CAB420 - Assignment 1C

Callum McNeilage - n10482652

Connor Smith - n9991051

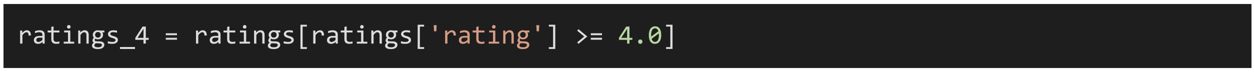
Queensland University of Technology

Problem 1: Clustering and Recommendations

For this task we used (Asad, 2020) as reference.

Data Processing and Handling

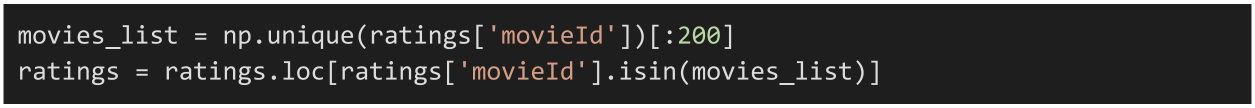
For this problem we were given several .csv files; ‘links.csv’, ‘movies.csv’, ‘ratings.csv’ and ‘tags.csv’. In order to improve performance, we limited the ratings dataset to only ratings above 4, movies dataset to the first 200 movies and users dataset to the first 315 users. We chose to limit the ratings to 4 and above as any rating below 4 would likely be unusable by the model as a user who rates a movie a 1, 2 or 3 generally does not like that movie. This also meant we did not have to do any processing of whether a user liked the movie later on as only liked movies are being considered by the model.



```
ratings_4 = ratings[ratings['rating'] >= 4.0]
```

Figure 1: Limit ratings to 4 and above

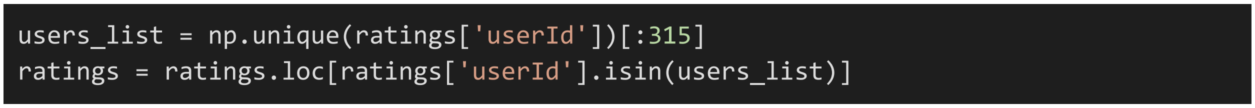
As the movies dataset contains upwards of 100836 individual movies, this would require a large number of clusters in our KMeans function. As such, we decided to limit the movies to the first 200 movies in the dataset. This allowed us to limit our number of clusters to 82, which was much faster to compute on the hardware available.



```
movies_list = np.unique(ratings['movieId'])[:200]
ratings = ratings.loc[ratings['movieId'].isin(movies_list)]
```

Figure 2: Limit movies to first 200

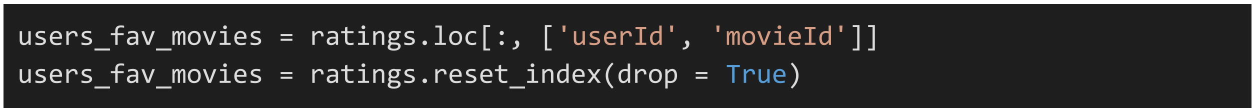
As only Users 4, 42 and 314 were required for testing the output of our model we decided to also limit our users dataset to the first 315 users to improve compute time.



```
users_list = np.unique(ratings['userId'])[:315]
ratings = ratings.loc[ratings['userId'].isin(users_list)]
```

Figure 3: Limit users to first 315

We then combined these datasets into a single combined dataset for training (See Appendix 1 for sample dataset).



```
users_fav_movies = ratings.loc[:, ['userId', 'movieId']]
users_fav_movies = ratings.reset_index(drop = True)
```

Figure 4: Combine all three datasets into 1

From this processed data, we then created a Sparse Matrix of users against the movies they have rated (See Appendix 2) which was then used in our Elbow Method to find the appropriate number of clusters for our dataset.

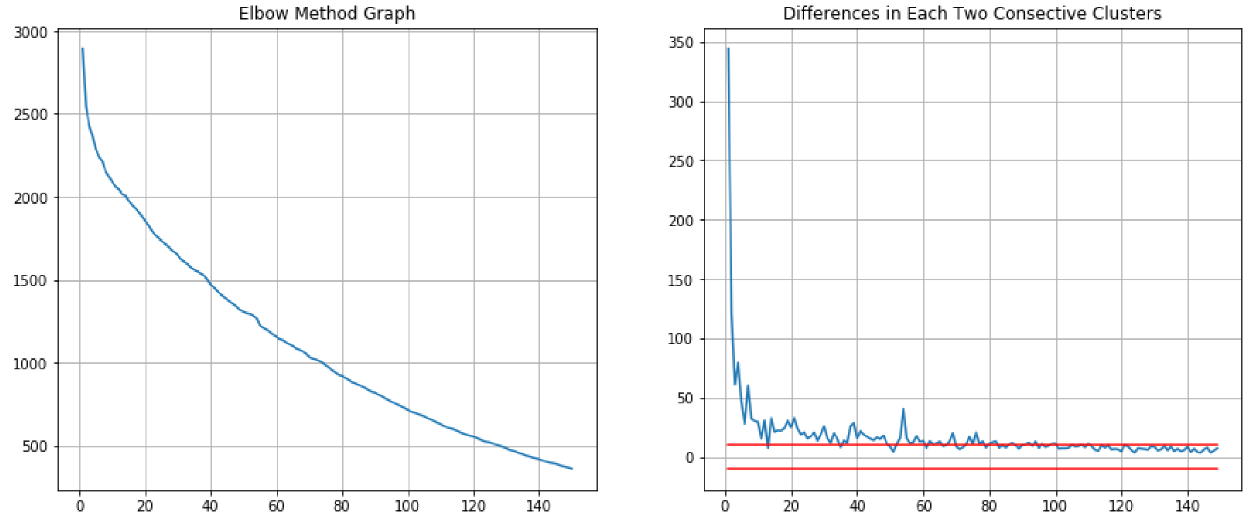
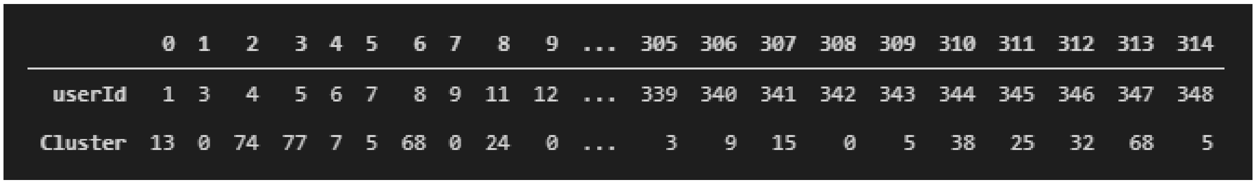


Figure 5: Elbow Method for dataset

Clustering Method

It was determined using the elbow method above that 82 clusters was the optimal for our KMeans model.



