UNIVERSITÉ DE NANTES

MASTER THESIS

Learning representation of movies through multimodal networks

Author:
Ankit LAMBA

Supervisor:
Dr. Hoel Le CAPITAINE

A thesis submitted in fulfillment of the requirements for the degree of M2 Data Science

in the

LS2N, CNRS
Polytech Nantes

July 2, 2021

Declaration of Authorship

I, Ankit LAMBA, declare that this thesis titled, "Learning representation of movies through multimodal networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"A single neuron in the brain is an incredibly complex machine that even today we don't understand. A single 'neuron' in a neural network is an incredibly simple mathematical function that captures a minuscule fraction of the complexity of a biological neuron."

Andrew Ng

UNIVERSITÉ DE NANTES

Abstract

Hoel LE CAPITAINE
Polytech Nantes

M2 Data Science

Learning representation of movies through multimodal networks

by Ankit LAMBA

This document discuss about the multimodality, multimodal learning , multimodal deep learning, various algorithms and networks that can used to learn the multimodal data, some approaches to multimodal deep learning. About some multimodal datasets that can be used to learn a multimodal network. Some exploratory analysis on the Datasets. Further about the movie ratings prediction using multimodal data using machine learning regression models, and multimodal network. Hence about the the results carried out, and conclusions made. Also about the way all the work done in vizualization using gantt diagram. One can find the usefull references in the end...

Acknowledgements

I want to acknowledge and very thankful to my internship supervisor Dr. Hoel Le CAPITAINE to give me very useful advises, guide lines that helped me in the project as well as in the the report...

Contents

List of Figures

List of Tables

List of Abbreviations

Acronym	Extended meaning
MMTF	**M**ultifaceted **M**ovie **T**railer **F**eature
MLP	**M**ulti **L**ayer **P**erceptron
SMR	**S**ocial-Aware **M**ovie **R**ecommendation
RBM	**R**estricted **B**oltzmann **M**achine
CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort **T**erm **M**emory
SOTA	**S**tate **O**f **T**he **A**rt

Dedicated to family, and my school(Polytech Nantes)...

Chapter 1

Introduction

Today, given the volume and the variety of the collected data, standard learning algorithms that only consider of the same type, i.e. a flat table data are not satisfactory anymore. One must consider algorithms that take into account different modalities of the data. For instance, in movie recommendation systems, we would like to consider not only the grades or ratings given by the users to the movies, but also the characteristic of the movies (and not only the meta-data). In this internship, we specifically considered the MMTF-14k dataset , that includes 13,623 Hollywood-type movies described by audio and visual features, and ranked by more than 138,000 users. Here, we would like to be able to give to the user a sort of interactive map of the movies, so that he can navigate within this world of movies, choosing his next view according to the visual proximity of the movies. Additionally, we also have other movie related datasets, such as the Cornell Movie Dialog corpus, in which dialogs of different movies are available. The way actors are exchanging is also a very interesting characteristic of movies. In practice, the code will be developed in Python, using the libraries Tensorflow/Keras for deep approaches, and scikit-learn for usual machine learning, matplotlib for visualization, and NumPy/Panda for data manipulation.

1.1 Objectives

1. Study the state-of-the-art of multimodal deep learning,
2. The second step of the project is to design a research prototype allowing to visually inspect the proximity of movies, given their visual and auditory characteristics, as well as their rating distribution among the users.
3. Finally, being able to use dialogs data as an additional information, in order to increase the quality of the model.

Chapter 2

State of the art of Multimodal Deep Learning

2.1 Introduction

Previously, the machine learning and deep learning models just used the single modality of a type of data to learn the models. But as day by day we have multiple modalities of a type of data in different types of modalities. If we can make use of them then we enhance the performance and efficiency of the models. There are various terminologies and approaches to performing multimodal learning whether its machine learning or deep learning few of them are discussed further below.

2.1.1 Multimodality

Multimodality is the way to make use of more than one semiotic mode to make meaning, to communicate, and to represent something in general, or in a specific situation. Its an application for one medium with multiple literacies. Data or information can be represented in different modalities or using different modalities. For example, A movie can be described or represented using different modalities such as its visual descriptions, audio descriptions, movie dialogues, and the multiple ratings given by multiple users. Another example, during a televised weather forecast written language, spoken language, weather-specific language, sign-language, geography, and symbols are involved. In real-world our experience is multimodal — we see objects, hear sounds, feel the texture, smell odors, and taste flavors on daily basis, and finally come up with a decision. when a number of our senses — visual, auditory, kinesthetic — are being engaged in the processing of information, we understand and remember more suggested by Multimodal learning. Learners can combine/fuse information from different sources.

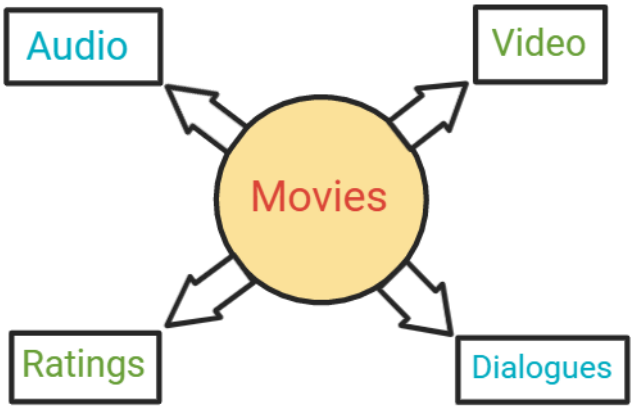

FIGURE 2.1: Representation of Movies using multimodalities.

2.1.2 Multimodal Learning

Multimodal machine learning is a vibrant multi-disciplinary research field which addresses some of the original goals of artificial intelligence by integrating and modeling multiple communicative modalities, including linguistic, acoustic and visual messages Morency and Baltrušaitis, 2017. The information in the real world usually comes in different modalities. For example, images are usually associated with tags and text explanations; texts contain images to more clearly express the main idea of the article. Different modalities are characterized by very different statistical properties. For instance, images are usually represented as pixel intensities or outputs of feature extractors, while texts are represented as discrete word count vectors. Due to the distinct statistical properties of different information resources, it is very important to discover the relationship between different modalities. Multimodal learning is a good model to represent the joint representations of different modalities. The multimodal learning model is also capable to fill missing modalities given the observed ones.

2.2 Multimodel Deep Learning

Deep networks have been successfully applied to unsupervised feature learning for single modalities (e.g., text, images, or audio). In the paper, a novel application of deep networks to learn features over multiple modalities proposed. It presents a series of tasks for multimodal learning and shows how to train deep networks that learn features to address these tasks. In particular, It is demonstrating cross-modality feature learning, where better features for one modality (e.g., video) can be learned if multiple modalities (e.g., audio and video) are present at feature learning time. Furthermore, It shows how to learn a shared representation between modalities and evaluate it on a unique task, where the classifier is trained with audio-only data but tested with video-only data and vice-versa. The models are validated on the CUAVE and AVLetters datasets on audio-visual speech classification, demonstrating best published visual speech classification on AVLetters and effective shared representation learning. Multimodal learning involves relating information from multiple sources of the same information. Ngiam et al., 2011.

For example, images and 3-d depth scans are correlated at first-order as depth discontinuities often manifest as strong edges in images. Conversely, in speech recognition audio and visual data have correlations at a "mid-level", as phonemes and visemes (lip pose and motions); it can be difficult to make relations between the raw pixels to audio waveforms or the spectrograms.

Humans are very known to integrate or to combine audio and visual information in order to understand speech. First it was exemplified in the McGurk effect (McGurk MacDonald, 1976) where a visual $/ga/$ with a voiced $/ba/$ is perceived as $/da/$.

Particularly, it was taken into consideration that there can be three learning settings – multimodal fusion, cross-modality learning, and shared representation learning. Data from

	Feature Learning	Supervised Training	Testing
Classic Deep Learning	Audio	Audio	Audio
	Video	Video	Video
Multimodal Fusion	A + V	A + V	A + V
Cross Modality Learning	A + V	Video	Video
	A + V	Audio	Audio
Shared Representation Learning	A + V	Audio	Video
	A + V	Video	Audio

FIGURE 2.2: Multimodal Learning settings where A+V refers to Audio and Video Ngiam et al., 2011.

all modalities are available at all phases(feature learning, training and testing etc.) in the multimodal fusion setting.

In cross-modality learning, data from multiple modalities is available only during feature learning.

A shared representation learning setting, is unique in that different modalities of information are presented for supervised training and testing. For feature learning the most straightforward approach is to train an RBM model separately for both audio and video information. After learning the RBM, the posteriors of the hidden variables given visible variables can be used afterwards as a new representation for the data.

A direct approach is to train an RBM over the concatenated audio and video data for training of a multimodal model. The RBM is an undirected graphical model with hidden variables (h) and visible variables (v). There are symmetric connections between the hidden and visible variables ($W_{i,j}$), but no connections within hidden variables or visible variables. The model defines a probability distribution over h, v (Equation 2.1). This configuration makes it easy to compute the conditional probability distributions when v or h is fixed (Equation 2.2) particularly.

$$-\log P(v,h) \propto E(v,h) = \frac{1}{2\sigma^2} v^T v - \frac{1}{\sigma^2}(c^T v + b^T h + h^T W v) \tag{2.1}$$

$$p(h_j|v) = sigmoid(\frac{1}{\sigma^2}(b_j + w_j^T v)) \tag{2.2}$$

In the cross-modality learning experiments, if we can learn better representations for one modality (e.g., audio or video) when given multiple modalities (e.g., audio and video) during feature learning the evaluation is done. To get the correlations among the modalities and to perform inference a deep autoencoder was used to resolve both the issues.

