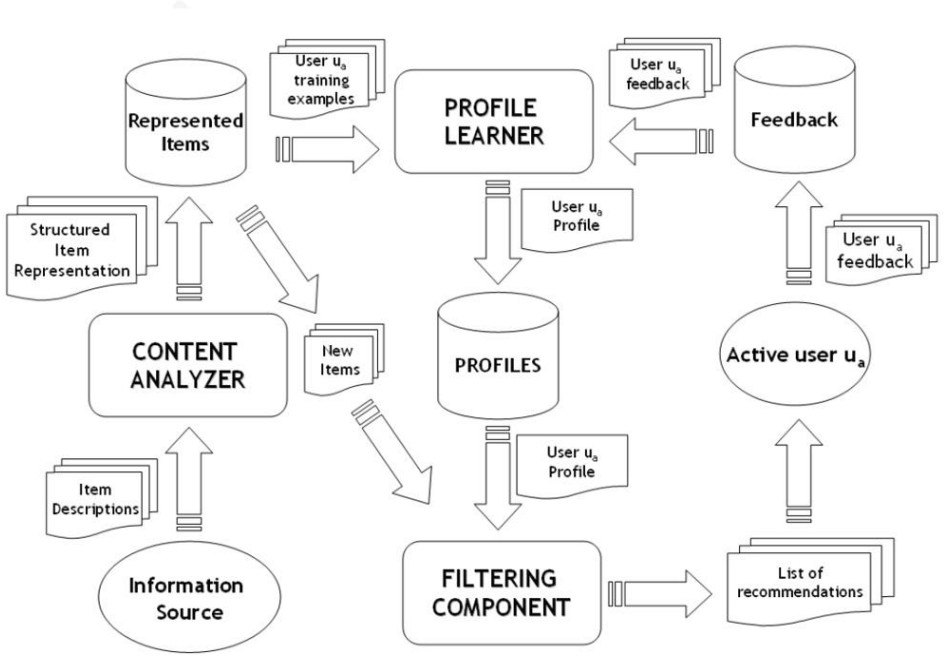
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 05
Lecture - 22

Lecture 22: Feature Engineering – I

Hello everyone. Today we are going to continue our discussion on the content based Recommender System. And in this regard last class we introduced the topic and in today's lecture we will start talking about Feature Engineering. So, this is the concepts which is going to be covered. We will be talking about feature extraction and that too in the context of text feature. Typically, the news items which are recommended in a news recommender system like that of Google news use such kind of text features.

So, if you remember last class we were talking in in fact, in the very first week and last class we were you have you must have seen this particular figure. And here it shows a high level architecture for content based recommendation system. The first activity here is to collect the data from some information source and do content analysis. Through this content analysis process what do we generate? We generate a structured representation of the item.



So, our discussion today is around this particular topic. To start with let us see what feature engineering is. It is the process of transforming the raw data into features. So, that we can represent it in terms of certain numerical values, we can which can better serve

the purpose of input to the predictive model which we are going to use. And we should select the features in such a manner so, that the prediction accuracy is high for on scene data.

So, basically under feature engineering three important terminologies are discussed. First one is feature extraction which we are going to cover today. So, it is the process of finding the features from the raw data and that raw data can be anything in the sense it can be text, it can be audio, it can be image or any other format which may not be directly usable by the prediction model that we are going to use. So, once we extract the features the next task is find out the best set of features that make the prediction accuracy better. So, this is a kind of subset selection problem.

Subset selection in the cell we have a set of features out of that a small part we are going to select and on what basis we are going to select that small part. That small part has to be determined based on certain criteria like it should be giving higher prediction accuracy, but how do we know that it is going to give higher prediction accuracy. One approach could be we keep on using subsets to the prediction algorithm and try finding out what is the best feature set best subset of the features. Now, the question is ah before we talk about what are the mechanism let us try to understand why do we select a subset of features higher the data more is supposed to be the accuracy, but when we talk about the data the data set at hand is here we have the items here we have the features. Now, if the number of features are very high and compared to the number of items or the number of objects what is going to happen what is wrong going to happen that we must first understand.

Why do we need to select this features because we are going to relate it to some kind of response variable and this response variable together in which the features try making one prediction model. So, you have a prediction model where the input will be the features and output will be the response variable and you would like to decide on these model parameters. Once you decide this model parameters and fix this then whenever a new item comes a new object comes whose features are already known then those features you can provide as input and get this response variable. Now, the problem is when you determine these model parameters you have to solve some kind of least square problem and sometimes this least square problem is also associated with certain additional terms to avoid over pitting. So, as a result you have very large number of parameters which is to be learned.