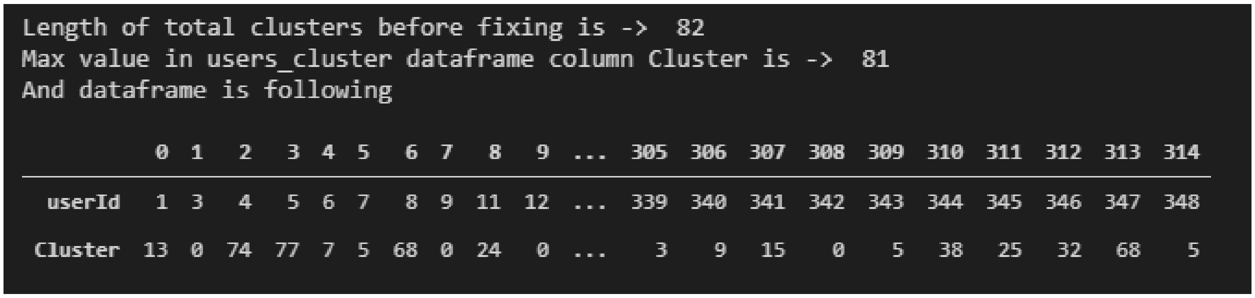
	0	1	2	3	4	5	6	7	8	9	...	305	306	307	308	309	310	311	312	313	314
userId	1	3	4	5	6	7	8	9	11	12	...	339	340	341	342	343	344	345	346	347	348
Cluster	13	0	74	77	7	5	68	0	24	0	...	3	9	15	0	5	38	25	32	68	5

Figure 6: Dataframe of clusters

We chose to use KMeans Clustering because the data utilized numerical values, and after or processing, the data contained few dimensions. In addition, we wanted any particular user to only be assigned one cluster.

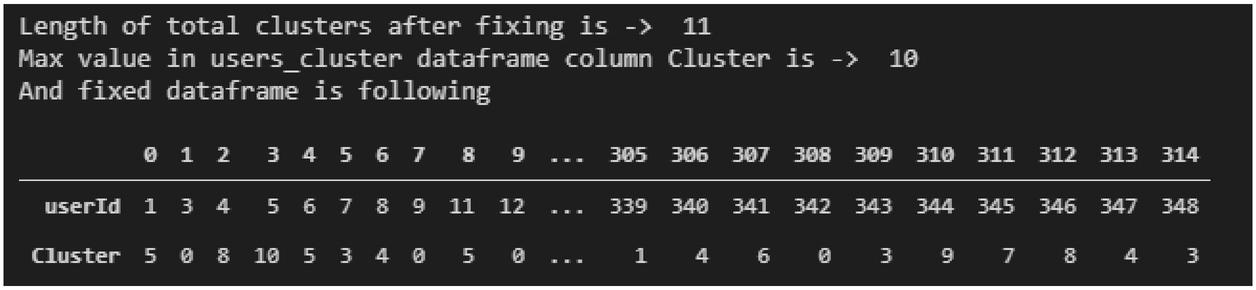
Results

After clustering, we discovered a large number of clusters contained only a single movie. As the users in these clusters would not receive any recommendations for movies if they remained in their current cluster. As such, we removed the users from those single user clusters and combined them into a single cluster ‘uncategorized’. This ensured that those users could still receive recommendations, although they may not be as reliable as recommendations for users who are still in their original clusters. This allowed us to reduce the number of overall clusters from 82 to 11.



```
Length of total clusters before fixing is ->  82
Max value in users_cluster dataframe column Cluster is ->  81
And dataframe is following

          0  1   2   3  4  5   6  7   8   9  ...  305  306  307  308  309  310  311  312  313  314
--------------------------------------------------------------------------------------------------
 userId   1  3   4   5  6  7   8  9  11  12  ...  339  340  341  342  343  344  345  346  347  348

Cluster  13  0  74  77  7  5  68  0  24   0  ...    3    9   15    0    5   38   25   32   68    5
```

Figure 7: Clusters before fixing

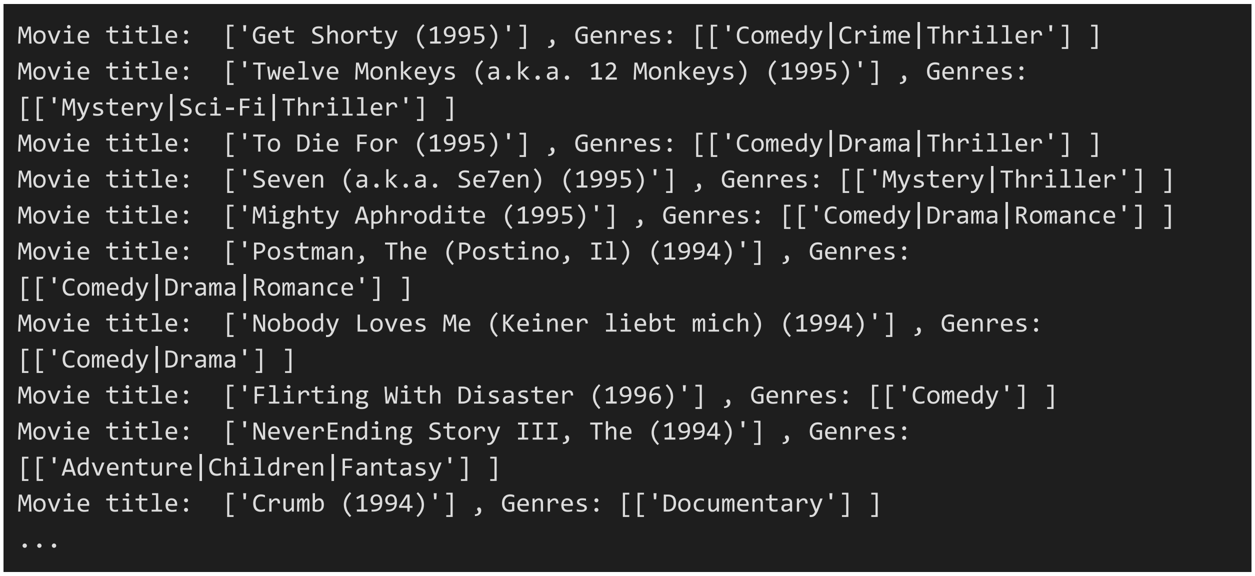


```
Length of total clusters after fixing is ->  11
Max value in users_cluster dataframe column Cluster is ->  10
And fixed dataframe is following

         0  1  2   3  4  5  6  7   8   9  ...  305  306  307  308  309  310  311  312  313  314
-----------------------------------------------------------------------------------------------
 userId  1  3  4   5  6  7  8  9  11  12  ...  339  340  341  342  343  344  345  346  347  348

Cluster  5  0  8  10  5  3  4  0   5   0  ...    1    4    6    0    3    9    7    8    4    3
```

