Anmol Chhabria – achhabria7
Assignment #1
CS 4641

Supervised Learning Assignment

Datasets

Movies. One of the datasets being examined in this report is an IMDB dataset containing attributes from 5,000 different movies. These movies have release dates within the past 100 years and pertain to 66 different countries. The dataset included the IMDB score of the movie. Based on rankings in IMDB, movies that are on the top of the charts have a rating of 6 or above. The classification variable in this dataset is whether the IMDB score rating is above a 6 or not. A subset of 14 attributes was used to predict the IMDB score classification. This subset removed variables that had a high number of unique instances (actor names, director names, country, title year) and variables that had the same instance (aspect ratio, color, face number in poster). For example, there were 2600 unique instances of directors, and for a dataset with 5,000 instances this variable would not be significant.

MOOC EEG Signals. The other dataset examined in this report is from a study conducted on 10 college students while they watched MOOC videos. This dataset analyzes the electroencephalogram (EEG) signals in a student's brain while the student watches videos. The data has approximately 12,000 instances. A total of 13 attributes were collected during the study. The purpose of the study was to predict if the student was confused while watching the video or not. The target classification variables were two: a prediction whether the student was confused and a self-reported answer from the student stating whether they were confused or not. The classification variable used in the algorithms below is the self-reported data from the student- whether the student was confused or not. The prediction of the confusion classifier is not used as an attribute ('Col14' in README.txt). A total of 13 attributes were used to predict the classification variable.

Why are the datasets interesting in the real-world?

Movies. Predicting whether a movie will do well after releasing in the box office is useful for stores that sell DVDs and for services like Netflix. This allows buyers of movie licenses to predict how popular the movie will continue to be amongst consumers. A popular central for movies is IMDB, which ranks movies from around the world.

MOOC EEG Signals. MOOCs are becoming increasingly important in education in today's era. Being able to understand how confused a student is while looking at a video can be leveraged in online education. This data can be used to understand what particularly leads to confusion with respect to a video. In addition, this can even use to understand which EEG signal features can predict confusion.

Why are the datasets interesting in machine learning?

Movies. The movie dataset performs similarly for all algorithms. For all algorithms, the accuracy is within a range of 50%-77%. None of the supervised learning algorithms performed exceptionally on this dataset. This could imply that more data instances were needed, since the movie variety in different countries is so wide. The initial dataset had director and actor names, and these variables were excluded. Because the movie industry spans so many countries and actors, these variables could influence the dataset classifier if there were many more instances in the dataset from IMDB.

MOOC EEG Signals. The MOOC dataset performs differently across all supervised algorithms. It ranges from 67%-99% in test accuracy. It is interesting to see which algorithms perform better or worse for the dataset with EEG signals. This results in different learning curves and model complexities.

Algorithm Experiments

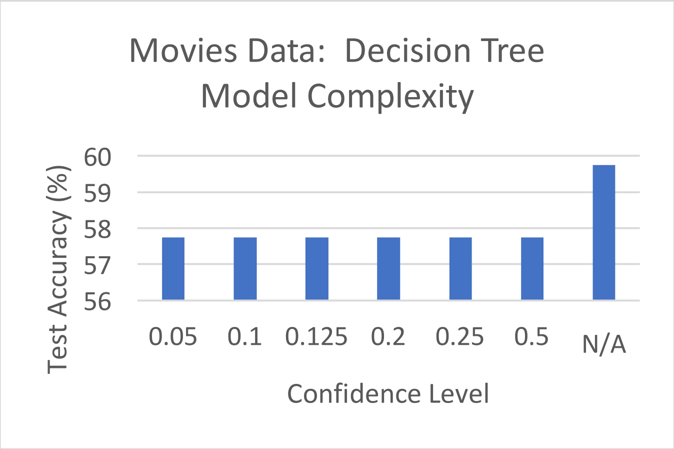
For all the algorithms, the iterations were run with a 30%/70% split for testing and training data, respectively and with a 10-fold cross-validation. The testing accuracy and error from both runs are indicated in the tables. After changing the hyperparameters for each algorithm, the model resulting from the hyperparameters with the highest cross-validation accuracy was indicated as optimal. These are highlighted in yellow in the results tables. A learning curve for the optimal hyperparameters was obtained. The x-axis of the learning curves represents the split % of the training dataset. The training dataset is 70% of the original dataset. The training errors were calculated for the respective splits. The testing error is calculated on the remaining 30% of the original dataset. The testing and training error for the algorithms is labeled on the graphs.

