Abstract—Decision-making has always been a crucial part in any organization or in fact, in any area of life. A decision-maker is often faced with multiple objectives while making choices. These choices can be conflicting or similar in nature. It is therefore helpful to have certain tools and techniques to help with informed decision-making. The aim of this project is to explore the usability and effectiveness of IBM Watson’s Tradeoff Analytics to help users in the decision-making problem of selecting best flight while making travel plans. In this study we will explore how Tradeoff Analytics can help the users make better decisions from the variety of flight options available, considering multiple criteria while making the choice. These criteria will be set according to the users preferences and can be opposing or similar to each other. The cognitive abilities of Tradeoff Analytics service will provide the users with a set of optimal alternatives and their respective tradeoffs. These tradeoffs and options can then be explored and analyzed and best option can be decided on, thus leading to informed decision-making. The Tradeoff Analytics service that will be used in this study is an IBM Watson service and is offered under the Watson Developer Cloud that has REST APIs and SDKs which implement cognitive computing to solve complex problems.

I. INTRODUCTION AND MOTIVATION

People are often encounter multiple flight options while booking airlines for their travel plans. Sometimes, these options can be overwhelming and a person with not much prior experience would not know what to choose and can end up selecting the wrong flight for their requirements, only to lead to dissatisfaction and wastage of money. There are definitely several filtering and sorting options available on the internet, but they do not help make the decisions per se, rather just present the choices. One can miss out on good options or may not be able to still find a seemingly good option from the filtered ones. This lead to the idea of using Tradeoff Analytics and studying how it helps in optimal decision-making in such case.

Decision-makers often face situations where they have to select between multiple candidate solutions and multiple conflicting criteria amongst them. Such situations can be time-consuming and also require the final choice to be as optimal as possible under the given time-constraints. Therefore, it is important to study the available options from different angles to make sure that every factor/criteria is addressed while making a choice. Sometimes, the required goals can be conflicting also. This is where Tradeoff Analytics comes into play to aid the user throughout the decision-making process. First, it reduces the options to only the optimal ones. Then the filtered options are compared on the basis of the tradeoffs between them in user-defined criteria. There are several visualization techniques that are especially helpful for visualizing and comparing multiple-objectives to enable the user to explore and analyze this in an interactive manner.

II. RELATED WORK

The two main stages in the working of a Tradeoff Analytics service involve identification of best options from large number of possible candidates followed by helping the user to select the best one through visual representations and analysis. To learn about the service better, this section will talk about the research studies that introduced different technologies that contributed to the science behind this service. We will discuss about the related-work on this technique and also on visualization-approaches that could complement the Tradeoff Analytics service, making it a more powerful tool.

The service uses Pareto Optimization techniques for finding the optimal solutions for the user. In order to provide a small set of options from a pool of larger alternatives, the service uses Pareto Optimization technique, where a partial order called Pareto Domination is given to all the objectives[1]. In the paper[2] the authors suggest a computerized method that can select optimum group of candidates from several multi-objective designs. The selection of these options is performed on the basis of match between an objective and the gain and loss thresholds. Multi-objective Optimization helps to optimize solutions with multiple and conflicting goals, all done simultaneously. It differs from an otherwise single-set optimization because it returns Pareto Frontier of a given-decision problem. This means, that a set of optimal solutions is returned, as opposed to a single solution based on plain maximization and minimization of objectives. These set of options are selected on the basis of the satisfaction of the Pareto Optimality conditions that are met. According to[2],
these conditions are. 1. Other inferior solutions should not dominate any solution within the set 2. there exists no solution that is straightforward better than the others in all aspects and criteria. Therefore, it is through this set of options that a decision-maker will make a final choice. The study on multi-criteria optimization is taken a step further in the paper[3] where an attempt is made to identify the reasons behind the observation that the standard technique Pareto technique works well when the weighted sums of different goals are minimized. Moreover, evenly distributed points are not yielded from all sections of the Pareto Frontier.