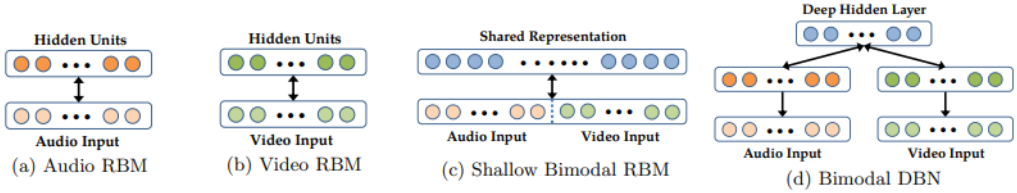

FIGURE 2.3: RBM Pretraining Models. The trained RBMs for (a) audio and (b) video separately as a baseline. The shallow model (c) is limited and we find that this model is unable to capture correlations across the modalities. The bimodal deep belief network (DBN) model (d) is trained in a greedy layer-wise fashion by first training models (a) & (b). Ngiam et al., 2011.

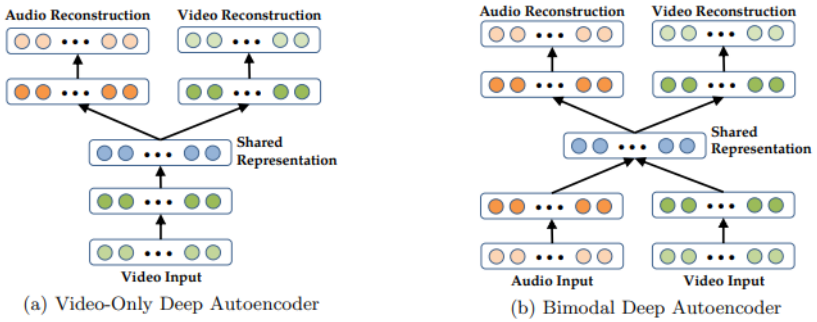

FIGURE 2.4: Deep Autoencoder Models. A "video-only" model is shown in (a) where the model learns to reconstruct both modalities given only video as the input. A similar model can be drawn for the "audio-only" setting. We train the (b) bimodal deep autoencoder in a denoising fashion, using an augmented dataset with examples that require the network to reconstruct both modalities given only one. Both models are pre-trained using sparse RBMs (Figure 2.3). Ngiam et al., 2011.

So the bimodal deep autoencoder used did not perform as well as the video-only deep autoencoder: while the video-only autoencoder learns only video features (which are also good for audio reconstruction), the bimodal autoencoder learns audio-only, video-only, and invariant features. As such, the feature set learned by the bimodal autoencoder might not be optimal when the task at hand has only visual input. Since the models are able to learn multimodal features that go beyond simply concatenating the audio and visual features, it was proposed to combine the audio features with the multimodal features. When the best audio features are concatenated with the bimodal features, it outperforms the other feature combinations. This shows that the learned multimodal features are better able to complement the audio features. A shared representation can be learned over audio and video speech data, During supervised training, the algorithm is provided data solely from one modality (e.g., audio) and later tested only on the other modality (e.g., video). One approach to learning a shared representation is to find transformations for the modalities that maximize correlations.

2.3 Social-Aware Movie Recommendation via Multimodal Network Learning

With the rapid development of the Internet movie industry, Social-aware Movie Recommendation systems (SMRs) have become a famous online web service that provides relevant movie recommendations to users Zhao et al., 2017. In this effort, many existing movie recommendation approaches learn a user ranking model from user feedback with respect to the movie's content. Unfortunately, this approach suffers from the sparsity problem inherent in SMR data. In the presented work, the sparsity problem by learning a multimodal network representation for ranking movie recommendations was addressed. There was the development of a heterogeneous SMR network for movie recommendations that exploits the textual description and movie-poster image of each movie, as well as user ratings and social relationships. With this multimodal data, a heterogeneous information network learning framework called SMR-MNRL for movie recommendation was presented. To learn a ranking metric from the heterogeneous information network a multimodal neural network model was also developed. The evaluation of this model is done on a large-scale dataset from a real-world social-aware movie recommender Web site, and it was found that SMR-MNRL achieves better performance than other state-of-the-art solutions to the problem.

With the rapid proliferation of online social networks and the rise of streaming movie services, people are changing their movie selection habits. A movie watcher, in turn, can then recommend a movie or show to their friends via social media. As a result, Social-aware Movie Recommendation systems (SMRs) have been developed to provide relevant movie recommendations to a target audience.

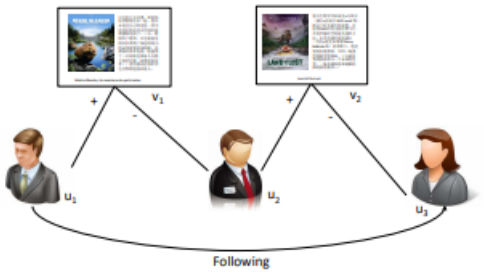

FIGURE 2.5: Multimodal Heterogeneous SMR Network for Movie Ranking.
Zhao et al., 2017

The basic task of movie recommendation is to predict which movies that a specific user may enjoy watching. The primary assumption shared by many existing movie recommendation methods is that relevant movies can be found by learning a user ranking model that is

trained based on a hand-crafted representation of multimodal movie content (e.g., a movie's description and images) and from user feedback. Movie descriptions are typically variable length text, so it is natural to employ deep recurrent neural networks to learn their semantic representation. Additionally, visual movie contents like movie posters, trailers, and scene photos also convey powerful information that provides a better understanding of a movie's story.

Convolutional neural networks have shown promising results for learning a transformation of an image into a high-level semantic representation, but a similar transformation of the video is more difficult. So a movie's poster is a visual summary of the movie. To learn a joint movie representation from text and images, one can employ a multimodal neural network framework that fuses, which is able to infer a model from different modalities.

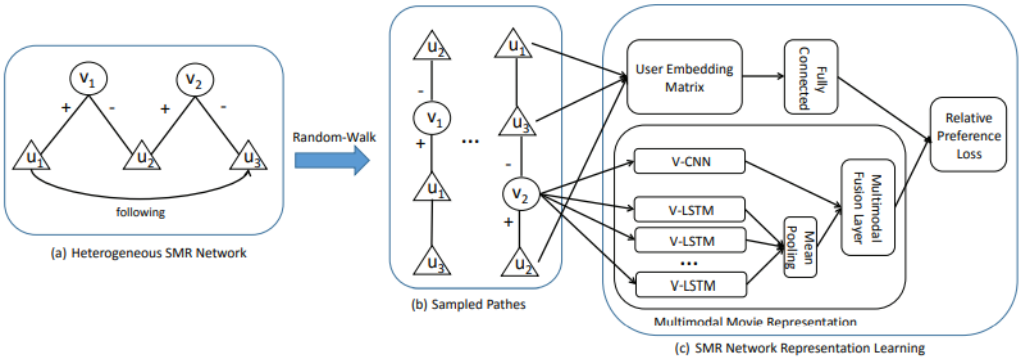

FIGURE 2.6: The Framework of the Multimodal Network Representation Learning for Social-aware Movie Recommendation (SMR-MNRL). (a) The heterogeneous SMR network is constructed by integrating multimodal movie contents, users' relative preference, and their social relationships in SMR Web sites. (b) Data paths are sampled by a random walker over the heterogeneous SMR network. (c) The representation of multimodal movie contents and user ranking model in the SMR network is encoded into fixed feature vectors for recommendation based on a relative preference loss model. Zhao et al., 2017

In the presented work, the problem of social aware movie recommendation in social media from the viewpoint of learning a multimodal heterogeneous network representation for ranking was considered. Firstly a heterogeneous multimodal SMR network that exploits multimodal movie contents, user feedback, and the social network to provide movie recommendations was introduced. Afterwards there was an introduction about a multimodal neural network with two sub-networks: (1) a recurrent neural network that learns from the textual representation of a movie's description, and (2) a convolutional neural network that learns a visual representation from a movie's poster. Then the shared movie representation using a multimodal fusion layer that combines two neural networks was learned. Next, the multimodal neural network with the heterogeneous SMR network to create a unified Multimodal Network Representation Learning (MNRL) framework for social-aware movie recommendation was intergrated. Finally, there was development of a random-walk-based learning method with the introduced multimodal neural networks to learn the representation of movie contents and user ranking model in the proposed SMR network, such that the learned multimodal ranking metric is implicitly embedded in the heterogeneous network representation for the recommendation.

Movie recommendation plays an important role in delivering videos to users. The related ranking oriented movie recommendation methods, which are mainly based on the text information of movie contents was reviewed. Unlike previous studies, It was formulated that the problem of social-aware movie recommendation from the viewpoint of learning multimodal heterogenous network representation for ranking, which can be solved via random-walk based learning method with multimodal neural networks.

Deep learning models have shown great potential for learning effective multimodal representations. The objective of multimodal ranking metric learning in the defined problem is

different from the objective of the deep multimodal representation learning methods.

2.4 A Deep Multimodal Approach for Cold-start Music Recommendation

An increasing amount of digital music is being published daily. Music streaming services often ingest all available music, but this poses a challenge: how to recommend new artists for which prior knowledge is scarce? In the presented work, the aim was to address this so-called cold-start problem by combining text and audio information with user feedback data using deep network architectures Oramas et al., 2017. The method is divided into three steps. First, artist embeddings are learned from biographies by combining semantics, text features, and aggregated usage data. Second, track embeddings are learned from the audio signal and available feedback data. Finally, artist and track embeddings are combined in a multimodal network. Results suggest that both splitting the recommendation problem between feature levels (i.e., artist metadata and audio track), and merging feature embeddings in a multimodal approach improve the accuracy of the recommendations.

Recommender systems can be broadly classified into collaborative filtering (CF), content-based, and hybrid methods. Directly training all the layers of a deep network together makes it difficult to exploit all the extra modeling power of a deeper architecture. Therefore, it was decided to separate the problem of music recommendation into artist and song levels.

Artist feature embeddings are learned from artist metadata in an artist recommendation scenario. Track feature embeddings are learned from audio signals in a song recommendation scenario. In both cases, a hybrid recommendation approach is used based on learning attribute-to-feature mappings. Both feature embeddings are combined in a multimodal network to predict song recommendations of cold-start artists.