Figure 8: Clusters after fixing

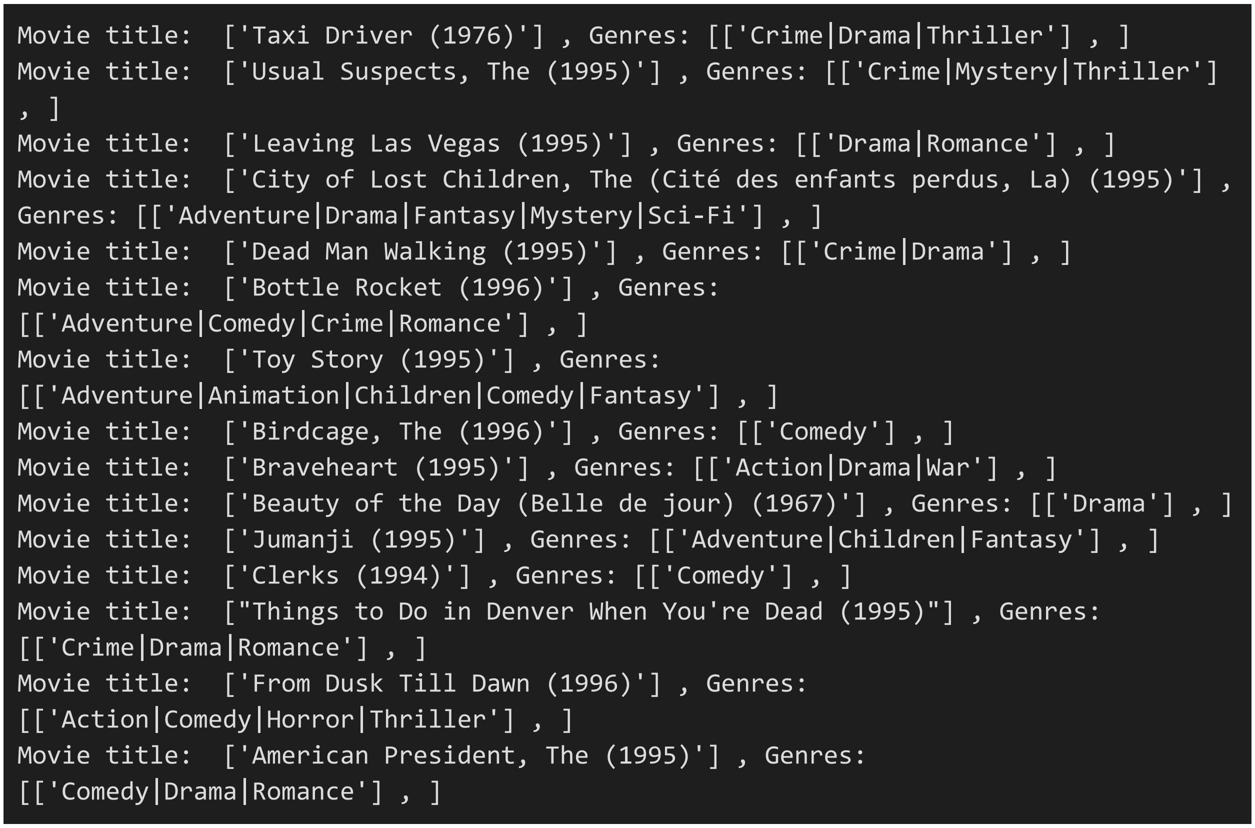
Recommendations

User 4 Recommendations



```
Movie title:  ['Get Shorty (1995)'] , Genres: [['Comedy|Crime|Thriller'] ]
Movie title:  ['Twelve Monkeys (a.k.a. 12 Monkeys) (1995)'] , Genres:
[['Mystery|Sci-Fi|Thriller'] ]
Movie title:  ['To Die For (1995)'] , Genres: [['Comedy|Drama|Thriller'] ]
Movie title:  ['Seven (a.k.a. Se7en) (1995)'] , Genres: [['Mystery|Thriller'] ]
Movie title:  ['Mighty Aphrodite (1995)'] , Genres: [['Comedy|Drama|Romance'] ]
Movie title:  ['Postman, The (Postino, Il) (1994)'] , Genres:
[['Comedy|Drama|Romance'] ]
Movie title:  ['Nobody Loves Me (Keiner liebt mich) (1994)'] , Genres:
[['Comedy|Drama'] ]
Movie title:  ['Flirting With Disaster (1996)'] , Genres: [['Comedy'] ]
Movie title:  ['NeverEnding Story III, The (1994)'] , Genres:
[['Adventure|Children|Fantasy'] ]
Movie title:  ['Crumb (1994)'] , Genres: [['Documentary'] ]
...
```

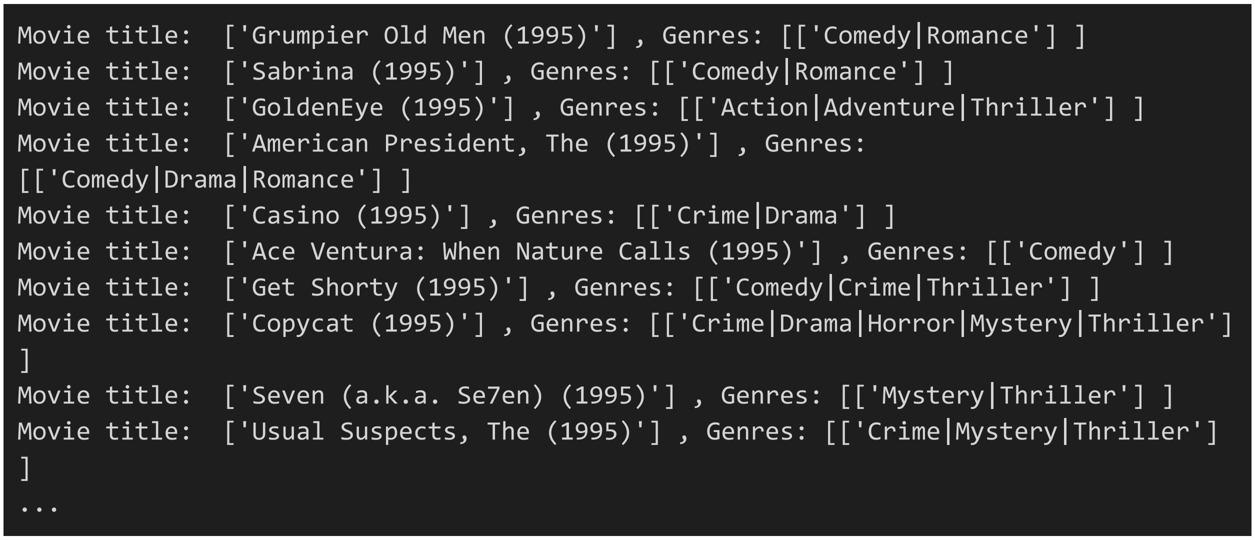
Figure 9: Movies watched by User 4



```
Movie title:  ['Taxi Driver (1976)'] , Genres: [['Crime|Drama|Thriller'] , ]
Movie title:  ['Usual Suspects, The (1995)'] , Genres: [['Crime|Mystery|Thriller']
, ]
Movie title:  ['Leaving Las Vegas (1995)'] , Genres: [['Drama|Romance'] , ]
Movie title:  ['City of Lost Children, The (Cité des enfants perdus, La) (1995)'] ,
Genres: [['Adventure|Drama|Fantasy|Mystery|Sci-Fi'] , ]
Movie title:  ['Dead Man Walking (1995)'] , Genres: [['Crime|Drama'] , ]
Movie title:  ['Bottle Rocket (1996)'] , Genres:
[['Adventure|Comedy|Crime|Romance'] , ]
Movie title:  ['Toy Story (1995)'] , Genres:
[['Adventure|Animation|Children|Comedy|Fantasy'] , ]
Movie title:  ['Birdcage, The (1996)'] , Genres: [['Comedy'] , ]
Movie title:  ['Braveheart (1995)'] , Genres: [['Action|Drama|War'] , ]
Movie title:  ['Beauty of the Day (Belle de jour) (1967)'] , Genres: [['Drama'] , ]
Movie title:  ['Jumanji (1995)'] , Genres: [['Adventure|Children|Fantasy'] , ]
Movie title:  ['Clerks (1994)'] , Genres: [['Comedy'] , ]
Movie title:  ["Things to Do in Denver When You're Dead (1995)"] , Genres:
[['Crime|Drama|Romance'] , ]
Movie title:  ['From Dusk Till Dawn (1996)'] , Genres:
[['Action|Comedy|Horror|Thriller'] , ]
Movie title:  ['American President, The (1995)'] , Genres:
[['Comedy|Drama|Romance'] , ]
```