Now, when you have to learn very large number of parameters corresponding data set which is basically the set of items should be very large. How large it is think of a simple solving some let us say from linear equations. Now, if you solve linear equations with n variables what is the minimum number of minimum number of equations you need you need n number of equations otherwise you cannot extract the values of this variable. So,

same principle is also applied here. If you have at least same number probably you will be able to find out some reasonable value, but the problem here is little different.

We are trying to solve certain optimization problem which is a quadratic equation mostly it is a because you are trying to minimize the root mean square error and as this and we are trying to find the best fit to it. So, it which means not only that for n variables we need n equations for the best fit we mean as many. So, experimentally you can say this should be pretty high compared to the number of parameters. So, typically experimentally they say we observe that it is at least 15-20 times more than the number of variables. So, by chance if you get for example, in cursor text document you get let us say 1000 number of features 1000 because text documents when you take word level or phrase level features they can easily get to some 1000 number of features or more than 1000 thousands of features.

So, now, to that 1000s of features how many data points you have each how do you construct this these are the item feature and this response you collect from one individual user right. So, for a individual user how many ratings in his lifetime he gives 200, 300 not even that much even it is much much more lower than these numbers. So, you do not have enough number of items to rightly determine these values of course, if you give some input you will be getting some output it is a kind of you garbage in and garbage out that is one issue. So, therefore, you have to to ensure the correctness of the algorithms that you use for error minimization for getting the right value of the parameters you should be selecting less number of features this is one aspect. Second aspect is the computational complexity as more number of features you have computational complexity is going to be high.

The third problem you are going to have is this feature set may not be complete there may be many sparsity within this. So, in that case the features which defines well and have less sparsity can be selected, but to this end we have another important terminology in case of feature extraction that is dimensionality reduction. So, it is about transforming the original set of features to a lower dimensional space and in this regard we have already studied about PCA and that is principal component analysis and singular value decomposition. Now, today ah we will now I mean the in this series we are not going to focus much on this, but still try to remember when the feature set would be very sparse probably you have to use little bit other version of PCA and SVD. So, which can also take care of the sparsity in case of SVD you already have seen how to take care of sparsity by ah first of all considering the UV decomposition followed by learning this UV decomposition from the available ah available ratings.

So, it was a kind of matrix completion problem. Moreover, you have to remember that this feature engineering is an iterative process at one go you cannot do it probably you have to try a number of times before deciding the right number of features. Moreover,

there is no formal guideline which is universally applicable to every problem. So, it depends on your data and you have to really try a number of times you have to do adopt the engineering principles that is why it is called feature engineering. So, you have to adopt engineering principles to decide the features.

So, we will be talking now about the feature extraction which is one part of feature engineering. So, what is feature extraction? It is the process of transforming the raw data into certain numeric form. So, that you it is usable by the target algorithm for further processing. So, all the algorithms and some of these algorithms were introduced you in one of the ah early little early lectures where you talked about we talked about various supervised and unsupervised algorithms. So, the when we discuss about them the basic assumption is those algorithms use numeric data as the feature of the ah as the input feature.

So, therefore, it is very important that we transform this data into numeric form which is otherwise unstructured. And when we do so, we should ensure that the features that we are extracting is actually is representative of the information content in that particular document. Here we are talking about document because mostly recommender system literature, content based recommender systems focus mostly on ah mostly on text kind of data like news recommender etcetera. Now, as I told you mostly this requires a lot of domain knowledge and a lot of manual effort is required. But in today's context it is becoming quite automated specifically by adopting deep learning approaches, but we will try we we are now focus mostly on the traditional approaches.