To produce cold-start music recommendations, the following framework was proposed. Given the set of artist features A_s of a song s, and the set of the track features T_s of s, the complete feature set of s is defined as the aggregation of its artist and track features $F_s = A_s T_s$. There are several approaches in the literature for multimodal feature learning and late fusion of multimodal feature vectors. In the presented work, audio and text feature vectors are learned separately and then combined via late fusion in a multimodal network. Given the

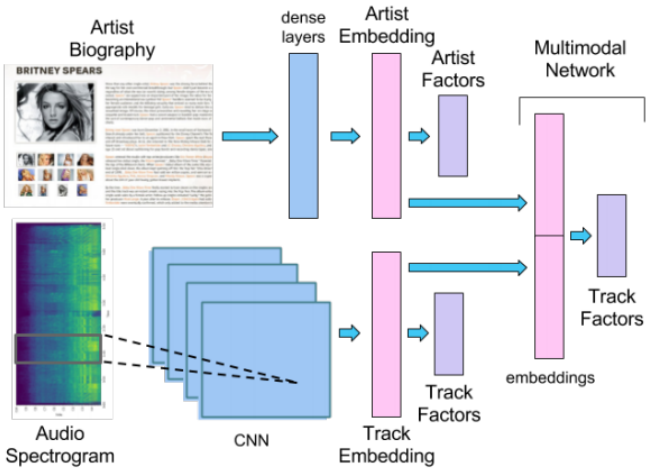

FIGURE 2.7: Model architecture Oramas et al., 2017

different nature of the artist and track embeddings, a normalization step is necessary. Normalized feature vectors are then fed to a feed-forward neural network (a simple Multi-Layer Perceptron, MLP). Two different architectures were explored: (i) each embedding vector is connected to an isolated dense layer of 512 hidden units with ReLU activations after a process of batch normalization. Then, both dense layers are connected to the output layer. The rationale behind this is that the isolated dense layers help the network learn non-linearities

from each modality separately. (ii) each embedding vector is l2-normed and then concatenated into a single feature vector which is directly connected to the output layer, resulting in a linear model.

Regularization is obtained by applying dropout with an empirically selected factor of 70% after the input layer for both architectures. In this work, a multimodal approach for song recommendation has been presented. The approach is divided into three steps. (1) Artist feature embeddings are learned from the text and semantic features in an artist recommendation scenario using a deep network architecture. (2) Track feature embeddings are learned from the audio spectrograms using convolutional neural networks. (3) Embeddings are combined in a multimodal network.

It is shown how a multimodal approach, based on the late fusion of track and artist feature embeddings that are learned separately, outperforms end-to-end multimodal approaches where the different modalities are learned simultaneously. Moreover, results have shown that our multimodal approach achieves better results than pure text or audio approaches.

2.5 Discussion

The above-mentioned approaches to multimodal deep learning or to learn a multimodal network are very different in nature, technique, also the datasets used in the different approaches are very different in nature, size and format. In "Multimodal Deep learning" Ngiam et al., 2011 the two modalities of the data were Audio-Video there were three learning settings like Multimodal Fusion of Data, cross-modality learning, shared representation learning but it is not the case in two more approaches. Also, the RBMs and Bimodal Deep Autoencoder were used for learning. In the "Social-Aware Movie Recommendation via Multimodal Network Learning" Zhao et al., 2017 a heterogeneous multimodal network was developed to rank the movies to recommend them using real-time data which contains movie descriptions, movie posters, movie ratings, user feedback, social relations of the movies. Word embeddings, CNNs(Convolutional Neural Networks), RNNs(Recurrent Neural Networks) LSTMs(Long-Short term Memories) Network were used to learn the multimodal network. The joint representation of all the modalities to represent the data was multimodal fusion layer that was also used to combine the two different neural networks. In "A Deep Multimodal Approach for Cold-start Music Recommendation" Oramas et al., 2017 the text and audio information along with user feedback used as data to learn deep network architecture. So two feature embeddings(Artist and Track) were created using the various modalities of data then these two features embeddings were combined into one to learn a multimodal network. The late fusion mechanism was used to combine the feature embeddings into one embedding. Deep Neural Networks and Convolutional Neural Networks were used.

In all the different approaches the results were also different as there were different data representations, different learning Networks, and architectures. There can be more methodologies, approaches, representation of multimodal data is possible to learn the multimodal Network. In all the approaches the feature learning of different modalities of data is done using some sort of machine learning, deep learning network, or architectures further those features are whether jointly represented by concatenating before learning the final multimodal network or they were combined or being fused inside the multimodal network using concatenation.

Chapter 3

Datasets

3.1 Multifaceted Movie Trailer Feature dataset(MMTF-14K)

The dataset consists of 13,623 Hollywood-type movie trailers, ranked by 138,492 users, generating a total of almost 12.5 million ratings Deldjoo et al., 2018. To address a broader community, metadata, audio, and visual descriptors are also pre-computed and provided along with several baseline benchmarking results for uni-modal and multi-modal recommendation systems. This creates a rich collection of data for benchmarking results and which supports future development of this field. The dataset is organized in 4 folders:

3.1.1 Data

The Data folder contains files with general information regarding the dataset. Here one can find information regarding user ratings, the titles of the movies we used and preferred download method, etc. The "rating.txt" file gives general information regarding the way the movie ratings were obtained, along with a download link for the corresponding movie rating files. The "movie_description.csv" file has general information regarding the movie files we used: movieId (an id shared across the system, for identifying movies trailers and their corresponding features and ratings), title (the title of the movie trailer), and finally YTId (the youtube link for each individual movie trailer). The "itemids_splitted_5foldCV" contains information regarding the 5 folds we used in our experiments (see Benchmark), including the train and test split we created for each of them, thus creating 10 files - 5 folds/files for train, 5 folds/files. Each of these files contains the movieId and userId information that can be further used to replicate our experiments exactly.

The description of the movies is below:

	movieId
count	13623.000000
mean	60973.805329
std	46634.419179
min	89.000000
25%	5154.500000
50%	78974.000000
75%	102994.000000
max	131262.000000

TABLE 3.1: Description of the movies.

3.1.2 Metadata descriptors

The Metadata descriptors folder contains three CSV files associated with each of the three provided features: genre features, tag features and year of production.

3.1.3 Audio descriptors

The Audio descriptors folder contains two sub-folders: Block level features and i-vector features. While the BLF data includes the raw features of the 6 subcomponents and similarities computed thereon, the i-vector features include different parameters for the Gaussian mixture model (GMM), total variability dimension (tvDim), and the folds information used for creating the feature vector. The BLF folder has two subfolders: "All" and "Component6"; the former contains the similarities computed using all 6 subcomponents, the latter contains the raw feature vectors of the subcomponents in separate CSV files. The i-vector features folder contains individual CSV files for each of the possible combinations of the three parameters, fold, gmm and tvDim.

3.1.4 Video descriptors

The Visual descriptors folder contains two subfolders: Aesthetic features and AlexNet features, each of them including different aggregation and fusion schemes for the two types of visual features. These two features are aggregated by using four basic statistical methods, each included in a different subfolder, that computes a video-level feature vector from frame-level vectors by using: average value across all frames (denoted "Avg"), average value, and variance ("AvgVar"), median values ("Med") and finally median and median average distribution ("MedMad"). Each of the four aggregation subfolders of the Aesthetic features folder contains CSV files for three types of fusion methods: early fusion of all the components (denoted All), early fusion of components according to their type (color based components denoted Type3Color, object-based components - Type3Object and texture - Type3Texture) and finally each of the 26 individual components with no early fusion scheme (example: the colorfulness component denoted Feat26Colorfulness), therefore generating a total of 30 files in each subfolder. Regarding the AlexNet features, In our context, we use the extracted output values of the fc7 layer, and therefore no supplementary early fusion scheme is required or possible, and therefore only one CSV file is present inside each of the four aggregation folders.

3.2 MovieLens dataset(20M)

MovieLens 20M movie ratings. Stable benchmark dataset. 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users "Abbas, Khushnood (2017), "Movielens 20M Dataset", Mendeley Data, V3". Includes tag genome data with 12 million relevance scores across 1,100 tags. The description of the average ratings given by users for the movies is below which are obtained by merging MMTF-14k and Movielens datasets using movieIds corresponding to each movie.

	movieId	rating
count	13623.000000	13623.000000
mean	60973.805329	3.109704
std	46634.419179	0.676707
min	89.000000	0.500000
25%	5154.500000	2.763158
50%	78974.000000	3.200000
75%	102994.000000	3.544242
max	131262.000000	5.000000

TABLE 3.2: Description of the mean ratings corresponding to movies.

3.3 Cornell Movie-Dialog Corpus

A large metadata-rich collection of fictional conversations extracted from raw movie scripts. (220,579 conversational exchanges between 10,292 pairs of movie characters in 617 movies) Danescu-Niculescu-Mizil and Lee, 2011 .

3.4 Data Preprocessing

After acquiring all the datasets we need to perform some sort of processing on it so that we can make it useful to use it for the further progress that is to feed it to the regression models to predict the movie ratings. we have used Pandas and Numpy to prepare the data to feed to the regression models and multimodal network regression model.

3.4.1 Final Movie Ratings

We have merged the MMTF14k and MovieLens Dataset using movieIds to obtain the final mean ratings to get exactly ratings equal to a number of movies in the MMTF14K dataset.

