

HUMAN BEHAVIOUR PREDICTION APPLICATION
BUILT OVER
DEEPDIVE-A DATA MANAGEMENT SYSTEM

SNEHA DAS
UNIVERSITY OF CALIFORNIA, SANTA CRUZ

TABLE OF CONTENTS

BACKGROUND	3
INTRODUCTION-DEEPDIVE	4
DEVELOPING A DEEPDIVE APPLICATION	6
WORK DONE	8
EVALUATION AND RESULTS	14
CONCLUSIONS, FUTURE WORK & SCOPE	16
BIBLIOGRAPHY	17

BACKGROUND

As "dark fiber" is to the telecommunication industries, is "dark data" to many businesses and organizations. These vast pools of untapped, largely unprotected data simply sit there, doing nothing. A few concerns that come up are whether the data is of sufficient quality to support the different business purposes for which it is being used or would there be any specific issues within the data decreasing its suitability for these business purposes. With the myriad of ways that data is captured like through online transactions, manual screen entry, spreadsheet uploads, direct database changes, there are many opportunities for unstructured data to be left behind untapped. Hence there is a need to structure data before using it.

Dark data is a type of unstructured, untagged and untapped data that is found in data repositories and have not been analyzed or processed. It is similar to big data but differs in how it is mostly neglected by business and IT administrators in terms of its value. Dark data is also known as dusty data. It is a form of data found in log files and data archives stored within large enterprise class data storage locations. It includes all data objects and types that have yet to be analyzed for any business or competitive intelligence or aid in business decision making. Typically, dark data is complex to analyze and stored in locations where analysis is difficult. The overall process can be costly. It also can include data objects that have not been seized by the enterprise or data that are external to the organization, such as data stored by partners or customers. According to ComputerWeekly.com, 60% of organizations believe that their "BI (business intelligence) reporting capability" is "inadequate" and 65% say that they have "somewhat disorganized content management approaches"

Many companies in the tech industry are looking at creating "cognitive computer systems" that are able to analyze unstructured dark data. The IBM Watson is considered to be a future system that would be able to analyze this unstructured data and be able to produce meaningful results that will utilize a lot of dark data that it is either practically impossible or very difficult to process at present. It is generally considered that as more advanced computing systems for analysis of data are built, the higher the value of dark data will be. It has been noted that "data and analytics will be the foundation of the modern industrial revolution". Of course, this includes data that is currently considered "dark data" since there aren't enough resources to process it. All this data that is being collected can be used in the future to bring maximum productivity and an ability for organizations to meet consumer demands. Furthermore, many organizations do not realize the value of dark data right now, for example, healthcare and educational organizations deal with large amounts of data that could create a significant potential to service students and patients in a manner in which the consumer and financial services pursue their target population.

The main idea behind the project is to analyze different datasets consisting of answers to online personality questionnaires and creating a knowledge base (structured database) out of this unstructured dark data for human behavior prediction research. This will be implemented as an application over DeepDive, a new type of data management system developed at Stanford, that enables one to tackle extraction, integration, and prediction problems in a single system, which allows users to rapidly construct sophisticated end-to-end data pipelines, such as dark data BI (Business Intelligence) systems.

INTRODUCTION - DEEPDIVE

DeepDive is a system used to extract value from dark data. It is an end-to-end framework for building Knowledge Base Construction systems. It helps bring meaning out of dark data by creating structured data (SQL tables) from unstructured information (text documents) and integrating such data with an existing structured database. It extracts sophisticated relationships between entities and make inferences about facts involving those entities.

Knowledge Base Construction (KBC) is the process of populating a knowledge base (KB), i.e., a relational database storing factual information, from unstructured and/or structured input, e.g., text, tables, or even maps and figures. DeepDive is one such automatic KBC system that is designed to take advantage of multiple (noisy) input data sources, existing knowledge bases and taxonomies, domain knowledge from scientists, state-of-the-art natural language processing tools, and, as a growing trend, machine learning and statistical inference and learning. It can declaratively specify domain knowledge without worrying about any algorithmic performance, or scalability issues. The input to the system is a heterogeneous collection of unstructured, semi-structured, and structured data, ranging from text documents to existing but incomplete KBs. The output of the system is a relational database containing facts extracted from the input and put into the appropriate schema. Creating the knowledge base may involve extraction, cleaning, and integration.