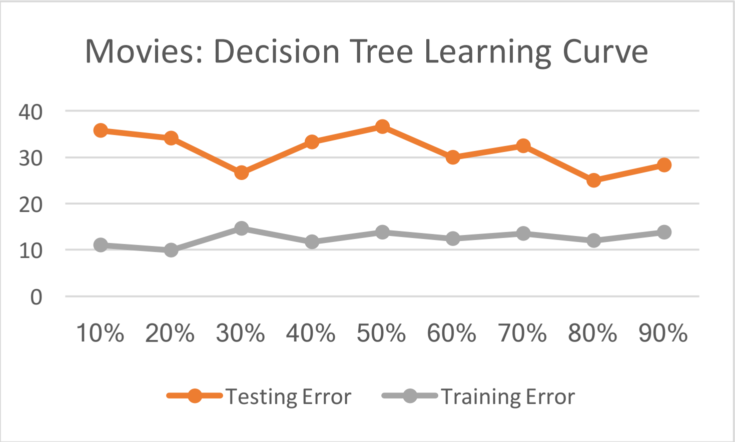
Decision Trees

The hyperparameters for decision trees were the confidence factor of the pruning if prune state was true. The training times were much higher than testing times for the decision trees. This is evidence of the fact that decision trees are eager learners.

Movies. For the movies data, the C4.5 decision tree algorithm was used. The C4.5 is an extension of the ID3 algorithm. After looking at the model complexity, the algorithm that performed the best was the unpruned tree. The unpruned tree, as expected, has a larger tree size than all the pruned trees. Unpruned trees are prone to overfitting, so this might be a reason for the higher cross-validation accuracy. The pruned trees yielded the same testing accuracy and error for all confidence levels. The decision tree algorithm does not perform very well on the Movie dataset, giving a maximum accuracy of 59.7%. The cross-validation and test accuracies for all confidence levels were the same for all pruning confidence levels. This is the worst performing algorithm for the Movies dataset.

The learning curve shows a downward trend in the testing error and a steady training error. The gap between both graphs tends to decrease. This means that there is not much overfitting. The test error is still very high.

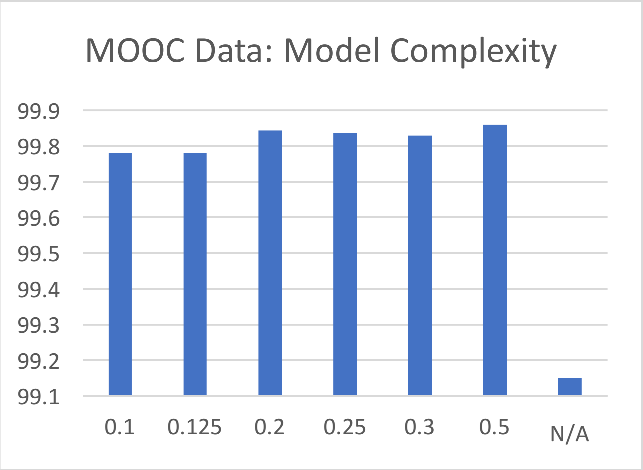


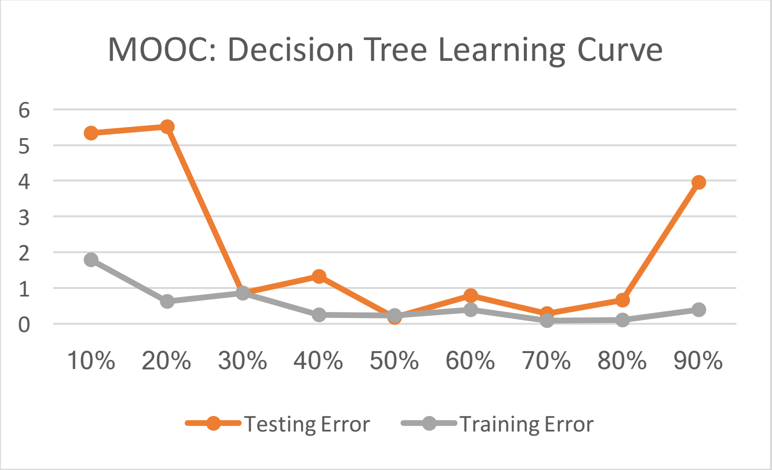


Prune State	Confidence	Leaves	Tree Size	CV Err	CV Acc	Test Error	Test Acc	Train Time	Test Time
Pruned	0.05	2	3	42.2566	57.7434	44.1507	55.8493	0.03 sec	0 sec
Pruned	0.1	2	3	42.2566	57.7434	44.1507	55.8493	0.04 sec	0 sec
Pruned	0.125	2	3	42.2566	57.7434	44.1507	55.8493	0.03 sec	0 sec
Pruned	0.2	2	3	42.2566	57.7434	44.1507	55.8493	0.03 sec	0 sec
Pruned	0.25	2	3	42.2566	57.7434	44.1507	55.8493	0.03 sec	0 sec
Pruned	0.5	2	3	42.2566	57.7434	44.1507	55.8493	0.03 sec	0 sec
Unpruned	N/A	597	860	40.2538	59.7462	42.3662	57.6338	0.03 sec	0 sec