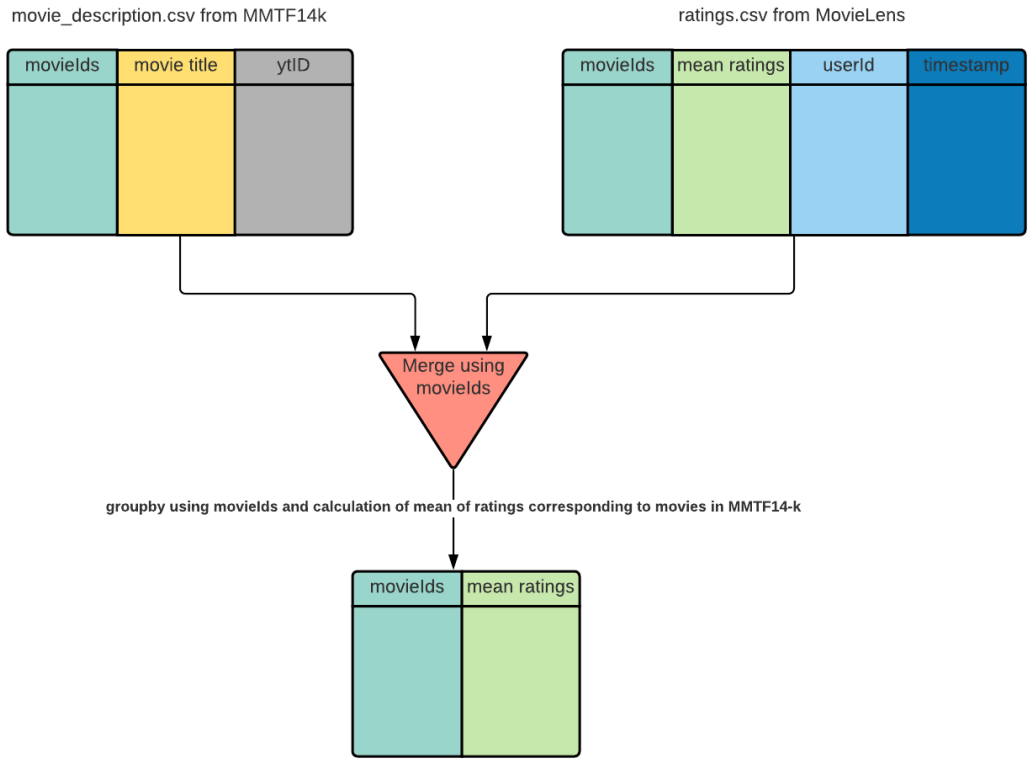

FIGURE 3.1: Final Movie ratings by Merging MMTF14k and MovieLens datasets.

3.4.2 Final Multimodal Features

We are using ivectors from the audio features and AvgVar Aesthetic from the visual features of the MMTF14K dataset, also for dialogs features we need to use a Doc2Vec model to obtain the dialog features to enhance the quality of the regression models.

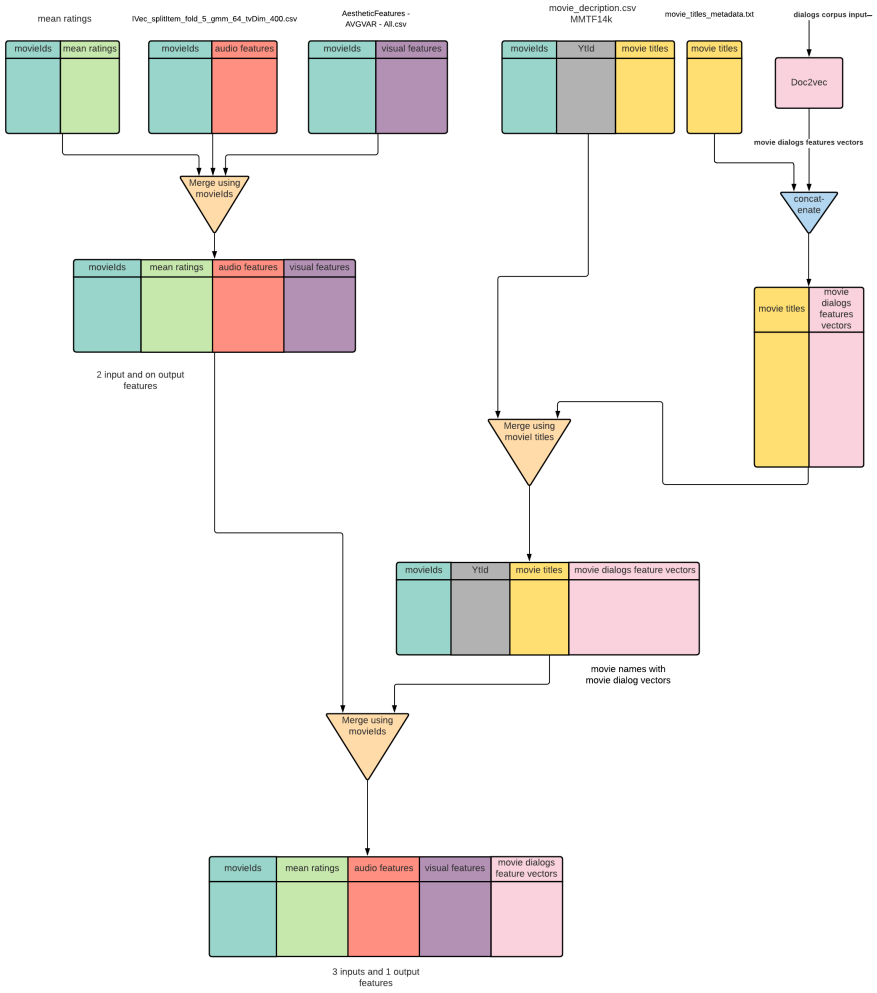

FIGURE 3.2: Final Movie ratings by Merging MMTF14k and MovieLens datasets.

Afterward, we are using movieIds to merge the mean ratings obtained previously with the audio and visual movie features, to merge the generated dialog vectors with the other features we needed to concatenate these dialog vectors to the movies, then we will use movie title to merge these dialog vectors with the other movie features. So finally we will use this data to feed it to the regression models.

3.4.3 Multimodal Features for state of the art Model

we will separately feed all the movie features and the movie rating to the multimodal network as the inputs and outputs.

3.5 Exploratory Data Analysis on the Datasets

We have explored the various datasets mentioned above and got the insights after analyzing them about their various properties, and statistics about them.

3.5.1 Genres of the Movies.

We have used the GenresFeatures.csv file from the MMTF-14K Datasets to get the information about Movie's Genres which contains the genres of the movies corresponding to the movies. The bar graph in Figure 3.1 below gives us an idea about the movie genre like what kind of genres does the movies from the datasets are, also it tells us about which genre have the most number of the movie here we can say the Drama movies are highest in number and Film_Noir genre have the least movies. Overall there are 18 genres in total.

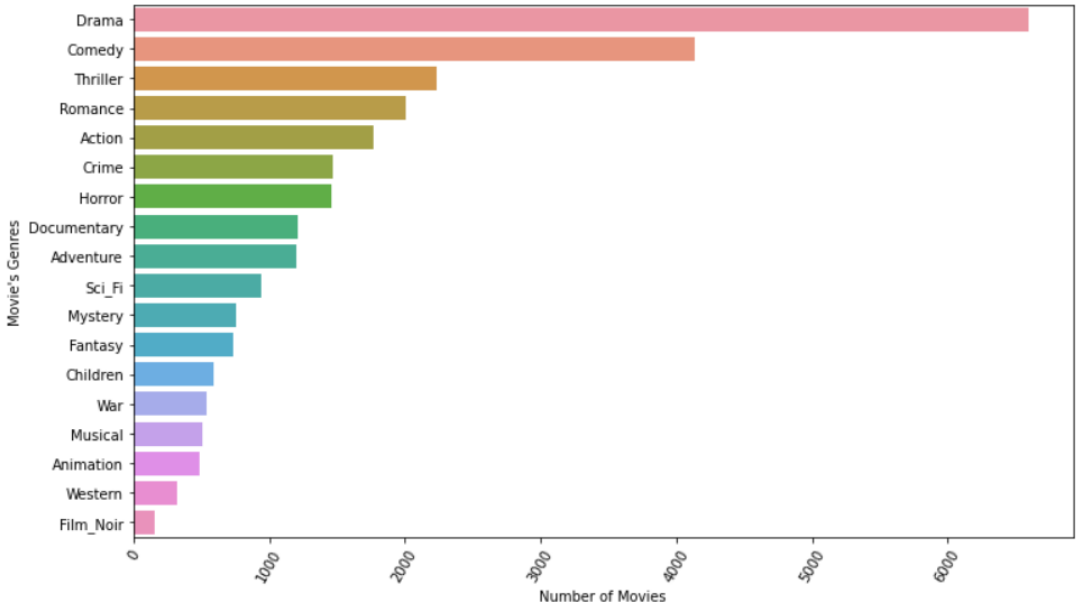

FIGURE 3.3: Visualization of Movie's Genres with respect to Number of movies per genre.

3.5.2 Years of Productions.

Using the YearOfProd.csv file from the MMTF-14K Datasets we got the information about the years in which the movies were produced or released. The bar graph in Figure 3.3 and pie-chart in Figure 3.3 describes the number of movies being produce per year from the visualization below we can say that the most number of movies are produced after the year 2000 and the highest in the year 2013. Before 2000 there is the least number of movies being produced and the lowest in the years of the 19th century.

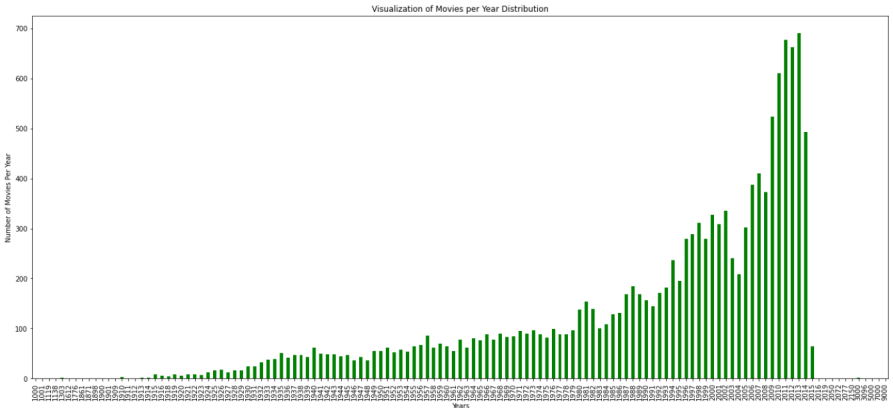

FIGURE 3.4: Visualization of the years the movies released with respect to Number of movies per year.