The execution model of DeepDive consists of three phases:

- (1) Feature extraction,
- (2) Probabilistic knowledge engineering, and
- (3) Statistical inference and learning.

Feature Extraction: There are four types of objects that the system seeks to extract from input documents, namely entities, relations, mentions, and relation mentions. DeepDive provides an abstraction to represent each type of objects as database relations and random variables, and provides a language for the user to specify indicators and the correlations between these objects. To extract features, the user specifies a mapping that takes as input both structured and unstructured information, and output features associated with entities, mentions, relations, or relation mentions.

- The user runs existing tools, such as Optical Character Recognition (OCR) and Natural Language Processing (NLP) tools to acquire information from text, HTML, or images. The output might contain JSON objects to represent parse trees or DOM structures.
- The user also has the ability to write user defined functions (UDF) in languages such as Python, Perl, or SQL, to further process the information produced by existing tools.
- DeepDive provides a way for the user to write these feature extractors, and automatically learn the weights, i.e., strengths, of features from training example provided by the user.

Probabilistic knowledge engineering: The integration of knowledge into computer systems is becoming increasingly essential in the development of automated solutions to complex problems, which would otherwise require a high-level of human expertise. Probabilistic inference is the task of deriving the probability of one or more random variables taking a specific value or set of values. It allows the user to write inference rules to specify how to create the factor graph. Factor graphs

are probabilistic graphical models, used for the statistical inference and learning phase. The user writes SQL queries to instruct the system about which variables to create. These queries usually involve tables populated during the extraction step. The variable nodes of the factor graph are connected to factors according to inference rules specified by the user, who also defines the factor functions which describe how the variables are related. The user can specify whether the factor weights should be constant or learned by the system. DeepDive writes the graph to a set of five files: one for variables, one for factors, one for edges, one for weights, and one for metadata useful to the system. The format of these files is special so that they can be accepted as input by the sampler.

Statistical inference and learning: DeepDive can learn the weights of the factor graph from training data that can be either obtained through distant supervision or specified by the user while populating the database during the extraction phase. In DeepDive, the default algorithm implemented for inference and learning is Gibbs sampling and stochastic gradient descent. The main general way for learning the weights is maximum likelihood. The learned weights are then written to a specific database table so that the user can inspect them during the calibration of the process. The final step consists in performing marginal inference on the factor graph variables to learn the probabilities of different values. The sampler takes the grounded graph as input, together with a number of arguments to specify the parameters for the learning procedure. The results of the inference step are written to the database. The user can write queries to analyze the results. DeepDive also provides calibration data to evaluate the accuracy of the inference. It performs joint inference (determines the values of all events at the same time). This allows events to influence each other if they are (directly or indirectly) connected through inference rules. Thus, the uncertainty of one event may influence the uncertainty of another event. As the relationships among events become more complex this model becomes very powerful.

DEVELOPING A DEEPDIVE APPLICATION

To build a KBC system with DeepDive, a developer writes a DeepDive program in a high-level declarative language. The ultimate goal is to keep the system developed flexible and simple such that domain experts can deal with diverse types of data and integrate domain knowledge easily.

There have been many applications that were built using the DeepDive framework such as PaleoDeepDive and GeoDeepDive which are KBC systems developed over DeepDive

Long Term Objective of the application to be developed:

The goal is to develop an application and create a knowledge base to predict human behavior with the help of simple statistics obtained from raw data i.e. collected from personality tests available on the web for research, in DeepDive.

Raw data corpus used is taken from: http://personality-testing.info/_rawdata/

All data is anonymous. Users were informed at the beginning of the test that their answers would be used for research and were asked to confirm that their answers were accurate and suitable for research upon completion (those that did not have been removed from these datasets). The downloadables are .zip files each containing a .csv file with the data (open with Open Office Calc, or Microsoft Office Excel) and a .txt codebook.

Project Goals: Phase 1

- a) Create a knowledge base to predict human behavior with the help of simple statistics obtained from three different datasets in the chosen raw data corpus namely Cattell's 16 Personality Factors Test, Big Five Personality Test, Rosenberg Self-Esteem Scale tests.
- b) The goal is to take in a set of unstructured (and/or structured) inputs, and populate a relational database table with extracted outputs, along with marginal probabilities for each extraction.