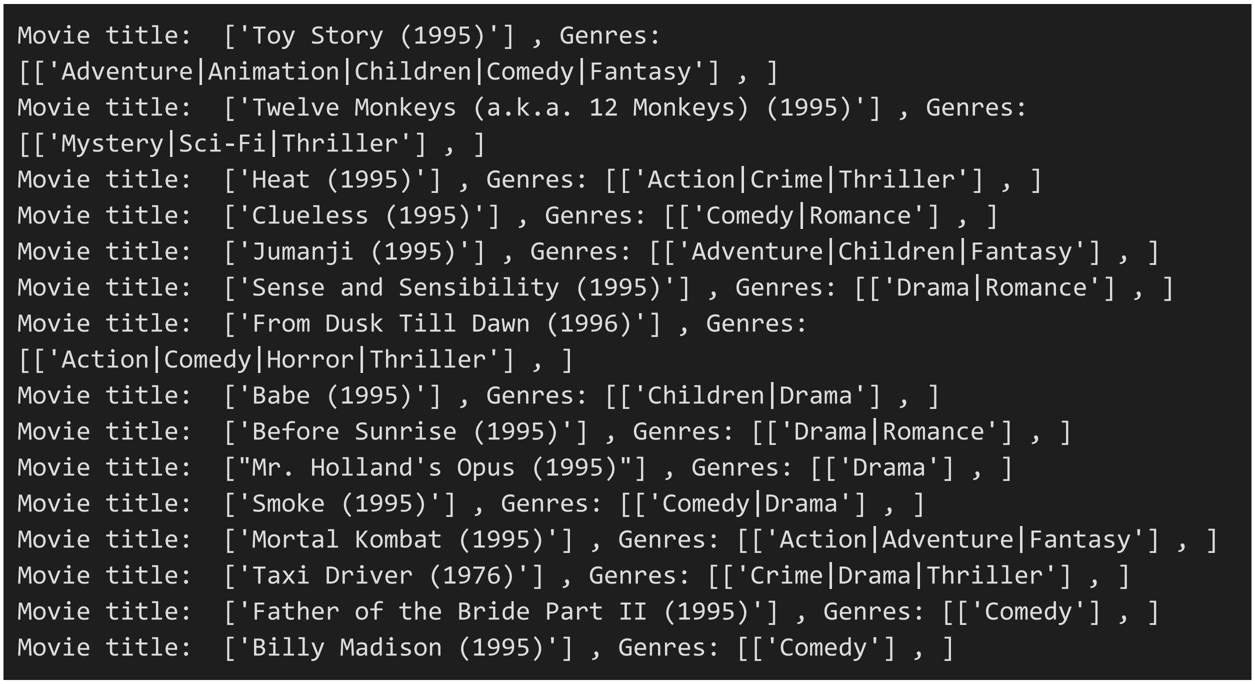
Figure 10: Recommendations for User 4

User 42 Recommendations



```
Movie title:  ['Grumpier Old Men (1995)'] , Genres: [['Comedy|Romance'] ]
Movie title:  ['Sabrina (1995)'] , Genres: [['Comedy|Romance'] ]
Movie title:  ['GoldenEye (1995)'] , Genres: [['Action|Adventure|Thriller'] ]
Movie title:  ['American President, The (1995)'] , Genres:
[['Comedy|Drama|Romance'] ]
Movie title:  ['Casino (1995)'] , Genres: [['Crime|Drama'] ]
Movie title:  ['Ace Ventura: When Nature Calls (1995)'] , Genres: [['Comedy'] ]
Movie title:  ['Get Shorty (1995)'] , Genres: [['Comedy|Crime|Thriller'] ]
Movie title:  ['Copycat (1995)'] , Genres: [['Crime|Drama|Horror|Mystery|Thriller']
]
Movie title:  ['Seven (a.k.a. Se7en) (1995)'] , Genres: [['Mystery|Thriller'] ]
Movie title:  ['Usual Suspects, The (1995)'] , Genres: [['Crime|Mystery|Thriller']
]
...
```