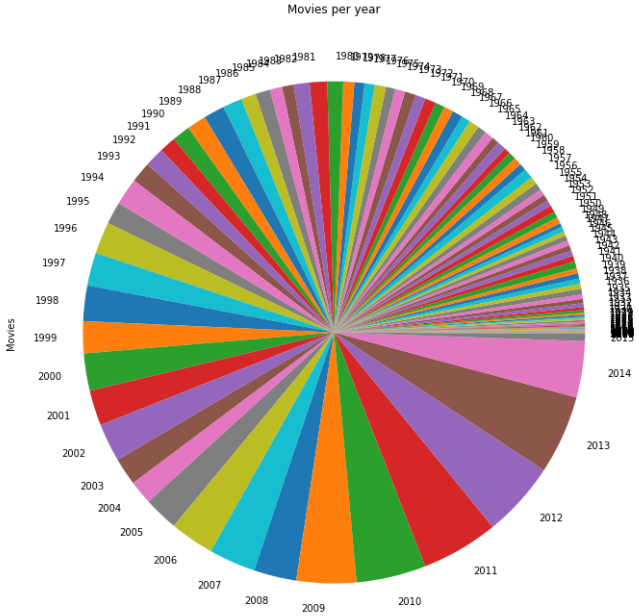

FIGURE 3.5: Visualization of Movie's Genres with respect to Number of movies per genre.

3.5.3 Movie Tags.

The final tags are obtained by merging the tags.csv from the Movielens and movie_descriptions.csv from the MMTF-14K datasets. We have generated a Word cloud to represent some of the most frequent tags in the movies. The WordCloud in Figure 3.4 is the visualization of movie tags.

3.5.4 Movie Ratings.

The final ratings are obtained by merging the movie_descriptions.csv file from MMTF-14K dataset and the ratings.csv file from the MovieLens dataset using the movieIds. The final ratings are the average/mean ratings given by the users corresponding to the movies.

Figure 3.4 is the Visualization of the Number of Movies per rating from it we can observe that the highest number of movies have a rating = 3.0, there is no movie having a rating = 0.0, and most of the number of movies have the rating between 3.0 to 4.0. Also, very few movies have the highest rating which is 5.0.

FIGURE 3.6: Visualization of movie tags.

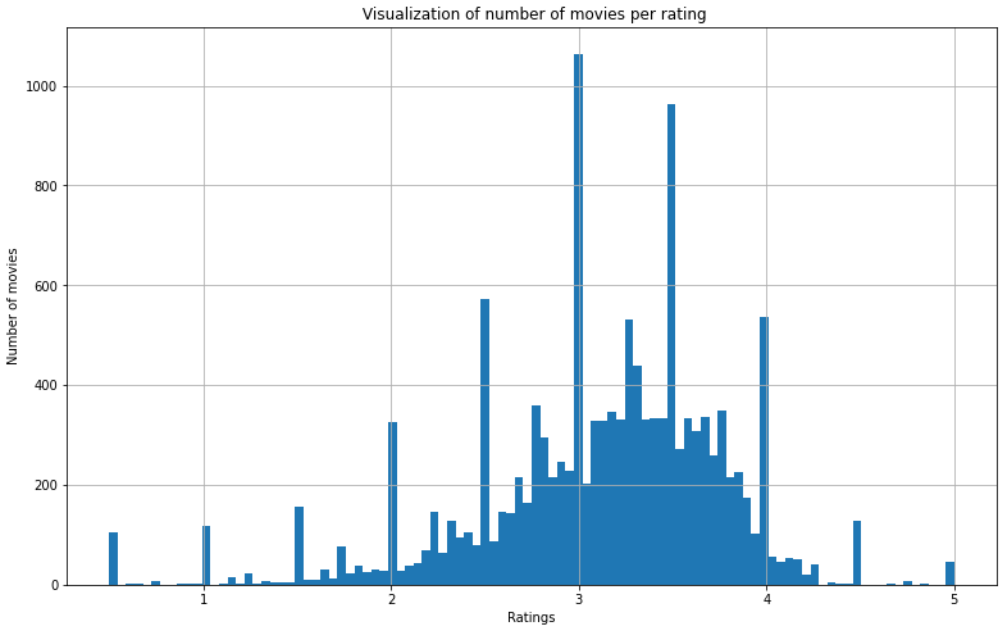

FIGURE 3.7: Visualization of Number of Movies per Rating.

Chapter 4

Movie Rating Prediction using Multimodal Data

We are going to predict the movie rating using the movie's visual, audio features from the MMTF-14k datasets and dialogs features from the Cornell movies dataset. The Ratings are mean ratings obtained by merging the MMTF-14K and the MovieLens datasets. Firstly, we merge the audio features and visual features with the obtained movie ratings using the movieIds then use some regression model to predict the movie ratings using the visual and audio features then we are computing the Regression metrics do visualization of bar graph with error for the same. Afterward, we will generate vectors of the dialogs using *Doc2Vec*(), *TaggedDocument*() from *gensim.models.doc2vec*() with epochs or iterations=100 vector size=318 as we are left with 318 movie after merging the datasets by using dialogs data from the Cornell movie dataset to use them as additional information to enhance the quality of the models. We are going to use the movie names to merge the dialogs dataset with the other two datasets and finally again predict the movie ratings using the audio, visual, and dialogs features. Hence we will compute again the regression metrics too and draw the bar graph with error to make the comparison among the regression models. The following Figures 4.1, 4.2, 4.3, and 4.4 show the whole procedure or the prototype for the prediction of movie ratings using multimodal data by regression models.

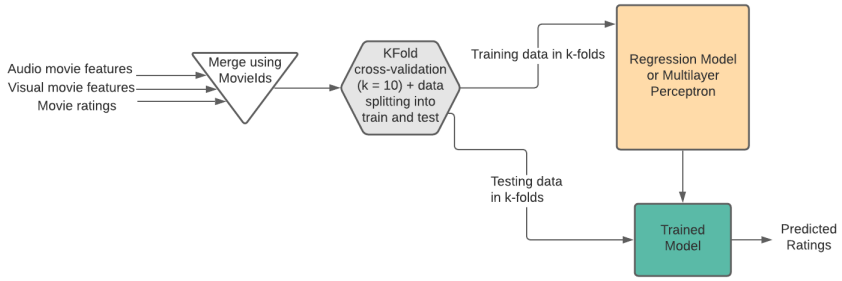

FIGURE 4.1: Block diagram or prototype for movie rating prediction by regression model using Audio and Visual movie Features.

We have created a *regression_model*() function which takes input the regression models from 1-9(regression models from scikit-learn and Keras), multimodal features and predict the movie ratings as output, do kfold cross-validation(*cross_val_score*() from scikit-learn with scoring='neg_mean_squared_error') and get mean and std scores. and data splitting into 10 folds using Kfold from scikit-learn, and training and testing on each fold thus calculate the MAE and MSE for each fold using *mean_absolute_error*() and *mean_squared_error*() from scikit-learn. Afterward, we are taking mean values and std values for each model from one to nine. For multimodal network regression, we are doing data splitting into 10 fold using Kfold from scikit-learn the and Keras to implement a multimodal network for movie rating prediction thus, same calculation of MAE, MSE for each fold and taking mean, and std values so that we can make comparison among the models. The *cross_val_score*() method is not applicable on multimodal network regression model.

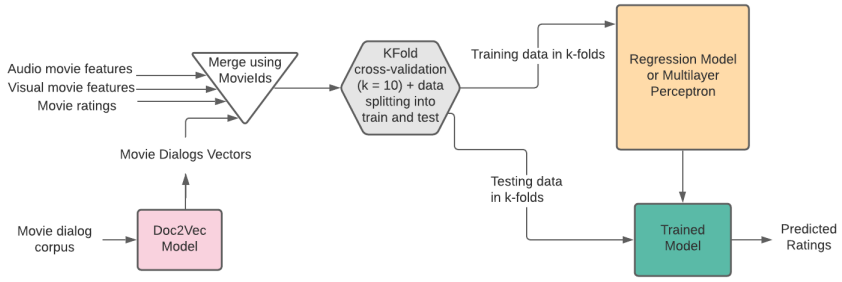

FIGURE 4.2: Block diagram or prototype for movie rating prediction by regression model using Audio and Visual, Dialogs movie Features.

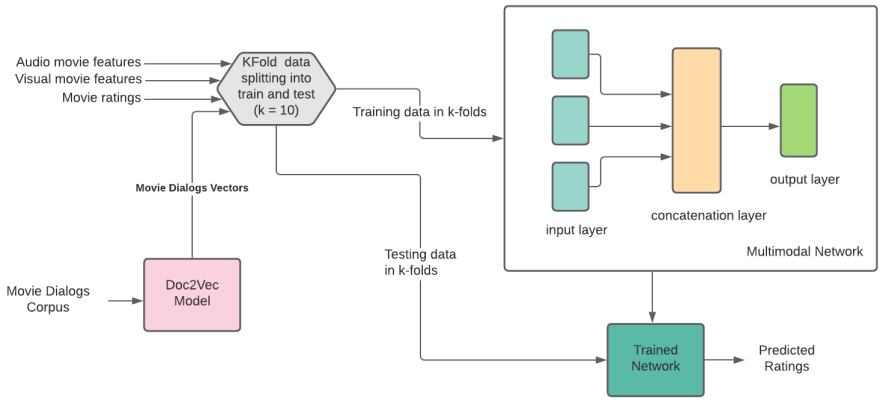

FIGURE 4.3: Block diagram or prototype for movie rating prediction by Multimodal network using Audio and Visual, Dialogs movie Features.

4.1 Regression Models

4.1.1 Linear Regression

Linear regression is a linear model, e.g. a model that makes an assumption of a linear relationship between the input variables (x) and the single output variable (y). More specifically, that the output variable y can be calculated from input variables (x) using their linear combinations. For instance, a simple regression problem (one x and one y) can be defined, the form of the model would be:

$$y = B0 + B1 * x \tag{4.1}$$

The representation is a linear equation that is the combination of a specific set of input values (x) the solution to which is the predicted output for that given set of input values (y). As such, both the input values (x) and the output value (y) are numeric. We are using $LinearRegression()$ from sklearn.linear_model to predict movie ratings using linear regression model.

4.1.2 XGBRegressor

XGBoost stands for "Extreme Gradient Boosting" and it's an implementation of gradient boosting trees algorithm. It is a popular supervised machine learning model having characteristics such as parallelization, computation speed, and performance.