MOOC EEG Signals. For the EEG MOOC dataset, the C4.5 decision tree algorithm was used as well. The best performing tree was a pruned tree with confidence level of 0.5 This indicates not much pruning, since a lower confidence level would indicate more pruning. However, it still implies there is some noise in the data and the pruning reduces the overfitting. The decision tree performs much better on the MOOC dataset than on the Movies dataset, and this could be due to the larger size of the dataset. The number of attributes is similar across both datasets, so the error must come from the dataset size.

The learning curve, for the most part, shows a decreasing testing error as the size of the training dataset increases. The training error decreases and remains consistent. This could imply not much overfitting in the model. The optimal model for the decision tree could be obtained by training over 35% of the original dataset.



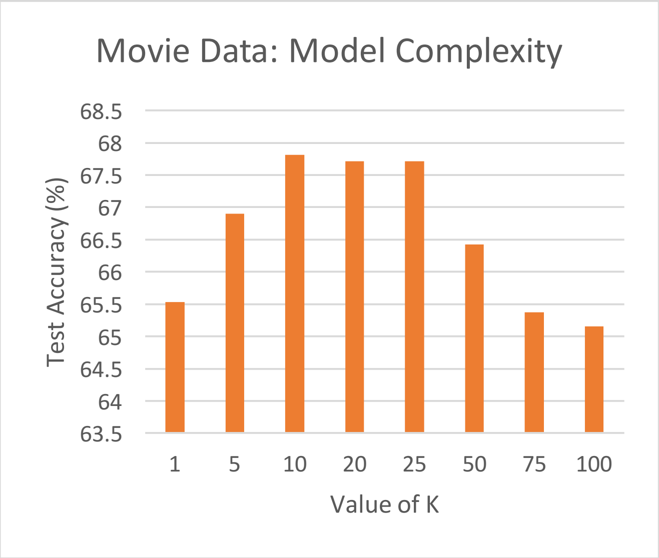


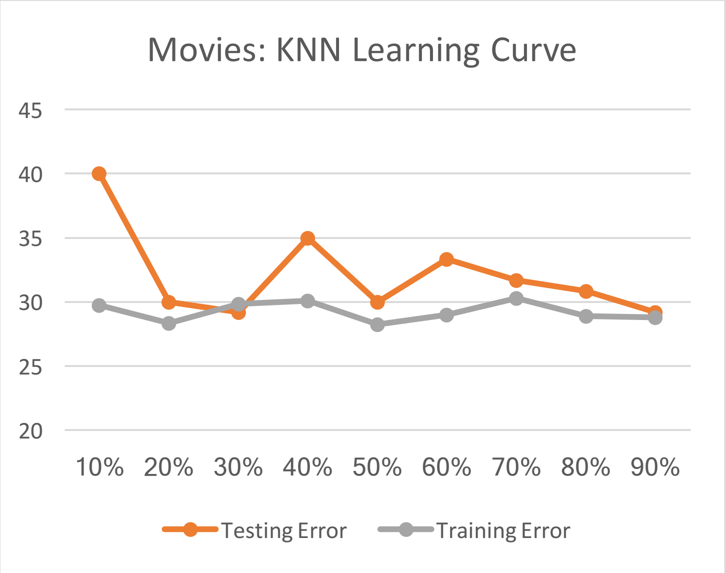
Prune State	Confidence	Leaves	Tree Size	CV Err	CV Acc	Test Error	Test Acc	Train Time	Test Time
Pruned	0.1	53	105	0.2186	99.7814	0.4424	99.5760	0.34 sec	0 sec
Pruned	0.125	53	105	0.2186	99.7814	0.4424	99.5576	0.34 sec	0 sec
Pruned	0.2	55	109	0.1561	99.8439	0.4424	99.5760	0.32 sec	0 sec
Pruned	0.25	58	115	0.1639	99.8361	0.4163	99.5837	0.31 sec	0 sec
Pruned	0.3	58	115	0.1717	99.8283	0.4163	99.5837	0.34 sec	0 sec
Pruned	0.5	58	115	0.1405	99.8595	0.4163	99.5837	0.33 sec	0 sec
Unpruned	N/A	189	377	0.8508	99.1492	1.0929	98.9071	0.23 sec	0 sec

K-Nearest Neighbors

The hyperparameter for KNN was k, the number of clusters. The training times for the K-NN algorithms run for both datasets are nearly 0, whereas the testing times are much higher. This is evidence of the fact that KNN is a lazy learning algorithm, which approximates the target function locally.