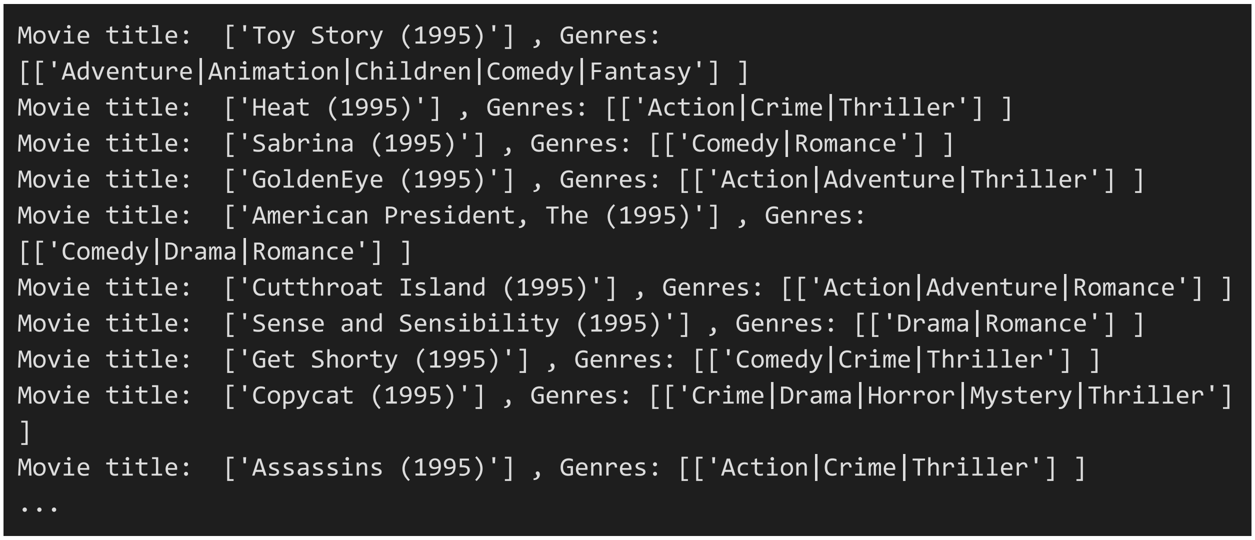
Figure 11: Movies watched by User 42



```
Movie title:  ['Toy Story (1995)'] , Genres:
[['Adventure|Animation|Children|Comedy|Fantasy'] , ]
Movie title:  ['Twelve Monkeys (a.k.a. 12 Monkeys) (1995)'] , Genres:
[['Mystery|Sci-Fi|Thriller'] , ]
Movie title:  ['Heat (1995)'] , Genres: [['Action|Crime|Thriller'] , ]
Movie title:  ['Clueless (1995)'] , Genres: [['Comedy|Romance'] , ]
Movie title:  ['Jumanji (1995)'] , Genres: [['Adventure|Children|Fantasy'] , ]
Movie title:  ['Sense and Sensibility (1995)'] , Genres: [['Drama|Romance'] , ]
Movie title:  ['From Dusk Till Dawn (1996)'] , Genres:
[['Action|Comedy|Horror|Thriller'] , ]
Movie title:  ['Babe (1995)'] , Genres: [['Children|Drama'] , ]
Movie title:  ['Before Sunrise (1995)'] , Genres: [['Drama|Romance'] , ]
Movie title:  ["Mr. Holland's Opus (1995)"] , Genres: [['Drama'] , ]
Movie title:  ['Smoke (1995)'] , Genres: [['Comedy|Drama'] , ]
Movie title:  ['Mortal Kombat (1995)'] , Genres: [['Action|Adventure|Fantasy'] , ]
Movie title:  ['Taxi Driver (1976)'] , Genres: [['Crime|Drama|Thriller'] , ]
Movie title:  ['Father of the Bride Part II (1995)'] , Genres: [['Comedy'] , ]
Movie title:  ['Billy Madison (1995)'] , Genres: [['Comedy'] , ]
```

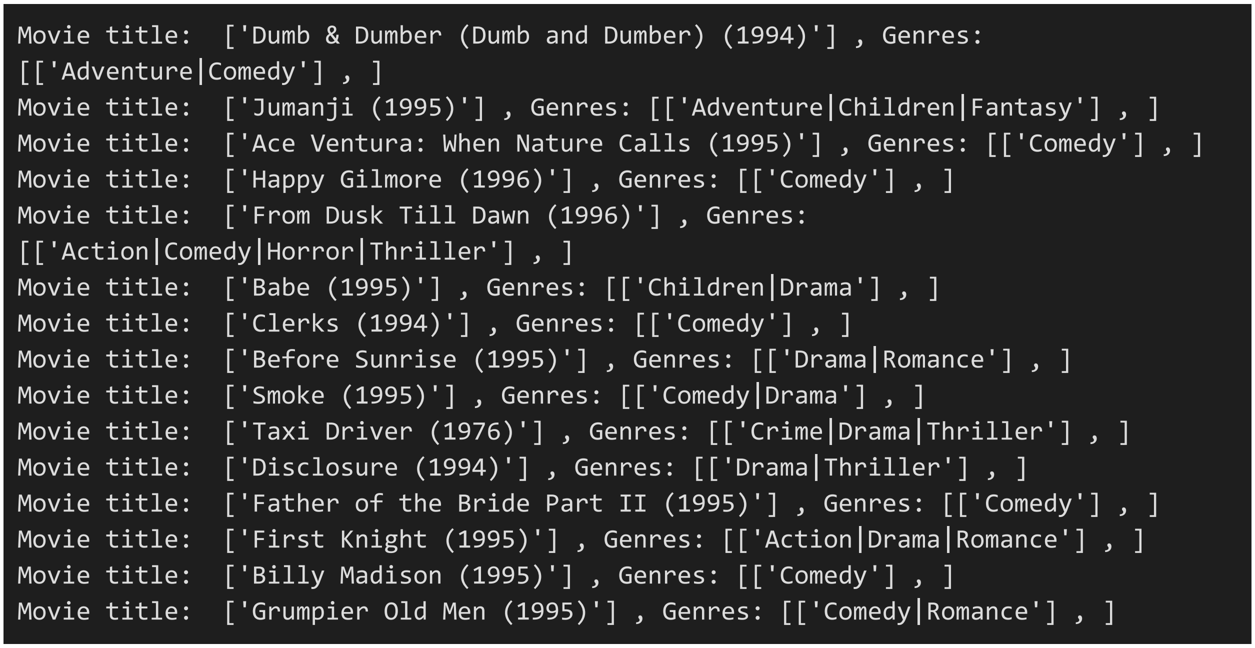
Figure 12: Recommendations for User 42

User 314 Recommendations



```
Movie title:  ['Toy Story (1995)'] , Genres:
[['Adventure|Animation|Children|Comedy|Fantasy'] ]
Movie title:  ['Heat (1995)'] , Genres: [['Action|Crime|Thriller'] ]
Movie title:  ['Sabrina (1995)'] , Genres: [['Comedy|Romance'] ]
Movie title:  ['GoldenEye (1995)'] , Genres: [['Action|Adventure|Thriller'] ]
Movie title:  ['American President, The (1995)'] , Genres:
[['Comedy|Drama|Romance'] ]
Movie title:  ['Cutthroat Island (1995)'] , Genres: [['Action|Adventure|Romance'] ]
Movie title:  ['Sense and Sensibility (1995)'] , Genres: [['Drama|Romance'] ]
Movie title:  ['Get Shorty (1995)'] , Genres: [['Comedy|Crime|Thriller'] ]
Movie title:  ['Copycat (1995)'] , Genres: [['Crime|Drama|Horror|Mystery|Thriller']
]
Movie title:  ['Assassins (1995)'] , Genres: [['Action|Crime|Thriller'] ]
...
```