In paper[4] the authors conducted experiments that could provide insights into what decision-makers find helpful to aid the process. This enabled them to come up with interactive tools to support decision-making. Several observations were made for example, the subjects found it easier to interpret data that was shown directionally like parallel coordinate plots, and plain representations in terms of ranking of options resulted in being ineffective. This pointed towards the need of better representation of ranked options. There was also a need to make comprehension of representations clearer for people who do not feel comfortable with computation-based representations. The authors in paper[5] proposed Interactive Decision Maps which use scatter-plot matrices to show the Pareto Frontier in slices where every slice represents a scatter-plot of two-objectives. This way, the optimal options can be explored by changing the values of the objectives. The Parallel Coordinates method suggested in the paper[6] is to-date the most effective technique to visualize Pareto Frontiers. However, this technique has a drawback of becoming cluttered as the number of probable options scale-up. The recommended visualization of the Tradeoff Analytics service has introduced a new interactive-approach that follows on the lines of paper[7]. The paper brings forth an approach that suggests to provide the users with an overview followed by filtering and then providing the details, if desired. There have also been several studies that helps the Pareto Frontier to be visualized and interpreted easily. For example, in paper[8], the authors address the problem of visualizing of optimal set of options in a multi-objective decision making scenario. The authors designed a Self-Organizing Map(SOM) for visualization of Pareto Frontier. The visualization enabled interaction and the users could explore the map, while analyzing the options. Lastly, the authors proposed a modification of the SOM algorithm (SOMMOS - Self-Organizing Maps for Multi-Objective Pareto Frontiers) so that anchor points could be introduced within the map so that every objective could be represented since this was not the case in the original SOM algorithm. It was also observed how important it was to have properly arranged objectives on the map so that the visualization could be more effective.

III. ABOUT THE SERVICE

Decision-making is part of our daily life. Cognitive computing aims to help humans with complex decision making problems. There is always a chance that one may loose out on good options in guesswork or take a lot of time to pick the best option. Thorough analysis of the available choices helps a great amount in such cases. If the preference or criteria that we want in our final decision are determined, we can see how the different choices perform in them. But there are not only various options to consider; our criteria can be multiple too. Moreover, they can be similar or opposing. This makes the entire decision-making process even more complicated than it may already seem. This is especially the case in big organizations to select a particular scheme/policy or while making financial decisions. It definitely helps to have a computing machine that aids human judgment in this scenario - a machine that can perform all computations and calculations under the hood, or conduct all intricate analysis with fine details that may skip our eye.

Tradeoff analytics is one such service that aims to help with complex decision-making by the means of helping the user analyze the options more carefully and understand the tradeoffs between the choices. It does not impose a single answer on the user, but, makes sure that the user is well-informed to make the best decision for himself. The service first helps reduce the option-space by limiting to the best options. Based on these options, user interaction can take place to explore tradeoff between the options through powerful visualizations.

A. Internal working

The main stages of the internal working of the service are:
1. Selecting top options: From a large pool of options, first step is always to narrow them down based on how well they meet the user criteria. This is done using a mathematical technique called Pareto Optimality. This technique has its advantages for example, the ability to take more than two criteria for selection. Therefore, when an input is provided to the service, multi-objective Pareto filtering is first conducted. Top options are decided based on satisfaction of all the criteria that are being used. The user can then focus this set of top options and the other are discarded. 2. Exploring tradeoffs: The next stage is to explore the tradeoffs between the options resulted from the above mentioned stage. This is done through effective and interactive visualizations. As opposed to conventional visualizations, they can handle large dimensionality of the data with multiple objectives. IBM introduces and recommends Parallel Coordinates or even Maps where goals are represented in each column. 3. Using cognitive abilities to help the user: This means that Tradeoff Analytics can show that compromising on a seemingly better option can gain better results in the longer run. What is to be finally selected is upto the user, but a complete picture of the problem and solutions is presented.