Movies. For the Movies dataset, the K-NN algorithm performs best when k is 10. As the number of clusters increases, the cross-validation and test error increase. This implies that more data results in overfitting of the algorithm.

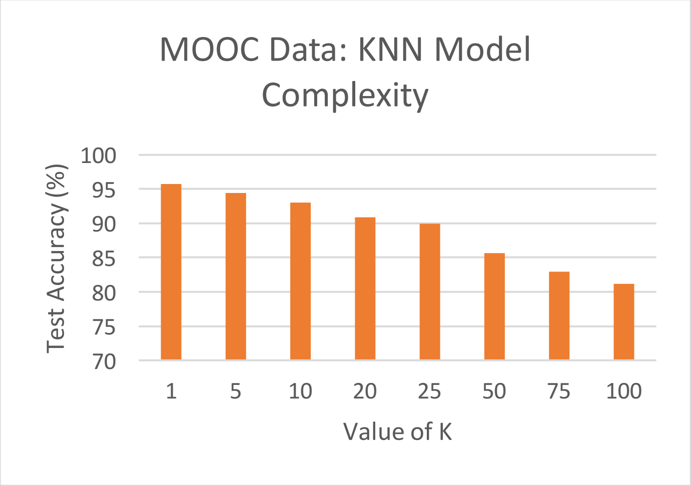


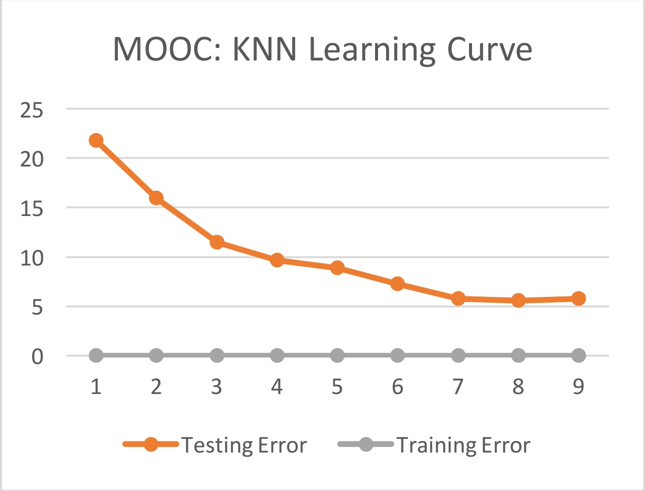


Number of Neighbors, K	CV Err	CV Acc	Test Error	Test Acc	Training Time	Test Time
1	34.4636	65.5364	34.3027	65.6973	0	0.26
5	33.0954	66.9046	34.8976	65.1024	0	0.3
10	32.1832	67.8168	33.3113	66.6887	0	0.32
20	32.2824	67.7176	31.8572	68.1428	0	0.37
25	32.2824	67.7176	32.3199	67.6801	0	0.39
50	33.5713	66.4287	34.9636	65.0364	0	0.39
75	34.6222	65.3778	35.889	64.111	0	0.4
100	34.8404	65.1596	36.5499	63.4501	0	0.41

The learning curve for the KNN algorithm shows a decreasing test error from 40% to 29%, and a training error that remains steady around 29%. From this graph, it can be concluded that there is reduced overfitting in the dataset as the test and train errors follow a decreasing trend.

MOOC EEG Signals. For the MOOC data, the best KNN algorithm is that when K is 1. As the number of clusters increases, the accuracy of the algorithm decreases. This is a similar trend to that of the Movies data. The learning curve for the optimal K shows that there is a large gap between the test error and the train error. Even though the test error decreases, there is some overfitting. The training error is 0% throughout all splits in the training dataset, which indicate overfitting.