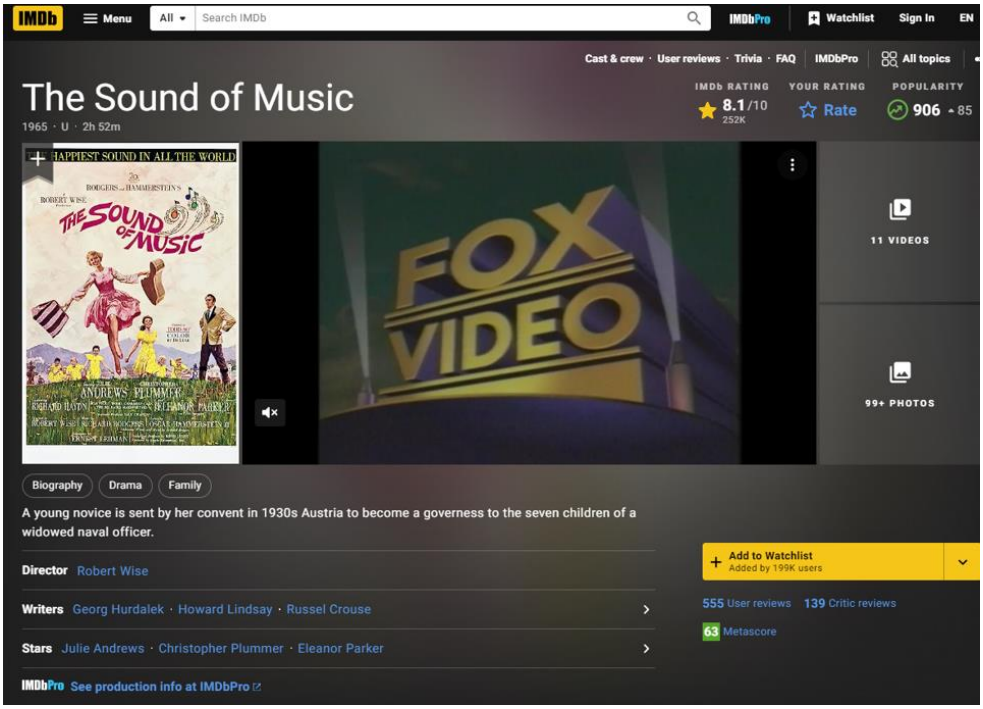
It is highly specific. So, for news recommender system it can be news articles or web pages which is text. Then for music recommendation it can be some track features which there is something called a music genome project. So, the features which specified there those things are supposed to be extracted from the music, but we as I told you we will be talking focusing on this text. So, now we talk about item representation for text documents. So, this items that can be recommended to the user are represented by a set of features called the attributes or the properties.

Specifically, in a movie recommender system the features could be the something which is which is explicitly mentioned or extrinsically mentioned like genre, actor, director and so on. These are subjective these these are quite structured data. However, many times you have to collect this from the related text documents like in case of movie there will be certain short description of the movie, there can be there can be user reviews and so on. Now, in case of feature extraction in under such setting mostly we target traditional keyword based profile besides that relationship among the words can be considered to make to have some kind of semantic analysis based approach where these lexicons or and ontologies those things play a key role. But we are not going to address this part because

our approach is just to introduce you to the topic of feature extraction, but let me tell here right now that we are really not going to cover everything about text analytics.

So, briefly we will be giving an overview and this should be used as an guideline to go for advanced methods for feature representation and extraction feature extraction and representation. Look at this is what I was telling this is I got from IMDb website. Now, look here many of the things are mentioned what is its rating ok, what is this some kind of popularity score, who is the director, writers, other stars and its genre and it also shows there are many user reviews. There is a short description here as well ok and there are many user reviews. So, this I have just take a screenshot of a first review and maybe a portion of the second review.

So, these reviews let us say there are 155 reviews. So, 155 reviews it is not possible for you to manually read all of them and extract features and this is not the only movie. There will be many such movies even under this genre suppose you try to categorize wise categorize them in terms of genre. So, around these genres also there will be hundreds of movies and assume that in each of these movies there are around 500 reviews and one such description. So, there has to be certain ways to get the text features that best represents the content description as well as the reviews.



Suppose this is our M, your M can be different maybe you will be happy with this extrinsic features, but this may not be the case as well. So, moving ahead let us look at how to convert the text data into structured data in the form of a structured data. So, what is a structured data here unstructured data here this is the unstructured data. What is the

structured data here? For example, here we can write in the form of certain table who are the director, who are the writer. So, for all these features we can have values.

So, this very structured representation and once they because they are of course, these are text values, but even if they are the text values we know that they are some kind of nominal attribute and how to deal with nominal attributes also we know right. Similarly, here is something let us say star rating this is a this is in which scale this is in ratio scale ok. So, such kind of combination of data which is in different scales can also come in, but right now we our focus is only on this kind of text data ok. So, moving ahead let us see how to convert this text data. Huge amount of preprocessing is required to convert text data.