B. Input-Output

When using the service, a decision problem is passed to the HTTP POST dilemmas method of the service. The decision problem is represented as a JSON object. This JSON object includes any number of objectives/parameters and options, along with their details from the dataset. Figure 1. shows the representation of input problem for Tradeoff Analytics.
When defining a Column object/the criteria object, the following fields were provided:

- **Key** - unique identifier to the column;
- **Type** - text, numeric, categorical, datetime;
- **Goal** - whether the goal is to minimize or maximized;
- **IsObjective** - indicates whether the objective is supposed to play a role in the problem;
- **Range** - range of valid values for the objective.

When defining an Option object/the criteria object, the following fields were provided:

- **Key** - unique identifier of the option;
- **Values** - set of values that enlist the values for every column/objective defined above. These values then map to the key on a per option basis.

After obtaining the results, the service returns a 'Dilemma' object that contains 'problem' and 'resolution'. The 'resolution' contains details of the solutions and determine how the options compare with each other according to the service. The Resolution in turn contains an array of Solution objects. The following fields are defined in a Solution object:

- **SolutionRef** - unique identifier of the option in the decision problem;
- **Status** - status of the service’s analysis for each option in the problem. The values returned as status are:
  - FRONT (represents that this option can be included amongst the optimal options),
  - INCOMPLETE (represents that this option does not have certain values specified as per the problem requirement),
  - EXCLUDED (represents that other options are better than this),
  - DoesNotMeetPreference (represents that a value specified by this option is not included in the preference).

**Figure 1. Input Structure for Tradeoff Analytics (courtesy:IBM)**

**C. Tradeoff Analytics vs Traditional Filtering**

A common question that arises after looking at the service is that how is it different from traditional filtering tools that also sort and filter and present the results to the user. For example, in the concept of flight booking problem, there are several commercial flight booking sites that help user filter out options to select a flight. The main difference here is that these tools do not help to make the decision. One can miss out on good options. If we narrow down our options during filtering, we will not know that there could have been some better opportunities in the filtered out options. In other case, we may filter out considerably but still end up with many options that we do not actually find good. Tradeoff Analytics is designed to show that one can get more out of other option if one is ready to make a small compromise in a particular criteria. Also, one may be looking at many goals and not just one or two criteria. For example, for selecting a flight, other than price and duration which are the basic criteria, there can also be layover city choices, layover time, comfort index, etc. What is eventually decided is up to the user, but Tradeoff Analytics will help see the complete picture so that it helps in informed decision-making.

**IV. APPROACH**

In the following subsections, we will discuss about each phase of the project.

**A. Data Preparation**

An airline dataset was prepared by referring to U.S. Transportation website and commercial airline-booking sites. This was done to keep the prices and travel duration as realistic as possible. Subjective factors like 'comfort' were represented through a comfort-index that was calculated using U.S. Airline Sentiment Analysis data provided by Kaggle, that contained sentiment of the tweets for six US airlines Figure 3. The index was between 0 to 10 which corresponded to the percentage of positive and negative tweets from the overall number of tweets. Airline routes considered in this project are the top domestic routes enlisted by the Bureau of Transportation Statistics (Figure 4.) This data is taken during the period from March 2015-February 2016, and is therefore an updated and latest information.

