

**Post Graduate Certificate in Big Data Analytics**

**Final Capstone Project on:**

**Building a Job Recommendation Engine for a career portal.**

**Submitted by:**

**Vishawjeet Deshmukh**

**Ganesh Shinde**

*(BDAP Part Time – Batch 2)*

*Year 2016-17*

**Submission date: 4th, June 2017**

**Declaration**

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Post Graduate Certificate in Big Data Analytics in the S. P. Jain School of Global Management, is entirely our own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfillment of the requirements of that stated above.

Author: Vishawjeet Deshmukh Dated: \_\_4th June, 2017\_\_\_

Ganesh Shinde

**Abstract**

This report implements the recommendation techniques to the problem of predicting jobs that career building portal users will apply to base on their previous applications, demographic information and work history. The process involved in the exploration, preparation, modelling and evaluation of the datasets are described. The analysis of different recommendation methods and implementation of collaborative filtering is provided. The process of relative performance analysis of the proposed classifiers is reviewed. The support of a business objective which will use the predictive capabilities of the proposed models to target users is reviewed including the use of precision to indicate the likely level of jobs to be applied.

**Table of Contents**

Page

[1. Background: 4](#_Toc493380626)

[2. Business Understanding: 8](#_Toc493380627)

[3. Data Understanding: 8](#_Toc493380628)

[4. Data Preparation 11](#_Toc493380629)

[5. Modelling 11](#_Toc493380630)

[6. Evaluation 12](#_Toc493380631)

[7. Conclusion: 12](#_Toc493380632)

# Background:

* Recently job seeking and recruiting websites have been experiencing a striking rise, Also the social media phenomenon has opened up new paths of engagement and revolutionized the exchange of information.
* As the amount of information is growing, a recommendation system is necessary to help match the right candidate with the right job.
* To do so, recommendation techniques such as content-based filtering, collaborative filtering and hybrid approaches can be applied.

**Why Recommendation Systems?**

* **Drive Traffic**

A *recommendation engine* can bring traffic to your site, with personalized email messages and targeted blasts.

* **Deliver Relevant Content:**

By analyzing the customer’s current site usage and his previous browsing history, a recommendationengine can deliver relevant product recommendations as he shops.

* **Reduce Workload and Overhead:**

The volume of data required to create a personal shopping experience for each customer is usually far too large to be managed manually. Using an engine automates this process, easing the workload of your IT staff and your budget

* **Provide Reports**

Providing reports is an integral part of a personalization system. Giving the client accurate and up to the minute reporting allows him to make solid decisions about his site and the direction of a campaign.

**Recommendation Techniques in brief:**



* **Content Based:** recommends items based on a comparison between the content of the items and a user profile.
* **Collaborative:** They exploit behavior of other users and items in terms of transaction history, ratings, selection and purchase information.
* **Model Based CF:** Uses*machine learning* algorithms such as **Bayesian network, clustering,** and **rule-based** approaches to predict the user rating.
* **Memory Based CF:** Memory-based algorithms utilize the entire user-item database to generate a prediction
* **Hybrid Technique:** To make predictions based on a weighted average of the content-based recommendation and the collaborative recommendation

**Job Recommendation Aspects:**

Different job applicant has different characteristics, so to develop well suited Job Recommendation Engine which will suitable to all job applicants is a challenge.

For better/improving JRE, we need to constantly flow through the cycle of following stages.

* **User Profiling -** Understand user characteristics, group them on similarities.
* **Recommendation Strategies - Identify** the suitable recommendation techniques based on available data and user profiles.
* **Recommendation Output -** Generate the recommendation output in form of personalized user profile, dashboards, recommendation emails.
* **User Feedback -** Gather the feedback from users about the recommendations in terms of relevance or usefulness.

**How Job Recommendation is Different than recommending Products or Movies?**

* **Rapid Inventory Growth:**

We aggregate millions of new jobs every day. The set of recommendable jobs is changing constantly.

* **New Users:**

Millions of new job seekers visit job portal every day and begin their job search. We want to be able to generate recommendations with very limited user data.

* **Churn:**

The average lifespan of a job on portal is around 30 days. Content freshness matters a lot because the older the job, the more likely it is to have been filled.

* **Limited Supply:**

One job posting is usually meant to hire one individual. If we over-recommend a job, we could bombard an employer with thousands of applications.