Application Skeleton:

The user writes the input data and the data to be produced in DDlog a schema, along with how data should be processed and transformed. Data transformation rules as well as user-defined functions (UDFs) written in Python are used for defining the data processing operations. Then, using the processed data, a statistical inference model describing a set of random variables and their correlations can be defined—also in DDlog—to specify what kind of predictions are to be made by the system.

1. **Defining data flow in DDlog:** DDlog is a higher-level language for writing DeepDive applications. A DDlog program is a collection of declarations and rules. Each declaration and rule ends with a period (.). Comments in DDlog begin with a hash (#) character. A DDlog program consists of Schema declarations for relations, Normal derivation rules (Relation derived (head atom), Relations used (body atoms), Conditions), User defined functions (UDFs) (Function declarations, Function call rules), Inference rules. All DDlog code is placed in a file named `app.ddlog`.

Similar to a SQL table definition, a schema declaration is just the name of a relation followed by a comma separated list of the column names and their types. Currently, DDlog maps the types directly to SQL, so any type supported by the underlying database can be used, e.g., PostgreSQL's types.


2. **Writing User Defined Functions in Python:** DeepDive supports user-defined functions (UDFs) for data processing, in addition to the normal derivation rules in DDlog. UDF can be any program that takes tab-separated values (TSV or PostgreSQL's text format) from stdin and prints TSV to stdout.
3. **Specifying a statistical model in DDlog:** Every DeepDive application can be viewed as defining a statistical inference problem using input data and data derived by a series of data processing steps. DeepDive requires the user to specify the name and type of the variable relations that hold random variables used during probabilistic inference. Currently DeepDive supports Boolean (i.e., Bernoulli) variables and Categorical variables. Variable relations are declared in `app.ddlog`. After declaring a variable relation, its scope needs to be defined along with the supervision labels. That means, all possible values for the variable relation's columns must be defined by deriving them from other relations, and whether a random variable in the relation is true or false (Boolean), or which value it takes from its domain of categories (Categorical) must be defined using a special syntax.

WORK DONE

The objective in mind during the development of this application was that the application needs to answer some questions related to personality like in the “spouse mention” example described in the example application provided on the webpage where the probability of a person being the spouse of another, according to news articles was found. In such a case, we need to convert the raw data into structured data which could be later used for finding probabilities of a question defined.

In data science, we first come up with well-defined questions. Next we seek data which can answer these questions. According to the question we strip out and convert the data into meaningful data to find answers to that particular question.

For example, if the Big five personality test is considered, at the end of the result they have factor labels such as given below, and this is collected from various persons in the data.csv file for that test.

Factor	Factor label	Raw score	Score percentile
I	Extroversion		50
II	Emotional stability		52
III	Agreeableness		14
IV	Conscientiousness		31
V	Intellect/Imagination		8

Big five personality trait scores calculated by openpsychometrics.org

So based on this data, we could answer questions like: if people selected 'strongly agree' to a certain set of questions ,that means their probabilities of one particular factor label is higher/hence this could also give out probabilities.

A. Datasets chosen:

The datasets chosen from the raw data and the questions defined for these datasets were as follows:

- 1) **Cattell's 16 Personality Factors Test:** In his explorations of personality, British psychologist Raymond Cattell found that variations in human personality could be best explained by a model that has sixteen variables (personality traits), using a statistical procedure known as factor analysis. Following this discovery he went on to create and promote the 16PF Questionnaire. This test uses a public domain scales from the International Personality Item Pool to measure the same traits.

This personality test consists of 164 statements about yourself. The test taker has to indicate how accurate it is on the scale of (1) disagree (2) slightly disagree (3) neither agree nor disagree (4) slightly agree (5) agree. The dataset used consisted the answers of 49,159 test takers.

The question defined for this dataset was:

"Predict which of the people might play a better leadership role?"