We are using $XGBRegressor()$ from xgboost to predict movie ratings using XGB regression model.

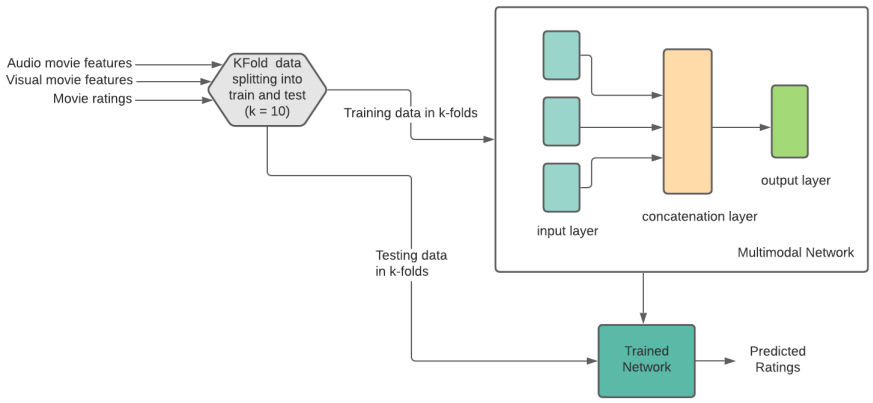

FIGURE 4.4: Block diagram or prototype for movie rating prediction by Multimodel network using Audio and Visual, Dialogs movie Features.

4.1.3 Decision Tree Regressor

In statistics, data mining, and machine learning the Decision tree learning or induction of decision trees is one of the predictive modeling approaches to be used. A decision tree used to go from observative values about an item to conclusive values about the item's target value. We are using $DecisionTreeRegressor()$ from sklearn.tree to predict movie ratings using the decision tree regression model.

4.1.4 KNeighbors Regressor

It is a Regression based on the k-nearest neighbors approach. So the target value is predicted by local interpolation of the targets values that are associated with the nearest neighbors inside the training set. We are using $KNeighborsRegressor()$ from sklearn.neighbors to predict movie ratings using the KNeighbors regression model.

4.1.5 SVR Regressor

The Support Vector Regression (SVR) uses principles as same as the SVM for classification, with few minor changes only. Firstly, as the output is a real number it becomes very difficult to make the prediction of the information very handy, which has many possible values. In regression, a margin of a lit bit of tolerance (epsilon) is being set in approximation to the SVM. We are using $SVR()$ from sklearn.svm to predict movie ratings using SVR regression model.

4.1.6 Lasso Regression

The Lasso regression is a form of linear regression that uses shrinkage. Shrinkage is where the data values are being shrunk towards a central point, just like the mean. The lasso procedure encourages very simply, and sparse models (models with fewer parameters). We are using $lasso()$ from sklearn.linear_model to predict movie ratings using the Lasso regression model.

4.1.7 Ridge Regressor

This model solves a regression model where the linear least squares function is the loss function, and l2-norm is the regularization. Also known as Ridge Regression or Tikhonov regularization. This estimator has been built-in to support the multi-variate regression (when y is a 2d-array of shape (n_samples, n_targets)). We are using $Ridge()$ from sklearn.linear_model to predict movie ratings using the Ridge regression model.

4.1.8 MLPRegressor

MLPRegressor() from sklearn.neural_network implements a multi-layer perceptron (MLP) that trains using the backpropagation without the activation function in the output layer. Therefore, the loss function is the square error, and the output is a set of real continuous values.

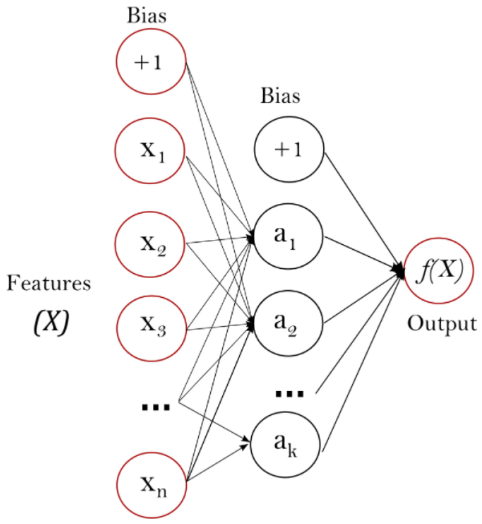

FIGURE 4.5: MLP with one hidden layer.

The Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(.) : R^m \rightarrow R^o$ by training on a dataset, where the number of dimensions for input is m and o is the number of dimensions for output or the predicted values. Given the set of features $X : [x_1, x_2.....x_m]$ and the target y , it can learn a non-linear function approximator for regression. It differs from logistic regression, it has hidden layers in between the input and the output layer, there can be one or more non-linear hidden layers.

In Figure 4.5, the leftmost layer, also known as the input layer, consists of a set of neurons $x_i|x_1, x_2....x_m$ representing the input features or input values. Each neuron in the hidden layer transforms the values from the previous layer with some weighted linear summation $w_1x_1 + w_2x_2 + ... + w_mx_m$, that is followed by a non-linear activation function. The output layer receives values from the hidden layer in the and then transforms them into output values or predicted values.

```
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(304, 152), learning_rate='constant',
             learning_rate_init=0.5, max_fun=15000, max_iter=200, momentum=0.9,
             n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
             random_state=None, shuffle=True, solver='adam', tol=0.0001,
             validation_fraction=0.1, verbose=False, warm_start=False)
```

FIGURE 4.6: MLPRegressor summary.

4.1.9 KerasRegressor

We can easily fit the regression data with Keras's sequential model and predict the test data. Keras Regressor is scikit-learn regressor API for the Keras library. Firstly, we have to build a sequential model in Keras that is a multilayer neural network implementation using the Keras after that we can make use of Keras Regressor to wrap it up so that we can make the predictions. The Figure 4.7 and 4.8 give the kerasregressor model summary.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 608)               370272
_________________________________________________________________
dense_5 (Dense)              (None, 304)               185136
_________________________________________________________________
dense_6 (Dense)              (None, 152)               46360
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 153
=================================================================
Total params: 601,921
Trainable params: 601,921
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 4.7: Summary of Keras sequential model with KerasRegressor when we are using audio, and visual movie features.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 609)               371490
_________________________________________________________________
dense_1 (Dense)              (None, 304)               185440
_________________________________________________________________
dense_2 (Dense)              (None, 152)               46360
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 153
=================================================================
Total params: 603,443
Trainable params: 603,443
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 4.8: Summary of Keras sequential model with KerasRegressor when we are using audio, visual, and dialogs movie features..

4.1.10 Multimodal network regression model based on SOTA

The multimodal network regression model based on SOTA is implemented using the Keras funtional_API(). The Keras functional API provides a more flexible way for defining models. It specifically allows you to define multiple input or output models as well as models that share layers. More than that, it allows you to define ad hoc acyclic network graphs. As per the SOTA the raw data of different modalities are feed into different types of networks or architectures like CNN, RNN, RBM, LSTM, MLP, autoencoders etc. for feature learning after this a multimodal network like MLP is used to combine, fusion, or to concatenate these features of different modalities into single feature space to learn all the predicted values in the multimodal network. So in our case we have all the different modalities features therefore we are concatenating the different modalities using the concatenate() layer of the keras to predict the movie ratings and taking all different modalities features as seperate inputs in the input layer and movie ratings predictions as output in output layer. Also we are training the Multimodal network regression model with 10 folds for each fold epochs=100, and batch size=500. The following Figures 4.9, 4.10, 4.11, 4.12 show the summary and information of the shape of the multimodal networks which learns to predict the movie ratings.

```
Model: "model_1"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
audio_features_input (InputLaye [(None, 400)]         0
_________________________________________________________________________________________________
visual_features_input (InputLay [(None, 208)]         0
_________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 608)           0           audio_features_input[0][0]
                                                                  visual_features_input[0][0]
_________________________________________________________________________________________________
movie_ratings_predictions_outpu (None, 1)             609         concatenate_1[0][0]
=================================================================================================
Total params: 609
Trainable params: 609
Non-trainable params: 0
_________________________________________________________________________________________________
```

FIGURE 4.9: Summary of Multimodal Network regression model based on SOTA using Audio and visual movie features.

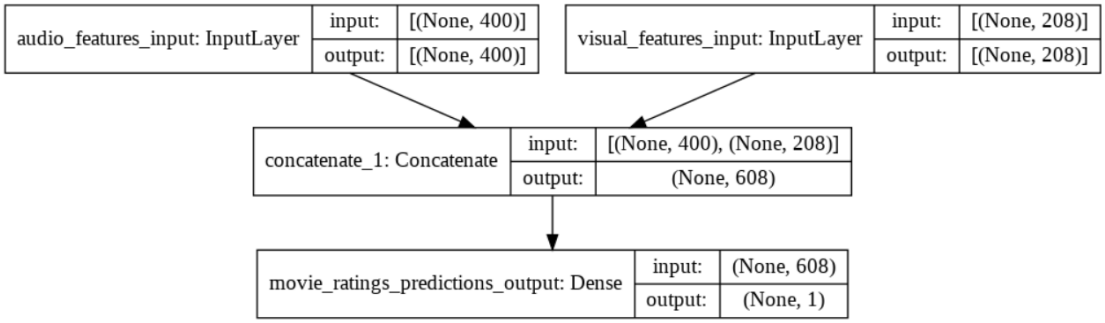

FIGURE 4.10: Shape information Multimodal Network regression model based on SOTA using Audio and visual movie features. .

```
Model: "model_2"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
audio_features_input (InputLaye [(None, 400)]         0
_________________________________________________________________________________________________
visual_features_input (InputLay [(None, 208)]         0
_________________________________________________________________________________________________
dialogs_features_input (InputLa [(None, 1)]           0
_________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 609)           0           audio_features_input[0][0]
                                                                  visual_features_input[0][0]
                                                                  dialogs_features_input[0][0]
_________________________________________________________________________________________________
movie_ratings_predictions_outpu (None, 1)             610         concatenate_2[0][0]
=================================================================================================
Total params: 610
Trainable params: 610
Non-trainable params: 0
_________________________________________________________________________________________________
```

FIGURE 4.11: Summary of Multimodal Network regression model based on SOTA using Audio, visual, and dialogs movie features.