Figure 13: Movies watched by User 314



```
Movie title:  ['Dumb & Dumber (Dumb and Dumber) (1994)'] , Genres:
[['Adventure|Comedy'] , ]
Movie title:  ['Jumanji (1995)'] , Genres: [['Adventure|Children|Fantasy'] , ]
Movie title:  ['Ace Ventura: When Nature Calls (1995)'] , Genres: [['Comedy'] , ]
Movie title:  ['Happy Gilmore (1996)'] , Genres: [['Comedy'] , ]
Movie title:  ['From Dusk Till Dawn (1996)'] , Genres:
[['Action|Comedy|Horror|Thriller'] , ]
Movie title:  ['Babe (1995)'] , Genres: [['Children|Drama'] , ]
Movie title:  ['Clerks (1994)'] , Genres: [['Comedy'] , ]
Movie title:  ['Before Sunrise (1995)'] , Genres: [['Drama|Romance'] , ]
Movie title:  ['Smoke (1995)'] , Genres: [['Comedy|Drama'] , ]
Movie title:  ['Taxi Driver (1976)'] , Genres: [['Crime|Drama|Thriller'] , ]
Movie title:  ['Disclosure (1994)'] , Genres: [['Drama|Thriller'] , ]
Movie title:  ['Father of the Bride Part II (1995)'] , Genres: [['Comedy'] , ]
Movie title:  ['First Knight (1995)'] , Genres: [['Action|Drama|Romance'] , ]
Movie title:  ['Billy Madison (1995)'] , Genres: [['Comedy'] , ]
Movie title:  ['Grumpier Old Men (1995)'] , Genres: [['Comedy|Romance'] , ]
```

Figure 14: Recommendations for User 314

As can be seen in Figures 9-14, our model provides relatively accurate recommendations for each user, with at least one of the recommended movie’s genres existing in the user’s watched movies. We are confident this pattern is followed by all other users in the dataset.

Problem 2: Semantic Person Search

Pre-processing

The data used consisted of a set of png files and a csv containing the labels. The training data and test data was already split 520 images for training and 196 for testing. As the dataset was already really small, we decided to accept the unknown class as its own class rather than missing data. Removing it would limit our data too much. The images were all of different sizes, with the smallest being 108x53 and the largest being 429x267. As all the images need to be the same size for the model, they were all resized to 268x160, the middle value between largest and smallest. This also helps to reduce runtime as the images are quite large. The csv had all irrelevant values removed and the rest was split into their own arrays. The segmentation data was ignored as it wasnt available with the test set.

Description of Approach

For this problem, we created a neural network with 6 different outputs. Each output was each of the traits to be classified; gender, torso clothing type, torso colour, leg clothing type, leg colour and luggage. Originally the model was a small model consisting of 3 convolutions before splitting off into each trait. The full plot for this can be seen in appendix 3. This produced bad results so we added some convolutions after splitting into each trait. This still had poor performance so we used the 3rd example from week 4 example 2. This is a much deeper network with more convolutions and some dropout to help with overfitting. Each trait was also split immediately giving each trait the entire network. This significantly increased training time so it was only trained for 20 epochs. This network can be seen fully in appendix 4.

Evaluation of Performance

Epoch	Gender	Torso Type	Torso Colour	Leg Type	Leg Colour	Luggage
16	17%	80%	45%	13%	15%	17%
17	32%	80%	13%	12%	14%	16%
18	5%	84%	9%	10%	16%	18%
19	1%	84%	9%	14%	14%	17%
20	13%	85%	10%	12%	12%	19%

Figure 15: Training accuracy for last 5 epochs

Epoch	Gender	Torso Type	Torso Colour	Leg Type	Leg Colour	Luggage
16	45%	61%	22%	31%	5%	64%
17	0%	61%	22%	27%	5%	64%
28	5%	61%	22%	26%	5%	64%
19	12%	38%	22%	26%	5%	64%
20	45%	61%	22%	24%	5%	63%

Figure 15: Validation accuracy for last 5 epochs

Overall our model performed poorly for all traits except torso type. The model behaved quite bizarrely for the rest of the traits. For gender, it's quite obviously guessing for both training and validation. With only 3 values to choose from, it explains the mix between high and low values. The images were quite low res and included the whole body. There are also a large amount of shots from behind. This makes it hard to see features that indicate gender, especially the face.

Looking at the data manually, it is hard to determine gender by eye so it makes sense that the model struggled with this category. Torso type is the only trait to have reasonable performance. Finishing at 85% for training and 61% for validation. Most images have a good shot of the torso allowing for features to be extracted. Torso Colour seems to be guessing as well, indicated by the low, varied accuracy. The validation accuracy being a consistent 22% is odd. This may indicate some user error in coding the network. Leg type and leg colour both have similarly bad training performance. It once again indicates that the model is guessing. Most of the images are shot from a higher angle, making the legs harder to see. This may indicate the bad performance in both these traits. The validation accuracy is once again strangely consistent for leg colour, maybe indicating user error. Luggage once again had poor performance in training but had an oddly high and consistent validation accuracy.

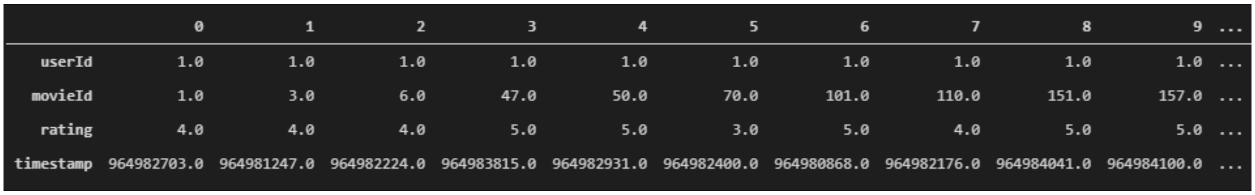
This model could be improved in a number of ways. Firstly there were some issues with the data. As all the images were different sizes, resizing them would remove some features, especially for the larger images. The images were also really inconsistent with angles and lighting. The dataset was also really small. A larger more consistent dataset would produce better results, and data augmentation could be used to increase the current dataset a bit. The model could also be adjusted to have both torso traits and both leg traits on the same branch for a bit before splitting off. This would reduce computation time allowing for more epochs to be added. Using a pre-trained model would also help improve accuracy.

References

Asad, S. M. (2020, August 19). *AI Movies Recommendation System with Clustering Based K-Means Algorithm*. Medium. https://asdkazmi.medium.com/ai-movies-recommendation-system-with-clustering-based-k-means-algorithm-f04467e02fcd