**Figure 2. Dataset Description**

<table>
<thead>
<tr>
<th>Serial No.</th>
<th>Attribute Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AIRLINE</td>
<td>nominal</td>
<td>Name of the airline carrier</td>
</tr>
<tr>
<td>2</td>
<td>SOURCECity</td>
<td>nominal</td>
<td>Name of the source city</td>
</tr>
<tr>
<td>3</td>
<td>SOURCEAirport</td>
<td>nominal</td>
<td>Airport code of the source</td>
</tr>
<tr>
<td>4</td>
<td>DESTINATIONCity</td>
<td>nominal</td>
<td>Name of the destination city</td>
</tr>
<tr>
<td>5</td>
<td>DESTINATIONAirport</td>
<td>nominal</td>
<td>Airport code of the destination</td>
</tr>
<tr>
<td>6</td>
<td>DEP_TIME</td>
<td>numeric</td>
<td>Time of flight departure</td>
</tr>
<tr>
<td>7</td>
<td>ARR_TIME</td>
<td>numeric</td>
<td>Time of flight arrival</td>
</tr>
<tr>
<td>8</td>
<td>PRICE</td>
<td>numeric</td>
<td>Cost of the flight</td>
</tr>
<tr>
<td>9</td>
<td>LAYOVERCITY</td>
<td>nominal</td>
<td>Name of layover city</td>
</tr>
<tr>
<td>10</td>
<td>LAYOVERNUM</td>
<td>numeric</td>
<td>Number of layovers</td>
</tr>
<tr>
<td>11</td>
<td>LAYOVERTIME</td>
<td>numeric</td>
<td>Duration of layover</td>
</tr>
<tr>
<td>12</td>
<td>TRAVELTIME</td>
<td>numeric</td>
<td>Duration of entire travel</td>
</tr>
<tr>
<td>13</td>
<td>COMFORT</td>
<td>numeric</td>
<td>Index based on user reviews</td>
</tr>
</tbody>
</table>
B. Application development

Tradeoff Analytics service is offered under the IBM Watson Developer Cloud. The approach that was decided upon, in order to use the Watson Developer Cloud services was the use of software development kits that implemented Watsons cognitive computing to help solve problems.

Platform: Android
SDKs: IBM Watson Java Sdk-3.0.0

The back-end of the application was a multi-threaded server created in Java using Networking APIs and was responsible for fetching appropriate records from the dataset. All other logic including User Interface, formulating a problem statement, making API calls and displaying the results in a List view with Watsons decision on it; was incorporated in the Android framework.

Initially for the project, a Java application was built, that performed all the tasks from input processing, API calls to the output acquiring. IBM Watson Java Sdk-3.0.0 was also used for this. After ensuring that the application functioned properly on a local machine, it was decided to build an interface to handle the interaction with the user. This is because the input-output was taking place through the command line and the output from the service was very raw (as shown in Figure 5.1), which could lead to inability of the decision-maker to comprehend the results. There are client libraries that are provided and recommended by IBM for enabling visualizations and user-interactions. However, I decided against using them and attempted to build a very basic interface by myself. Thus, the handling of client-related tasks was done with an Android application.

```
SERVICE OUTPUT:
{
    "solutions": [
        {"solution_ref": "1","status": "EXCLUDED"},
        {"solution_ref": "2","status": "FRONT"},
        {"solution_ref": "3","status": "FRONT"},
        {"solution_ref": "4","status": "FRONT"},
        {"solution_ref": "5","status": "FRONT"}
    ]
}
```

Figure 5.1. Initial Raw Output for an Airline Decision Problem

The source, destination and criteria that may be desired for booking a flight were taken in as a Query object through the client application. These query-details are then sent to the server, hosted on local machine. The server is responsible for accessing the prepared airline database and sending the relevant flight-records back to the client-side (Android) application. Here, the extracted data is converted as a problem statement in the service-required input format (explained in the previous section) where the customized criteria of the user and all possible options from the provided source and destination are defined. The problem is then sent to the service, and the results are obtained from the service and are displayed in a new page, where the details of all the candidate flights are shown, along with the services suggestion of each candidate being a good option or not.

A pictorial representation of the architecture of the application is given below.
Figure 5.2

Figure 6. shows the objective and options definition in a sample query that contains only two criteria and two options. As it can be observed, the withObjective() field determines whether that particular criteria is desired by the user or not. If not, then it will not be included by the service to get the solution. For example, if price is not an issue, the price objective for that particular query will be defined as withObjective(False). Similarly, the withGoal() field determines whether that particular criteria is desired to be minimized or maximized for the decision-making problem. For example, if comfort is a desired objective and is required to be as high as possible in the optimal solutions, the goal of the comfort objective for that particular query will be defined as withGoal(Goal.MAX).

```
"options": {
  "key": "1",
  "name": "Delta",
  "values": {
    "price": 200,
    "comfort": "6"
  }
},
{
  "key": "2",
  "name": "Southwest",
  "values": {
    "price": 267,
    "comfort": "4"
  }
},
```