- 2) **Big Five Personality Test:**

The big five personality traits are the best accepted and most commonly used model of personality in academic psychology. The big five come from the statistical study of responses to personality items. Using a technique called factor analysis researchers can look at the responses of people to hundreds of personality items and ask the question "what is the best way to summarize an individual?". This has been done with many samples from all over the world and the general result is that, while there seem to be unlimited personality variables, five stand out from the pack in terms of explaining a lot of a person's answers to questions about their personality: extroversion, neuroticism, agreeableness, conscience and openness to experience. The big-five are not associated with any particular test, a variety of measures have been developed to measure them. This test uses the Big-Five Factor Markers from the International Personality Item Pool, developed by Goldberg (1992).

The test consists of fifty items that you must rate on how true they are about you on a five point scale where 1=Disagree, 3=Neutral and 5=Agree. The dataset used consisted the answers of 19,719 test takers.

The question defined for this dataset was:

"Predict which of the people might be an introvert or extrovert?"

- 3) **Rosenberg Self-Esteem Scale:**

This scale is the most widely used measure of self-esteem for research purposes.

The scale has been used in more than one hundred research projects. Because the concept of self-esteem is one most people should be familiar with, this test will probably not tell you anything you do not already know. You should have a pretty good grasp of your results just by asking yourself the question, "Do I have low self-esteem?" The scale

can however give you a better picture of your state in relation to other people. Your results will also include a little bit more about the relationship between self-esteem and life outcomes.

The scale consists of ten statements that you could possibly apply to you that you must rate on how much you agree with each. The items should be answered quickly without overthinking, your first inclination is what you should put down.

The question defined for this dataset was:
"Predict which of the people might be successful?"

B. Application Directory Structure:

A DeepDive application is a directory that contains the following files and directories:

- a) **app.ddlog:** The file is the blueprint of the DeepDive application. DDlog declarations and rules written in this file tell DeepDive what each relation looks like and how one is derived from others, what user-defined functions are there, what input/output schema they expect, and their implementation details, what random variables are to be modeled and how they are correlated. The questions designed for the different datasets are defined in this file.
- b) **deepdive.conf:** Extra configuration not expressed in the DDlog program is in this file. Extractors, and inference rules can also be written in HOCON syntax in this file, although DDlog is the recommended way. The HOCON syntax keeps the semantics (tree structure; set of types; encoding/escaping) from JSON, but make it more convenient as a human-editable configurable file format.
- c) **db.url:** A URL representing the database configuration is supposed to be stored in this file. For example, the following URL can be the line stored in it:
Eg: postgresql://postgres@snehasvm:5432/deepdive_person_success_postgres
- d) **Input/:** Any data to be processed by this application is suggested to be kept under this directory. The csv answer dataset file is decompressed and placed in the input folder. The people.tsv.sh file extracts information such as age, gender, country. It also adds a field called people id to identify the user since the user's name is not available for all the datasets considered.

id	age	gender	country
1	17	1	US
2	37	1	US
3	31	1	US
4	32	1	US
5	46	2	NZ
6	36	2	IT
7	35	1	US
8	61	2	US
9	17	2	US
10	19	1	US
11	21	1	US
12	23	2	US
13	17	2	US
14	24	2	TR
15	21	2	PH
16	37	2	US
17	32	2	NL
18	21	1	IN
19	25	1	US
20	22	2	US
21	19	1	US
22	44	1	US
23	26	2	US
24	18	2	FI
25	21	2	US
26	22	1	US
27	20	2	US
28	21	2	US
29	20	2	US
30	21	2	US
31	19	1	US
32	19	1	US
33	19	1	US
34	21	1	US
35	22	1	US
36	15	2	US
37	21	1	US
38	33	1	CA
39	24	2	US
40	23	1	PH
41	37	1	IN
42	27	2	PH
43	33	1	US

The people_personality.tsv.sh file for example used for Cattell's 16 Personality Factors Test extracts information about the user's personality, his ability to bring people together, lead , cheer up people, trust people etc. .