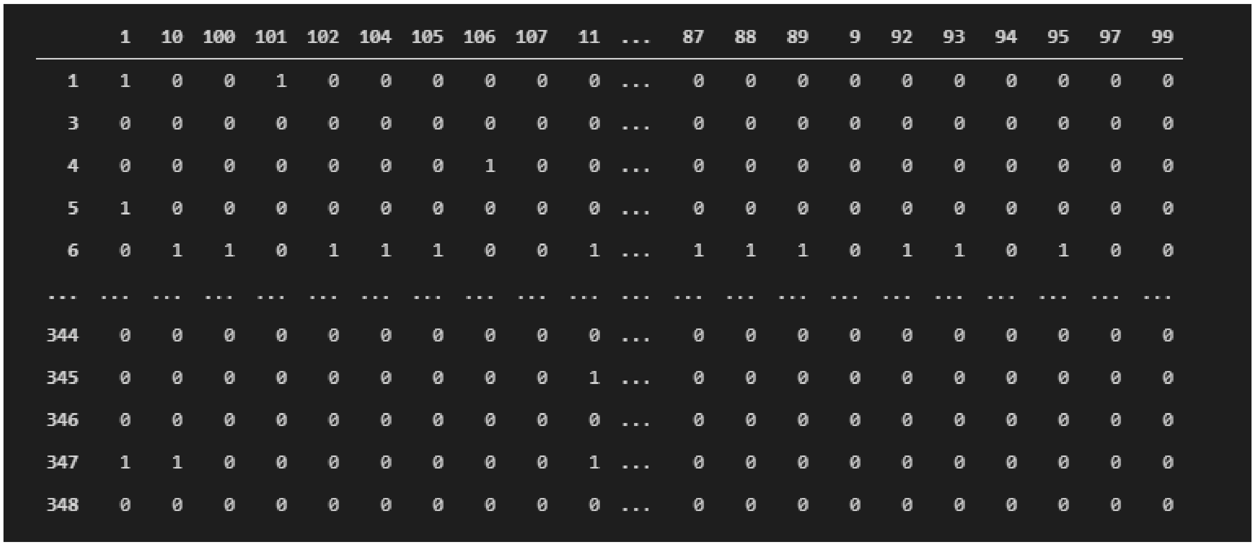
Appendix

Appendix 1



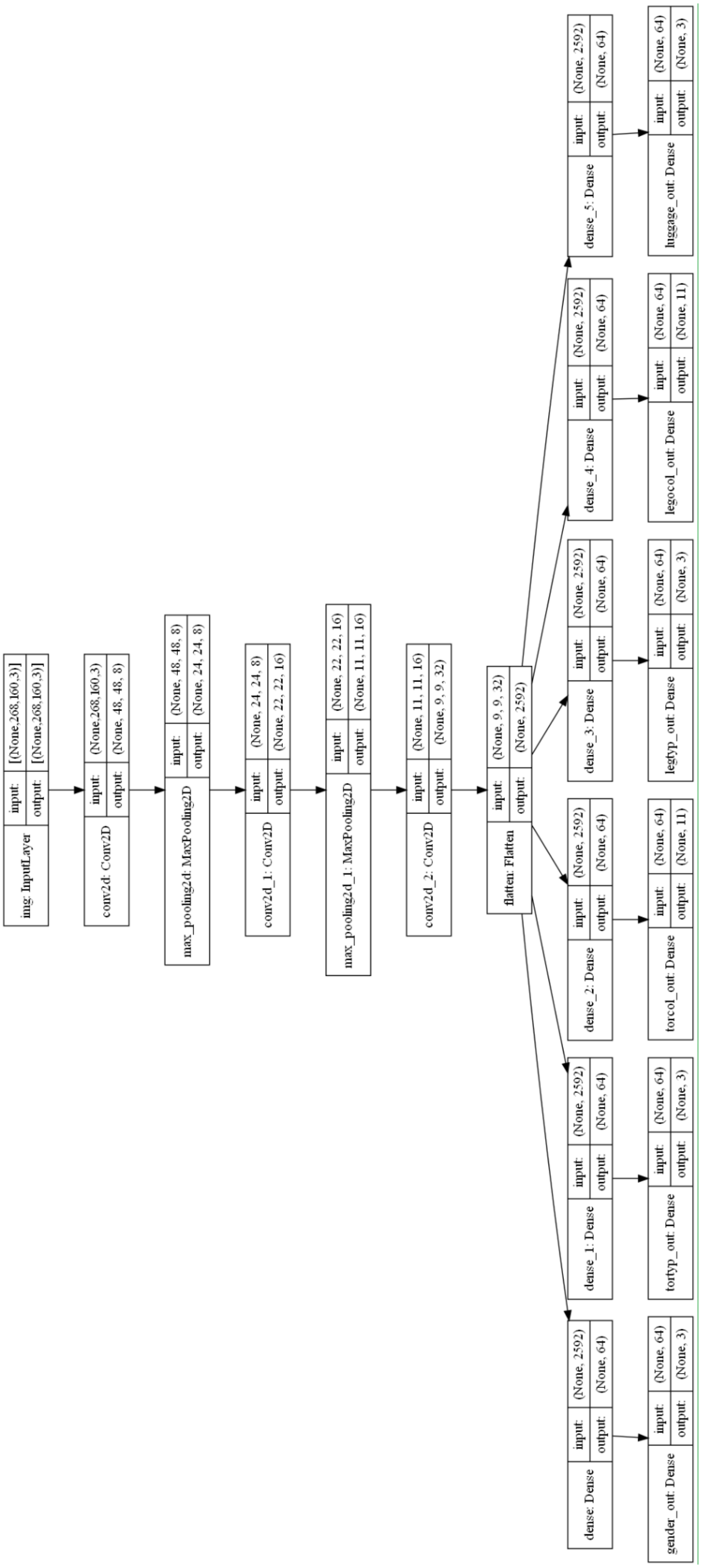
	0	1	2	3	4	5	6	7	8	9	...
userId	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...
movieId	1.0	3.0	6.0	47.0	50.0	70.0	101.0	110.0	151.0	157.0	...
rating	4.0	4.0	4.0	5.0	5.0	3.0	5.0	4.0	5.0	5.0	...
timestamp	964982703.0	964981247.0	964982224.0	964983815.0	964982931.0	964982400.0	964980868.0	964982176.0	964984041.0	964984100.0	...