Figure 6. Representation of a Decision Problem ‘Object’

User specifies the selection criteria for a flight including price, comfort index, travel duration and number of layovers. The candidate records are then pulled out from the dataset for the given source and destination. Objectives are set according to user preferences, for example, minimum price and maximum comfort. User manipulates the number of options presented by determining the range of values of the objectives that they had customized. Final decision of the appropriate flight-choice is reached by the user by weighing the options and their criteria. With every new user-query, a different definition of objectives and options were established and sent to the service dynamically in order to adapt to changing user queries. Incorporation of the interface, basic as it may be, brings in a look-and-feel to the original stand-alone application, making the whole decision-making process easier for the user.

V. EVALUATION AND DISCUSSION

The developed Tradeoff Analytics application successfully suggests appropriate flights when the user enters the source, destination and personal preferences for flight selection. These preferences/criteria offered to the user to choose from are: Price, Number of Layovers, Travel Duration and Comfort. In my knowledge, there was not any quantitative or comparative technique to test the application or more so, to test the service. It is an exploratory project where the usefulness of the Tradeoff Analytics technology is tested on a given problem statement of appropriate airline selection given a source and destination. Secondly, even though the service presents the users with a list of optimal options and its suggestions on the tradeoffs, the final decision is on the user because making a choice from those optimal and equally good candidates becomes a subjective matter. This leads to absence of any metrics to evaluate the final decision made for the given problem. Application was tested on different combinations of user inputs with different cities and criteria. It was observed that the service adapts well to the varying user inputs. Overall, the service did well in terms of scalability and dealing with queries of varying complexities presented to it. Users have
the flexibility to determine the desired ranges and although numeric ranges were used in this project, ranges in categorical and date-time formats are also accepted by the service.

One disadvantage that was observed was that even though service can handle large number of options and criteria, it accepts only a fixed format for input and is not flexible. Hence it fails to deal with unstructured data. Another shortcoming that was observed was that the usage of this service without the IBM-provided interface libraries hinders proper result consumption and interaction due to the raw output obtained. This does not leave much room for user interactions.

Sample query is used to test the application. Here, the source is 'Los Angeles' and destination is 'New York'. Figure 5. shows how the application initially handled input and outputs, and the Figure 7. and 8. shows results of the same query after using Android and building a separate client-interface with it. The result from the service comes up in the form of a reference number of the that particular solution, that is used to pull out the respective airline record, and a status or result of the service’s analysis for each record as compared to other options.

Decision sent by the service indicate ‘status’ of that candidate, such as ‘FRONT’, ‘EXCLUDED’ and ‘INCOMPLETE’.

![Figure 7. User Interface](image)

Figure 7. User Interface

![Figure 8. Displaying the results of one Tradeoff Analytics call](image)

Figure 8. Displaying the results of one Tradeoff Analytics call

As we can see, in comparison to Figure 5, the use of an interface remarkably improves not only the look-and-feel of the application, but also makes the results easier for the user to comprehend. The application currently makes one API call per user-query, thus only displaying the optimal results. More interaction and repetitive calls to the service after bringing in slight changes on the existing-query can help users explore the options even better, thus slowly and eventually converging towards a single choice. It was however concluded that sophisticated and recommended visualization techniques like maps and glyphs are the best tools to use Tradeoff Analytics in its maximum effectiveness.

VI. CONCLUSION AND FUTURE WORK

Despite there being several filtering and sorting based airline-booking sites available, the cognitive aspect of Tradeoff Analysis enables efficient decision-making. Although this service was tested on a small and common problem of flight selection, it can find its application in several areas including drug-selection by doctors, investment schemes selection or e-commerce usage helping customers decide what to buy.

In the future, the project can be taken further to develop a complete commercial application that provides real-time data and enables the user to book their flights after exploring the tradeoffs with sophisticated interface and visualizations.

REFERENCES