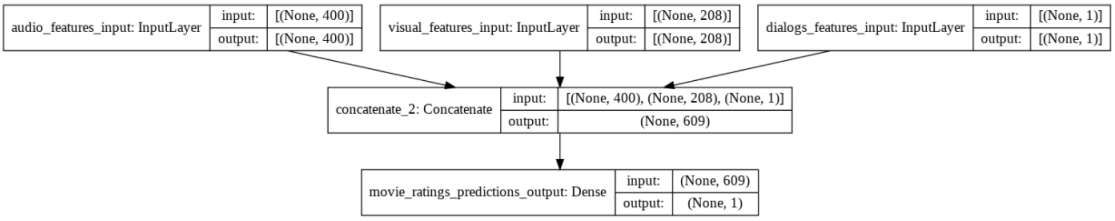

FIGURE 4.12: Shape information of Multimodal Network regression model based on SOTA using Audio, visual, and dialogs movie features.

4.2 Cross validation and Data splitting

We are using the KFold from scikit-learn for cross-validation and splitting of data. It provides train/test indices to split data into train/test sets. we can split the dataset into k consecutive folds. So in our case, we are using k(n_splitts)-10 therefore, we have 10 folds for the cross-validation in that we are using *cross_val_score*() from scikit-learn and scoring = neg_mean_squared_error to compute the mean and standard deviation of scores for the models except the model based on the state of the art for which it is not applicable. Also, for the splitting of the train, test data, we are using the Kfold with 10 folds for each fold of test data we are calculating the regression metrics MSE(Mean Squared Error) and MAE(Mean Absolute Error) then we are taking their mean and standard deviation values of all the 10 folds of all the models so that we are making use of those values to compare and distinguish all the regression models.

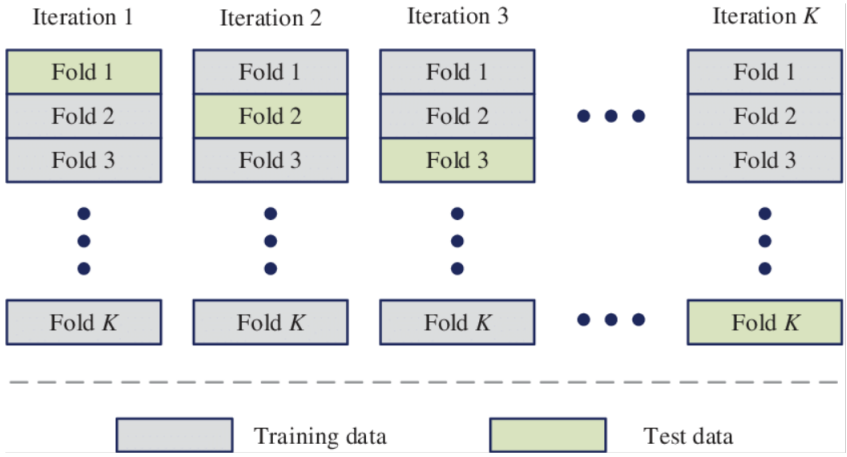

FIGURE 4.13: K-fold cross-validation and data splitting into training and test data.

We are also visualizing these mean and standard deviation values of regression metrics in bar graphs with error in the end to have a better idea or understanding of the quality of the regression models.

4.3 Regression Metrices

We are not able to do the calculation of the accuracy for a regression model. The performance of a regression model must be calculated as an error in those predictions. It makes sense if we think about it. If we are predicting numeric values like a height or a euro amount, we don't want to know if the model predicted the values exactly the same(this might be intractably difficult practically); so instead, we are required to know how close the values of the prediction were to the expected values or the real values. Error addresses this exactly, and it summarizes on average how close predictions were to their expected values or the real values.

4.3.1 Mean Squared Error

The mean squared error or mean squared deviation of an estimator or regressor make the measurement about the average of the squares of the errors—that is, the average squared difference between the estimated or predicted values (y_predict) and the actual or test value (y_test) so in our case the predicted movie rating and the actual/real movie rating. It is a risk function, which corresponds to the expected value of the squared error loss. We will use *mean_squared_error*() from scikit-learn to calculate it. The following equation denotes the MSE.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y\_test_i - y\_predict_i)^2 \tag{4.2}$$

MSE = mean squared error

N = number of data points

y_test_i = i^{th} observed/test values

$y\_predict_i$ = i^{th} predicted values

,

4.3.2 Mean Absolute Error

The mean absolute error is a measurement of the errors between predicted values and the real values expressing the same meaning. Examples of y_predict versus y_test include comparisons of predicted versus observed values, subsequent time versus initial time, so in our case the predicted movie rating and the actual/real movie rating. We will use $mean\_absolute\_error()$ from scikit-learn to calculate it. The following equation denotes the MAE.

$$MAE = \frac{\sum_{i=1}^{N}|y\_predict_i - y\_test_i|}{N} \tag{4.3}$$

MAE = mean absolute error

N = number of data points

y_test_i = i^{th} observed/test values

$y\_predict_i$ = i^{th} predicted values

Chapter 5

Results and Conclusion

5.1 Results

5.1.1 Cross validation results of regression models

Cross validation for movie rating prediction using Audio, and Visual movie features

Regression Models	Mean of score	std of score
Linear Regression Model	-0.457986	0.214572
XGBRegressor Model	-0.451995	0.215671
Decision Tree Regressor Model	-0.944338	0.217163
KNeighborsRegressor Model	-0.542476	0.500000
SVR Regression Model	-0.464721,	0.228844
Lasso Regression Model	-0.459657	0.214674
Ridge Regression	-0.454693	0.213789
MLPRegressor Model	-0.462530	0.214302
KerasRegressor Model	-0.461246	0.213174
Multimodal Network Regression Model	NA	NA

TABLE 5.1: Description of Cross validation scores of regression models for movie rating prediction using Audio, and Visual movie features

Cross validation for movie rating prediction using Audio, Visual, and Dialogs movie features

Regression Models	Mean of score	std of score
Linear Regression Model	-1.058702	0.490119
XGBRegressor Model	-0.378389	0.224434
Decision Tree Regressor Model	-0.666569	0.326573
KNeighborsRegressor Model	-0.424084	0.196457
SVR Regression Model	-0.386016,	0.225625
Lasso Regression Model	-0.381650	0.218266
Ridge Regression	-0.756912	0.435469
MLPRegressor Model	-893.620579	1905.871427
KerasRegressor Model	-319.1556846	616.291793
Multimodal Network Regression Model	NA	NA

TABLE 5.2: Description of Cross validation scores of regression models for movie rating prediction using Audio, Visual, and Dialogs movie features

5.1.2 Training and testing results of regression models

Mean and STD of MSE and MAE of regression models for movie rating prediction using Audio, and Visual movie features

Regression Model	Mean of MSE	STD of MSE	Mean of MAE	STD of MAE
Linear Regression Model	0.660537	0.155193	0.509636	0.109621
XGBRegressor Model	0.655772	0.156201	0.507016	0.107056
Decision Tree Regressor Model	0.968932	0.099637	0.740856	0.080451
KNeighborsRegressor Model	0.723770	0.143885	0.559518	0.098866
SVR Regression Model	0.663891,	0.163195	0.506364	0.112413
Lasso Regression Model	0.661995	0.154268	0.512714	0.105059
Ridge Regression	0.658053	0.155132	0.507562	0.109187
MLPRegressor Model	0.675494	0.144743	0.529048	0.096819
KerasRegressor Model	0.662441	0.152683	0.512342	0.104014
Multimodal Network Regression Model	2.215118	0.046974	2.117881	0.080671

TABLE 5.3: Description of Mean and std scores of MSE and MAE of regression models for movie rating prediction using Audio, and Visual movie features

Mean and STD of MSE and MAE of regression models for movie rating prediction using Audio, Visual, and movie features

Regression Models	Mean of MSE	STD of MSE	Mean of MAE	STD of MAE
Linear Regression Model	1.006563	0.224927	0.790009	0.160751
XGBRegressor Model	0.594585	0.166192	0.476392	0.122361
Decision Tree Regressor Model	0.807185	0.190159	0.626131	0.168373
KNeighborsRegressor Model	0.634856	0.152905	0.511470	0.126964
SVR Regression Model	0.599327,	0.228844	0.483864	0.131537
Lasso Regression Model	0.596414	0.169773	0.485506	0.141180
Ridge Regression	0.842937	0.226983	0.658263	0.167539
MLPRegressor Model	6.860232	7.105126,	6.823534	7.125080
KerasRegressor Model	9.888266	11.573382	8.619970	10.573196
Multimodal Network Regression Model	2.253213	0.135576	2.210381	0.138703

TABLE 5.4: Description of Mean and std scores of MSE and MAE of regression models for movie rating prediction using Audio, and Visual movie features

5.1.3 Visualization of Mean and STD scores of MSE and MAE scores of regression models

Mean and std scores of MSE and MAE of regression models for movie rating prediction using Audio, and Visual movie features

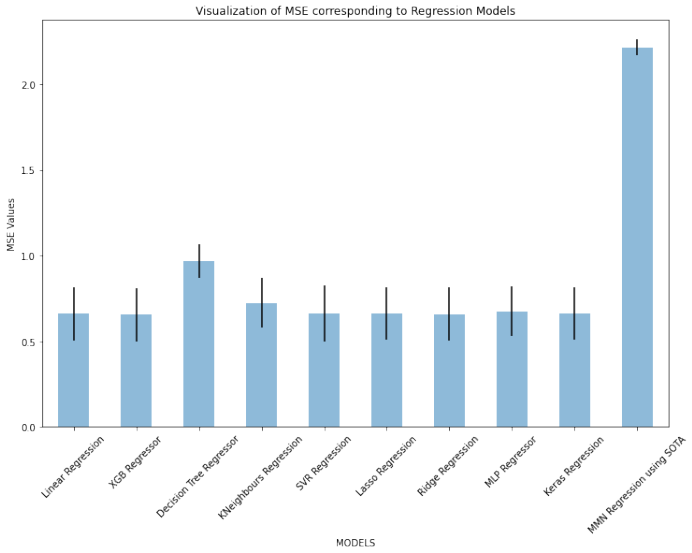

FIGURE 5.1: Visualization of Mean and STD of MSE of regression models for movie rating prediction using Audio, and Visual movie features using bar chart with errors