Appendix 2

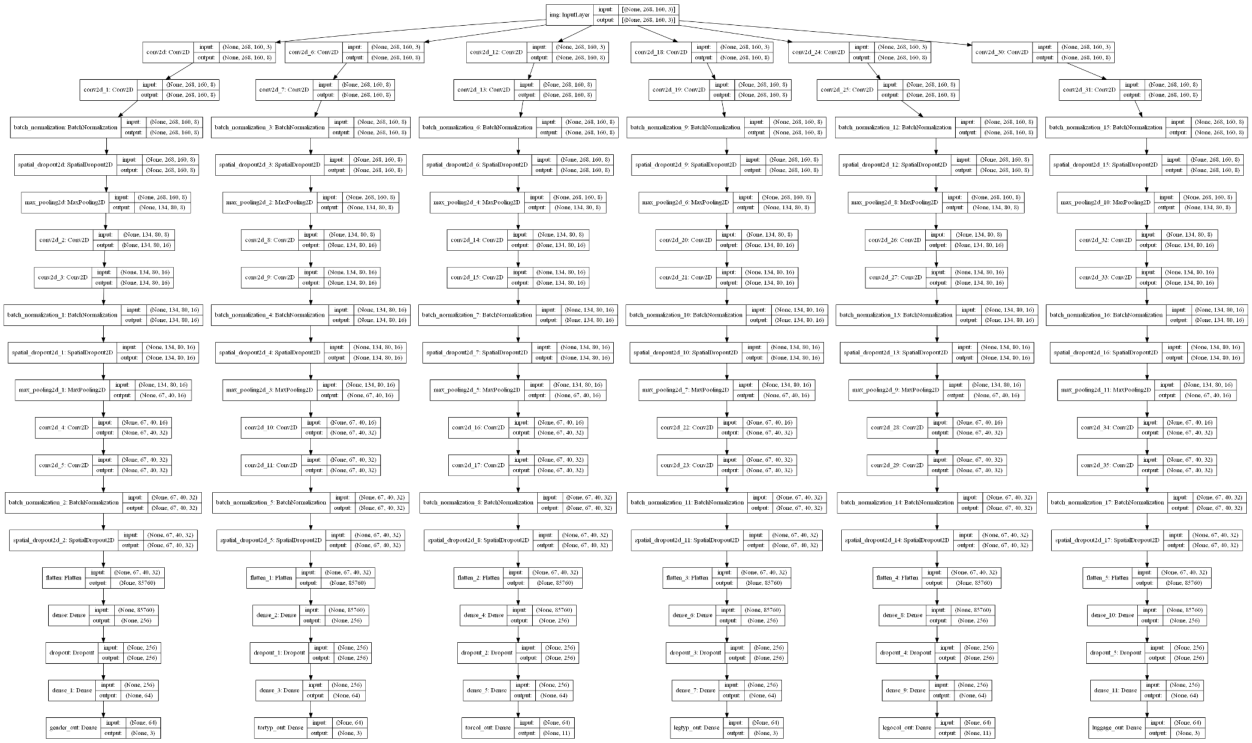


	1	10	100	101	102	104	105	106	107	11	...	87	88	89	9	92	93	94	95	97	99
1	1	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6	0	1	1	0	1	1	1	0	0	1	...	1	1	1	0	1	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
344	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
345	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
346	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
347	1	1	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
348	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Appendix 3


img: InputLayer
input: [(None,268,160,3)]
output: [(None,268,160,3)]
conv2d: Conv2D
input: (None,268,160,3)
output: (None, 48, 48, 8)
max_pooling2d: MaxPooling2D
input: (None, 48, 48, 8)
output: (None, 24, 24, 8)
conv2d_1: Conv2D
input: (None, 24, 24, 8)
output: (None, 22, 22, 16)
max_pooling2d_1: MaxPooling2D
input: (None, 22, 22, 16)
output: (None, 11, 11, 16)
conv2d_2: Conv2D
input: (None, 11, 11, 16)
output: (None, 9, 9, 32)
flatten: Flatten
input: (None, 9, 9, 32)
output: (None, 2592)
dense: Dense
input: (None, 2592)
output: (None, 64)
gender_out: Dense
input: (None, 64)
output: (None, 3)
dense_1: Dense
input: (None, 2592)
output: (None, 64)
tortyp_out: Dense
input: (None, 64)
output: (None, 3)
dense_2: Dense
input: (None, 2592)
output: (None, 64)
torcol_out: Dense
input: (None, 64)
output: (None, 11)
dense_3: Dense
input: (None, 2592)
output: (None, 64)
legtyp_out: Dense
input: (None, 64)
output: (None, 3)
dense_4: Dense
input: (None, 2592)
output: (None, 64)
legcol_out: Dense
input: (None, 64)
output: (None, 11)
dense_5: Dense
input: (None, 2592)
output: (None, 64)
luggage_out: Dense
input: (None, 64)
output: (None, 3)

Appendix 4


img: InputLayer input: [(None, 268, 160, 3)] output: [(None, 268, 160, 3)]
conv2d: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_1: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_2: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_3: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_1: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_1: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_1: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_4: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_5: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_2: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_2: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense: Dense input: (None, 85760) output: (None, 256)
dropout: Dropout input: (None, 256) output: (None, 256)
dense_1: Dense input: (None, 256) output: (None, 64)
gender_out: Dense input: (None, 64) output: (None, 3)
conv2d_6: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_7: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization_3: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d_3: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d_2: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_8: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_9: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_4: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_4: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_3: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_10: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_11: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_5: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_5: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten_1: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense_2: Dense input: (None, 85760) output: (None, 256)
dropout_1: Dropout input: (None, 256) output: (None, 256)
dense_3: Dense input: (None, 256) output: (None, 64)
tortyp_out: Dense input: (None, 64) output: (None, 3)
conv2d_12: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_13: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization_6: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d_6: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d_4: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_14: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_15: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_7: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_7: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_5: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_16: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_17: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_8: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_8: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten_2: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense_4: Dense input: (None, 85760) output: (None, 256)
dropout_2: Dropout input: (None, 256) output: (None, 256)
dense_5: Dense input: (None, 256) output: (None, 64)
torcol_out: Dense input: (None, 64) output: (None, 11)
conv2d_18: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_19: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization_9: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d_9: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d_6: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_20: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_21: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_10: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_10: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_7: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_22: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_23: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_11: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_11: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten_3: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense_6: Dense input: (None, 85760) output: (None, 256)
dropout_3: Dropout input: (None, 256) output: (None, 256)
dense_7: Dense input: (None, 256) output: (None, 64)
legtyp_out: Dense input: (None, 64) output: (None, 3)
conv2d_24: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_25: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization_12: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d_12: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d_8: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_26: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_27: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_13: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_13: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_9: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_28: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_29: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_14: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_14: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten_4: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense_8: Dense input: (None, 85760) output: (None, 256)
dropout_4: Dropout input: (None, 256) output: (None, 256)
dense_9: Dense input: (None, 256) output: (None, 64)
legcol_out: Dense input: (None, 64) output: (None, 11)
conv2d_30: Conv2D input: (None, 268, 160, 3) output: (None, 268, 160, 8)
conv2d_31: Conv2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
batch_normalization_15: BatchNormalization input: (None, 268, 160, 8) output: (None, 268, 160, 8)
spatial_dropout2d_15: SpatialDropout2D input: (None, 268, 160, 8) output: (None, 268, 160, 8)
max_pooling2d_10: MaxPooling2D input: (None, 268, 160, 8) output: (None, 134, 80, 8)
conv2d_32: Conv2D input: (None, 134, 80, 8) output: (None, 134, 80, 16)
conv2d_33: Conv2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
batch_normalization_16: BatchNormalization input: (None, 134, 80, 16) output: (None, 134, 80, 16)
spatial_dropout2d_16: SpatialDropout2D input: (None, 134, 80, 16) output: (None, 134, 80, 16)
max_pooling2d_11: MaxPooling2D input: (None, 134, 80, 16) output: (None, 67, 40, 16)
conv2d_34: Conv2D input: (None, 67, 40, 16) output: (None, 67, 40, 32)
conv2d_35: Conv2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
batch_normalization_17: BatchNormalization input: (None, 67, 40, 32) output: (None, 67, 40, 32)
spatial_dropout2d_17: SpatialDropout2D input: (None, 67, 40, 32) output: (None, 67, 40, 32)
flatten_5: Flatten input: (None, 67, 40, 32) output: (None, 85760)
dense_10: Dense input: (None, 85760) output: (None, 256)
dropout_5: Dropout input: (None, 256) output: (None, 256)
dense_11: Dense input: (None, 256) output: (None, 64)
luggage_out: Dense input: (None, 64) output: (None, 3)