people_id accuracy	comfort_others	bring_people_together	cheer_people	dislike_myself	wait_for_others	difficulty_approach_others	distrust_people	afraid_doing_things	listener
1 92	1	4	3	3	3	4	3	3	4
2 100	4	3	4	4	4	2	2	3	4
3 80	3	4	4	2	3	3	3	4	5
4 93	4	5	4	2	3	3	3	3	5
5 87	4	0	4	4	1	2	4	4	4
6 80	3	5	4	4	2	2	3	2	4
7 80	4	2	2	1	3	4	2	4	5
8 100	4	5	4	3	2	4	2	4	4
9 78	2	4	5	2	4	2	2	4	4
10 95	5	5	5	1	2	2	3	3	4
11 95	5	5	5	1	3	4	3	4	5
12 82	5	4	5	1	1	1	2	2	4
13 90	5	4	4	1	1	2	3	3	4
14 90	5	4	4	3	5	2	4	4	5
15 54	3	3	3	3	3	3	3	3	3
16 90	4	4	4	4	2	2	4	5	4
17 70	5	5	5	3	2	1	5	3	5
18 95	5	5	5	1	1	1	2	1	5
19 97	4	4	5	1	2	1	3	1	5
20 85	4	4	4	3	3	2	2	4	4
21	5	4	4	1	2	2	2	4	4

- e) **udf/**: Any user-defined function (UDF) code is suggested to be kept under this directory. They can be referenced from deepdive.conf with path names relative to the application root.
- f) **Run/**: Each run/execution of the DeepDive application has a corresponding subdirectory under this directory whose name contains the timestamp when the run was started. All output and log files that belong to the run are kept under that subdirectory.

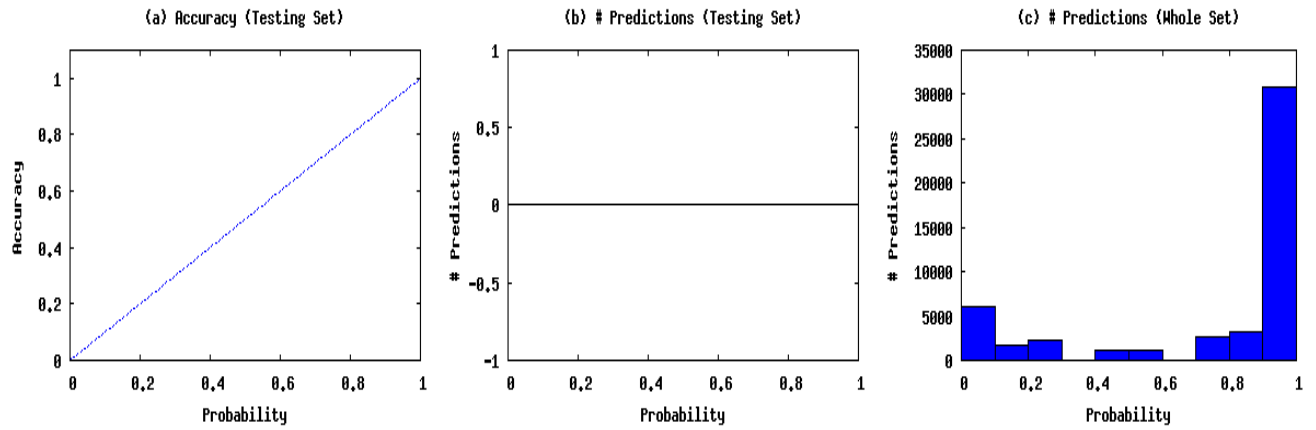
Heuristic rules were written giving positive and negative weights. For example, ability to bring people together gives a +2 weight. Distrusting people will have a -3 weight. All rules were written into the app.ddlog file. The rules are basic and the results can be improved by modifying the rules.

DeepDive predicts on these defined rules for all the users.

people_id	expectation
1	0.982
2	0.951
3	0.98
4	1
5	0.503
6	0.951
7	0.504
8	0.999
9	0.894
10	1
11	0.995
12	1
13	1
14	0.953
15	0.478
16	0.882
17	0.992
18	1
19	1
20	0.999
21	0.997
22	1
23	0.038
24	0.729
25	1
26	0.982
27	1
28	0.999
29	0.992
30	0.247
31	1
32	0.012
33	1
34	0.985
35	1
36	0.952
37	0.999
38	0.021
39	0.999
40	0.044
41	1
42	0.122
43	0.493

EVALUATION AND RESULTS

These predictions were not based on the training set but based on heuristic rules. Hence a and b images are not formed.