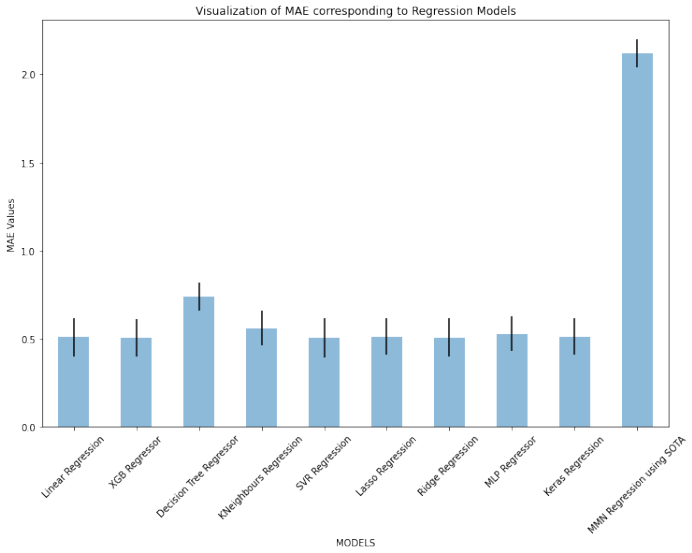

FIGURE 5.2: Visualization of Mean and STD of MAE of regression models for movie rating prediction using Audio, and Visual movie features using bar chart with errors

Mean and std scores of MSE and MAE scores of regression models for movie rating prediction using Audio, Visual, and Dialogs movie features

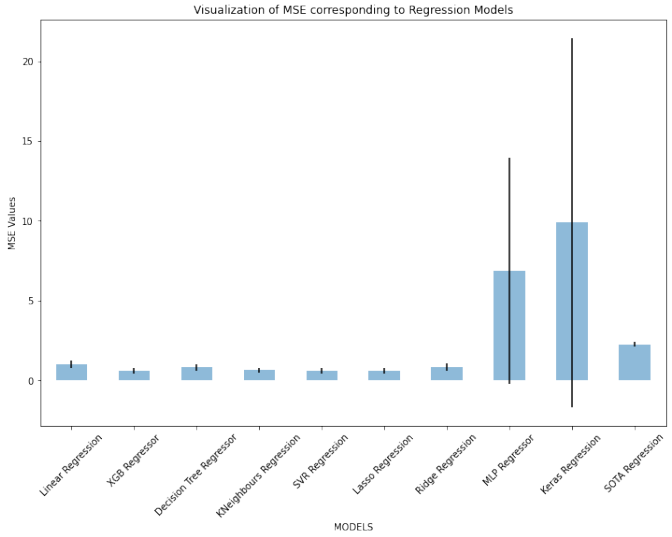

FIGURE 5.3: Visualization of Mean and STD of MSE of regression models for movie rating prediction using Audio, Visual, and dialogs movie features using bar chart with errors

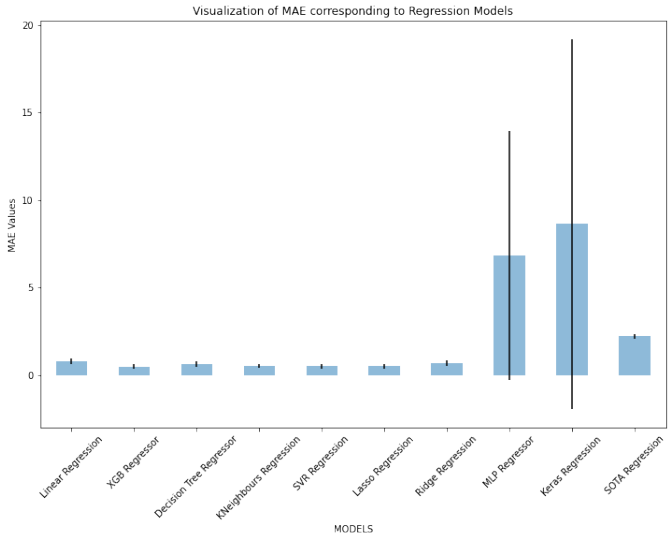

FIGURE 5.4: Visualization of Mean and STD of MAE of regression models for movie rating prediction using Audio, Visual, and dialogs movie features using bar chart with errors

5.2 Conclusion

By analyzing the cross validation scores we can say that the XGBRegressor Model give a good mean score which is -0.451995 it is a negative value because we are using scoring='neg_mean_squared_error' so if we just take the value without minus sign this is the lowest value in cross validation results , hence this model give a good fit for the data when we are just using Audio and Visual movie features for movie rating prediction. When we are using all three features Audio, Visual, and dialogs movie features again the XGBRegressor Model give best fit.

From Training and testing results we can say that when we are predicting movie ratings using the Audio and Visual features and with all three features(Audio, Visual, dialogs features) the XGBRegression Model gives the lowest Mean of MSE which is 0.655772 in case of Audio and Visual movie features and 0.594585 in case of using all three modalities or features so this is best model as per mean of MSE values as they are lowest. But if we see the mean of MAE when we are using two modalities(Audio and Visual) it has lowest value so we can say this model is also a good. If we see the mean of MAE of XGBRegression model in case of three modalities(Audio, Visual, and Dialogs) it has the smallest value which is 0.476392. So overall we can say that the XGBRegressor Model performs the best. Afterward, the worst performing model is Multimodel Network Regression model in case of two modalities(Audio and Visual) as it has highest mean of MSE and MAE values which are 2.215118, 2.117881 respectively. In case of three features(Audio, Visual, and Dialogs) the Keras Regressor is worst fit as the mean values of MAE and MAE are highest which are 9.888266, 8.619970. If we talk about the Multimodal Network Regression model is has some moderate error values also they are remained almost the same. Most of the regression models perform better with three modalities that shows that their quality is enhanced using when we are the dialogs features additionaly.

In the end, we can conclude that the idea of using multi modalities of movies to predict the movie ratings is quite good as the results also approve it. Also the research area of Multimodal machine learning or multimodal deep learning has a great significance that helps to make use of different modalities of a thing like movies to learn and combine those modalities together.

Appendix A

Gantt Diagram

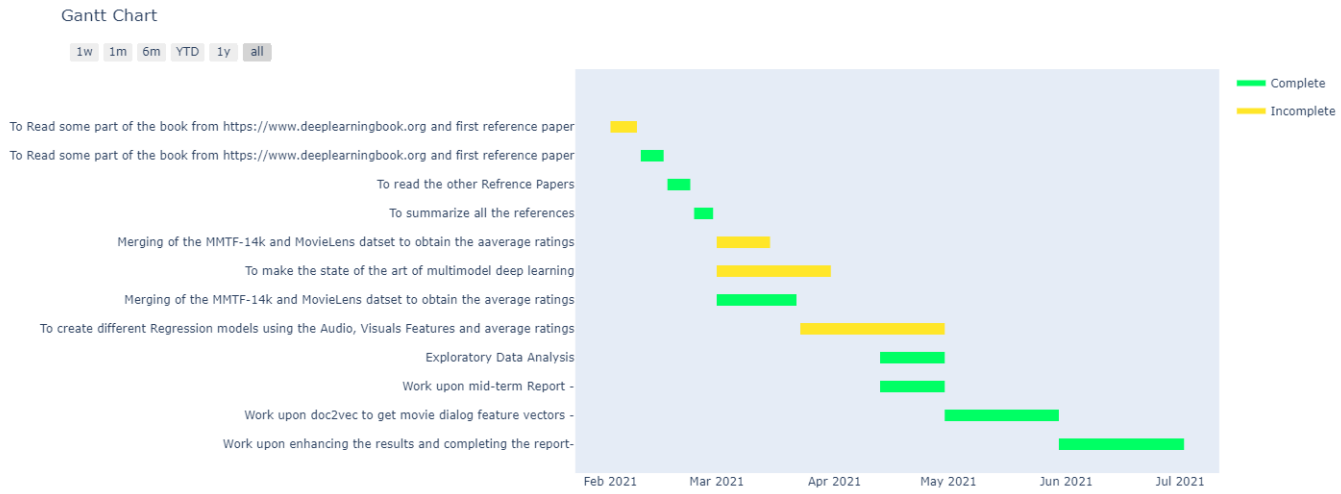

FIGURE A.1: Gantt Chart showing progress of the work.

Bibliography

Abbas, Khushnood. "Abbas, Khushnood (2017), "Movielens 20M Dataset", Mendeley Data, V3". In: *"Movielens 20M Dataset 2017"*. DOI: 10.17632/n6sjkpy87f.3.

Danescu-Niculescu-Mizil, Cristian and Lillian Lee (2011). "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs." In: *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Deldjoo, Yashar et al. (2018). "MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval". In: *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM. DOI: 10.1145/3204949.3208141.

Morency, Louis-Philippe and Tadas Baltrušaitis (July 2017). "Multimodal Machine Learning: Integrating Language, Vision and Speech". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Vancouver, Canada: Association for Computational Linguistics, pp. 3–5. URL: https://aclanthology.org/P17-5002.

Ngiam, Jiquan et al. (Jan. 2011). "Multimodal Deep Learning". In: pp. 689–696.

Oramas, Sergio et al. (Aug. 2017). "A Deep Multimodal Approach for Cold-start Music Recommendation". In: pp. 32–37. DOI: 10.1145/3125486.3125492.

Zhao, Zhou et al. (Aug. 2017). "Social-Aware Movie Recommendation via Multimodal Network Learning". In: *IEEE Transactions on Multimedia* PP, pp. 1–1. DOI: 10.1109/TMM.2017.2740022.