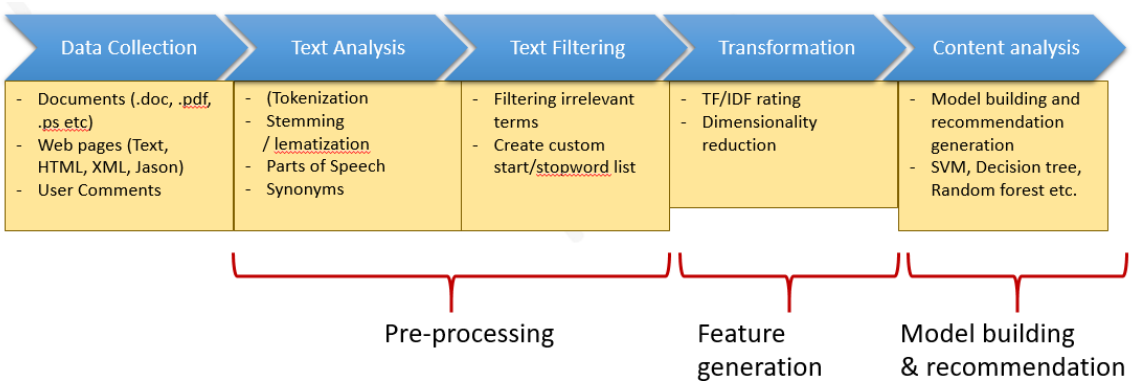
So, we have to first remove some of the things which usually users give and they do not have any grammatical ah grammatical sense or syntactical sense. For example, there can be misplaced misspelled words, there can be some strings where user is trying to provide some kind of emphasis ah, but do is it a valid word this is not not many users will be using it. Similarly, somebody might be using emotions in form of emojis. So, those also unless otherwise you make specific provision to take care of them they are supposed to be removed.

Then next axis you have to tokenize. When you tokenize the data a particular sentence let us say a huge amount of preprocessing is required to convert a text. So, this is a sentence. So, this sentence you have to now split into small small parts. This basic small part is a word. So, by tokenization you will be making them into words besides that you will be removing punctuations etcetera and if you necessary you may convert everything to upper or lower case, but look here also while doing so, you should be little careful because sometimes upper case letters specifically somewhere might be giving some kind of syntactical meaning.

Then you can also split the sentences you can identify multi word expressions which basically mean one thing some phrases connected words named entities names and so on. You can also think of adding linguistic information by considering the parts of speech analysis a considering whether it is a noun verb adjective and so on. Now after you filter the non significant or irrelevant words sorry you also have to filter non-significant and irrelevant words. So, those non significant and irrelevant words are called stop words ok. So, this stop words are those which are very frequently used and really do not are very generic enough.

So, that they can they do not have any distinguishing ah power. So, you have to remove them then you may have to come down to the basic root word for example, token, tokenization, tokenizing all these things go to the root word root word token. Sometimes you can go for certain other ah techniques where you consider more than one words of

course, here also you consider more than one word, but there can be n grams considering any 3, 2, 2 grams by by grams, try grams and so on. So, these are the typical steps in content based recommender system for text document. What are the steps? Data collection, what is the data collection? You collect the let us say you are trying to build one movie recommender system.



What is the source from which will be collecting the data? You have rating matrix let us say from movie lens and you would like to get item details from let us say IMDB. So, the from IMDB when you try collecting this data what will you access? You will access this IMDB web page if they provide you access with ah some API or something you can download. If they do not provide it then probably you have to write down a crawler and remove ah the tag tags HTML tags and all and make a file. So, data collection then text analysis which consists of tokenization, stemming, parts of speech analysis and synonyms analysis finding the synonyms if any, then filtering irrelevant terms like stop word etcetera, then TF ah transforming the data into numeric figures.

So, that can be done using TF IDF rating. Then you have finally, you have to analyze this content. How will you analyze the content? You will be analyzing the content by building various classification or regression models. So, while you do so, you create the data set and data set consists of the features and the ah features and the response variable. So, with these steps in the next class we are going to talk about each of this in a much more elaborate manner. So, these are the references which I have used here and I am going to use in the upcoming lectures.

Here we have discussed that we have to extract the features from a text document in the form of tokens and this token will be representing this tokens are the terms will be representing the features. We call them the dictionary based on which will be now creating the numeric data. And each row in this data set will be representing in a document. A document in the cell ah the text features associated with the text associated with the item and each column will be representing the features. With this we finish this lecture. Thank you.