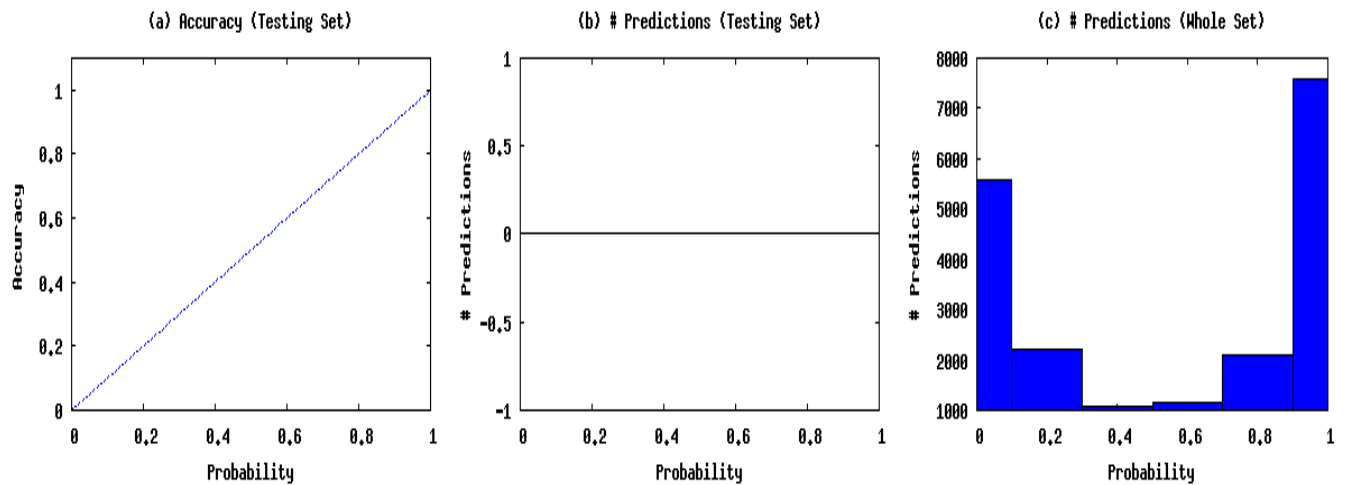


* (a) and (b) are produced using hold-out on evidence variables; (c) also includes all non-evidence variables of the same relation.

Similarly results obtained for:

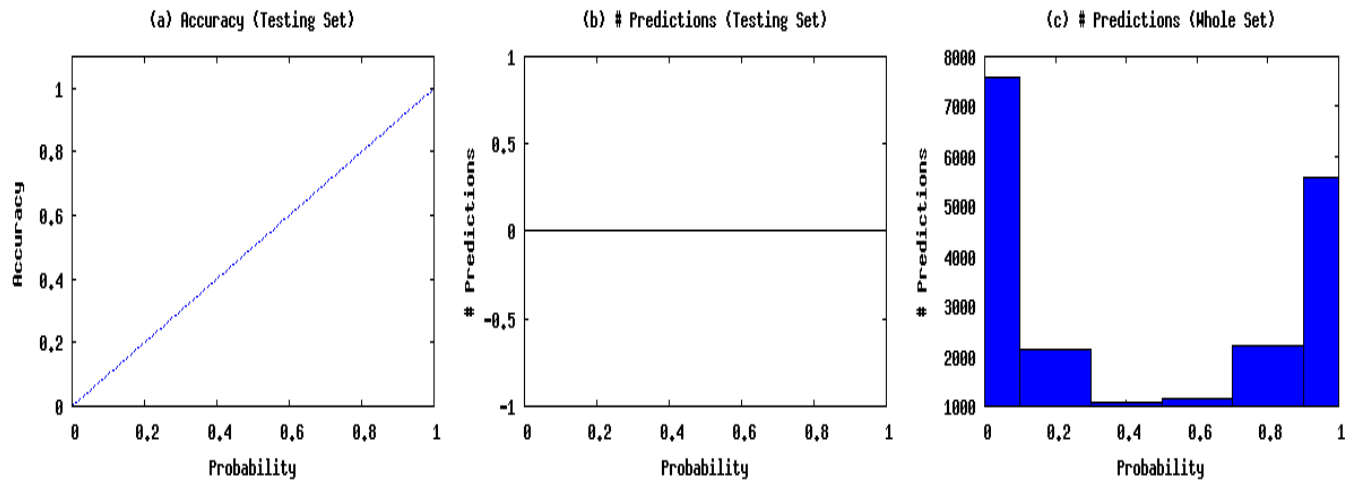
The Big five Personality tests were as follows:

A) If person is an introvert?