**How do we evaluate Job Recommendation System?**

* **Precision** - Precision is the fraction of the recommendations made that are relevant to the user.

I.e. If recommendation system has made 5 recommendations and 4 out of them are relevant t0 user then precision is 4/5 i.e. 0.8

* **Recall –** Recall is the fraction of recommendations that are successfully made to user.

I.e. If user has applied for 10 jobs and we have recommended 6 out of it then our recall is 6/10 i.e. 0.6

* **Mean Absolute Precision**

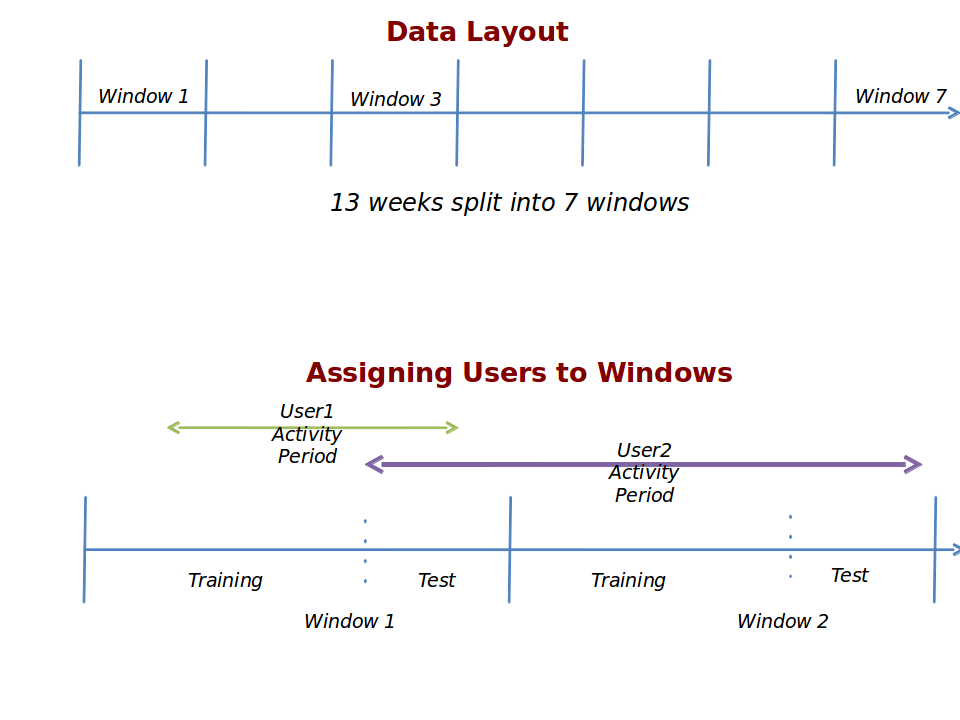
# Business Understanding:

A career building organization (ABC Portal) undertook a Job Recommendation Engine Challenge under Kaggle, which asks you to predict what jobs its users will apply to base on their previous applications, demographic information, and work history. The insights you discover from this data, and the algorithms the winners create, will allow ABC portal to improve its job recommendation algorithm, a core part of its website and a key element in improving user experience.   

# Data Understanding:

# 

In outline, Data is provided on users, job postings, and job applications that users have made to job postings. In total, the applications span 13 weeks. The application data is split into 7 groups, each group representing a 13-day window. Each 13-day window is split into two parts: The first 9 days are the training period, and the last 4 days are the test period.  These splits are illustrated below.



Each user and each job posting is randomly assigned to exactly one window.

Each job is assigned to a window with probability proportional to the time it was live on the site in that window.  Each user is assigned to a window with probability proportional to the number of applications they made to jobs in that window, during that window.  In the above image, User1 only made submissions to jobs in Window 1, and so was assigned to Window 1 with probability 100%.  User2, however, made submissions to jobs in both Window 1 and Window 2, and so may have been assigned to either Window1 or Window2.

In each window, Data is provided for all the job applications that users in that window made to jobs in that window during the 9-day training period. This data can be found in apps.tsv.

In each window, users have been split into two groups, Test and Train. The Test users are those who made 5 or more applications in the 4-day test period, and the Train users are those who did not.

