conduct in-depth research on Hive to analyze partitioned and bucketed data

CRANE & JENKINS
Senior Hadoop Developer
Jan '13 - Jun '15
- Developed Oozie workflow to automate the loading of data into HDFS and Pig for data preprocessing
- Architected 60-node Hadoop clusters with CDH4.4 on CentOS
- Successfully implemented Cloudera on a 30-node cluster

TRADELOT
Hadoop Developer
Mar '02 - Dec '12
- Leveraged Sqoop to import data from RDBMS into HDFS
- Developed ETL framework using Python and Hive (including daily runs, error handling, and logging) to glean useful data and improve vendor negotiations
- Performed cleaning and filtering on imported data using Hive and MapReduce
10.9 Project pages of front end and back end components

EDISON Community project page: https://github.com/EDISONcommunity/EDSFapps