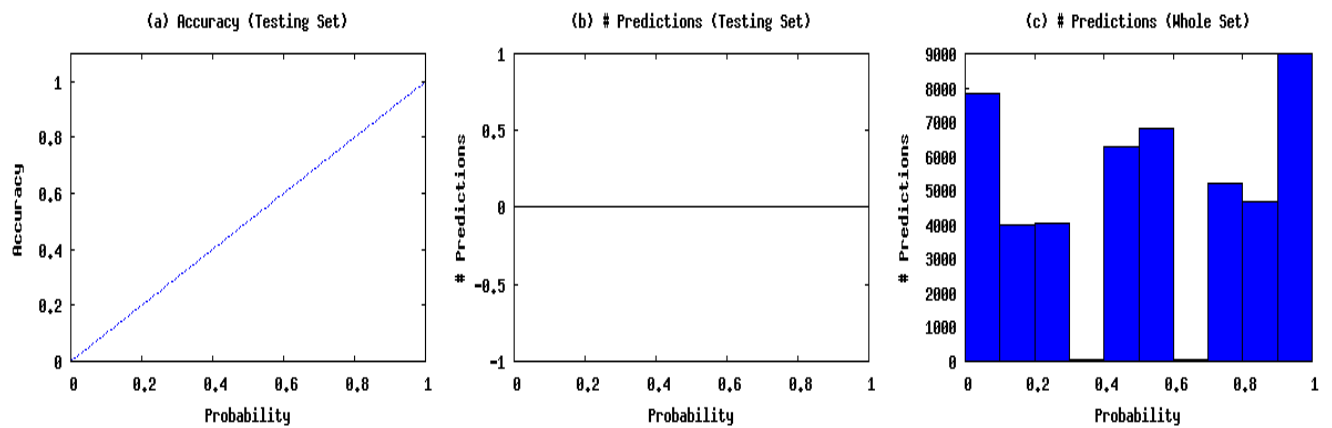


* (a) and (b) are produced using hold-out on evidence variables; (c) also includes all non-evidence variables of the same relation.

B) If person is an extrovert?



The Rosenberg Self-Esteem Scale test



CONCLUSION

The knowledge base to predict human behavior was created with the help of simple statistics obtained from three different datasets in the chosen raw data corpus namely Cattell's 16 Personality Factors Test, Big Five Personality Test, and Rosenberg Self-Esteem Scale tests.

DeepDive took in a set of unstructured (and/or structured) inputs, and populated a relational database table with extracted outputs, along with marginal probabilities for each extraction.

A few difficulties faced were in understanding the documentation of the data management system and getting familiar with using it. Also depending on the questions defined, the data structures used for different datasets changed which was a little hard to debug. For each dataset one question was defined in its inference rules however as part of future work, it can be extended to answer multiple questions. Rules for the same have to be written out separately. As part of future work, the plan is to extend this work by including data from other personality questionnaires used in the raw data corpus selected for this application.

SCOPE:

A system of behavior prediction developed on the foundations of simple statistics has great potential in times to come. Once such a database of thousands of words and phrases is developed, which is called a 'reflection database', it can be used to fine grain over certain important characteristics. For e.g., different databases can be developed for different age groups, and likewise for different regions or countries, cultures, religion, ethnicities, financial background, gender and all such factors which have been proven to contribute to personality.

Also, such databases could be developed for different purposes and the scope of this method could be diversified. For e.g. how behavior is related to facial features, or how behavior is associated with where a person is brought up, i.e. rural vs. urban. Moreover certain psychological puzzles concerning career choices can be resolved like 'are introverts better at science or vice versa?' Also how behavior in childhood is associated with behavior in adulthood as well as how behavior can be used to enhance recruitment process in the corporate sector.

BIBLIOGRAPHY

- 1) Association Rule Mining Technique for Psychometric Personality Testing and Behavior Prediction-Syed,Hamza,Muhammad
<http://www.enggjournals.com/ijet/docs/IJET13-05-05-469.pdf>
- 2) Deepdive Documentation: <http://deepdive.stanford.edu/index.html#documentation>