For each window, we need to predict which jobs in that window the Test users applied for during the windows test period.

**File Formats:**

Each of the files is in .tsv (tab-seperated value) format. This means that each line in a .tsv file consists of several fields, which are separated by tabs. To accommodate this file format, fields composed of text have been changed in the following ways to escape tabs, newlines, and carriage returns.

1. Tabs have been replaced by '\t'
2. Newlines have been replaced by '\n'
3. Carriage returns have been replaced by '\r'
4. Backslashes have been replaced by '\\'

**The Files:**

**window\_dates.tsv** contains information about the timing of each window. Each row corresponds to a window, and has the date and time that the training period begins, that the training period ends, and that the test period ends.

users.tsv contains information about the users. Each row of this file describes a user. The UserID column contains a user's unique id number, the WindowID column contains which of the 7 windows the user is assigned to, and the Split column tells whether the user is in the Train group or the Test group. The remaining columns contain demographic and professional information about the users.

**test\_users.tsv** contains a list of the Test UserIDs and windows, for your convenience. All of the information in this file can be found in users.tsv.

**user\_history.tsv** contains information about a user's work history. Each row of this file describes a job that a user held. The UserID, WindowID, and Split columns have the same meaning as before. The JobTitle column represents the title of the job, and the Sequence column represents the order in which the user held that job, with smaller numbers indicating more recent jobs.

**jobs.tsv** contains information about job postings. Each row of this file describes a job post. The JobID column contains the job posting's unique id number, and the WindowID column contains which of the 7 windows the job was assigned to. The other columns contain information about the job posting. Two of these columns deserve special attention, the StartDate and EndDate columns. These columns indicate the period in which this job posting was visible on careerbuilder.com. Each job was visible for part of its 13-day window, but not necessarily for the entire 13 days. Users can only apply to a job between its StartDate and EndDate, so don't predict that a user applied for a job if the job was not visible for at least part of the 4-day Test period.

**splitjobs.zip** is a directory containing jobs1.tsv, jobs2.tsv, ... , jobs7.tsv, each of which contain all jobs in a given window. Thus, for example, jobs3.tsv contains all jobs in Window 3. This directory contains the exact same information as jobs.tsv, in the same format, and is provided merely for your convenience.

**apps.tsv** contains information about applications made by users to jobs. Each row describes an application. The UserID, WindowID, Split, and JobID columns have the same meanings as above, and the ApplicationDate column indicates the date and time at which UserID applied to JobId.

**popular\_jobs.py** is the python code used to generate the popular jobs benchmark.

**popular\_jobs.csv** is the benchmark submission file produced by popular\_jobs.py.

# Data Preparation

Performing user to user memory based filtering required extracting features that measured the similarity between the users.

# Modelling

* User to User memory based and Job to Job memory based output were produced using standard collaborative filtering techniques as described.
* Parameters were selected using a small test set of 2009 users. These users were real test users but we used Day 9 for each
* Test users with no apps in training day 9 were ignored.
* To produce an actual measure of similarity between two users, extracted features of user were combined into single number.
* A simple user to user collaborative filtering method was not producing the great precision.
* To improve the performance of the model i.e. to improve the MAP (Mean absolute Precision), Clustering algorithm was applied within user data set and users within same cluster were identified.
* After clustering the Users, User-User Collaborative filtering technique was used, which improved the precision of the results.

# Evaluation

* Before applying cluster algorithm:

Mean Absolute Precision: MAP = 0.0.

Precision with plain User-User Collaborative was not great.

Clustering applied to user dataset and then UU Collaborative filtering was applied which yield improved results.

* After applying cluster algorithm:

Mean Absolute Precision: MAP =0.2

# Conclusion:

# 

A job portal provides an efficient search for online information on job vacancies for jobseekers. The main goal of this project is to attempt to select the right graduates based on industry needs. However, it is important that be aware the job web portals can never fulfill all the problems of jobless graduates. it is focused on improving online job portal and try to reduce problems that existed in existing job systems by developing better portal.

Project focused on improving the online job portals and tried to reduce problems that are encountered in existing systems by developing a knowledge system that also acts as and job portal. Thus this portal can be more beneficial with further to the services and the features.

The Advantages of the new job portal are as follows:

1. Achieve the main targets of the project

2. Standard content services and display

3. High level management and flexibility