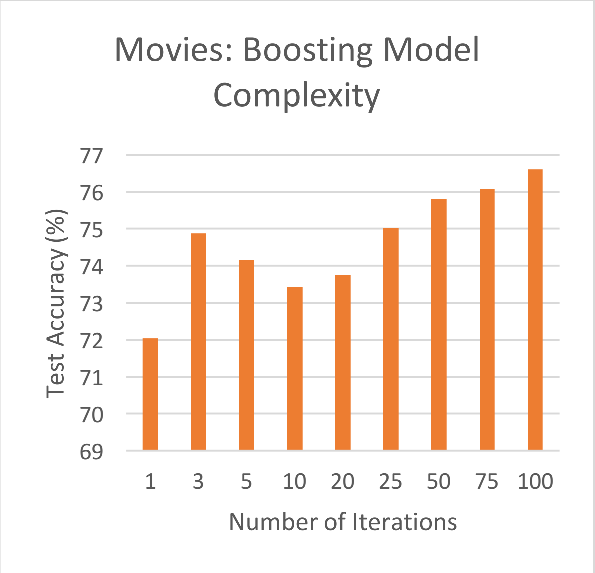


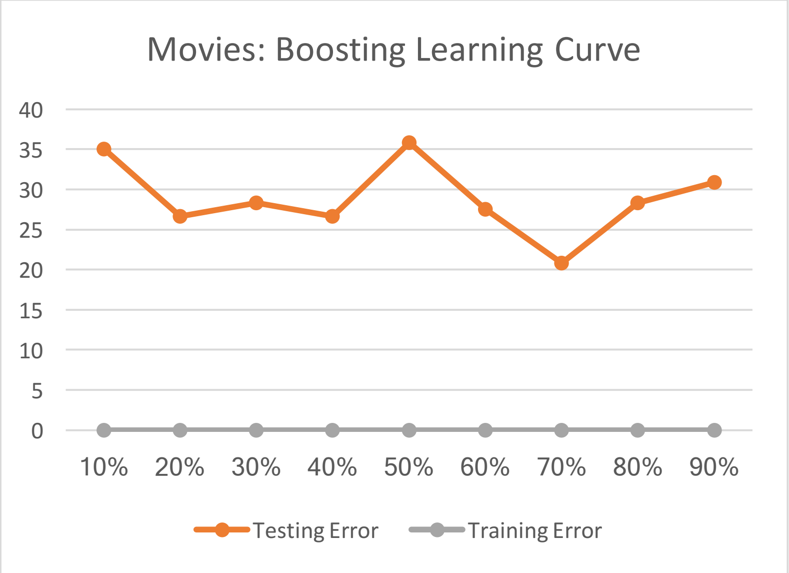
K	CV Err	CV Acc	Test Error	Test Acc	Training Time	Test Time
1	4.3088	95.6912	5.4905	94.5095	0.01	1.47
5	5.5968	94.4032	6.4533	93.5467	0.01	1.63
10	6.9550	93.0450	8.1447	91.8553	0.00	1.76
20	9.1094	90.8906	10.5647	89.4353	0.00	1.90
25	10.1007	89.8993	12.0739	87.9261	0.00	2.10
50	14.3705	85.6295	16.2373	83.7627	0.01	2.35
75	17.0400	82.9600	18.5272	81.4728	0.00	2.54
100	18.8276	81.1724	21.4416	78.5584	0.01	2.63

Boosting

The hyperparameter for boosting was the number of iterations. The boosting algorithm was run for the optimal decision tree obtained for each dataset. For the Movies dataset, the optimal tree was unpruned C4.5 tree. For the MOOC dataset, the optimal tree was a C4.5 tree with a confidence factor of 0.5 for pruning.

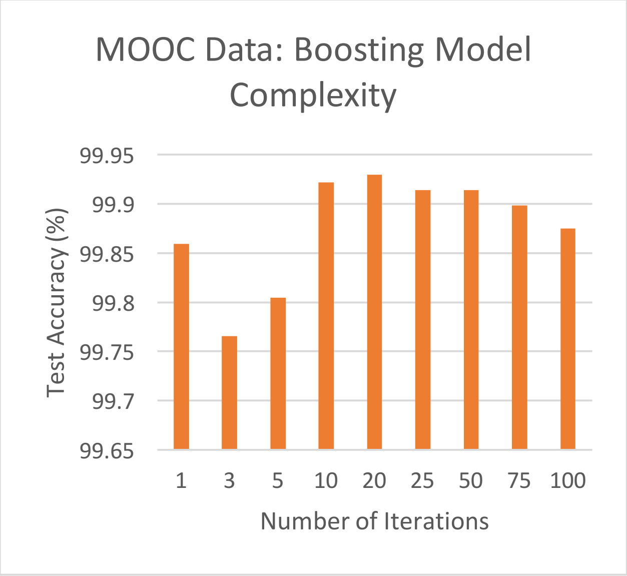
Movies. The best performing boosting algorithm was that with 100 iterations. The learning curve for 100 iterations shows a very large gap between the testing and training error, which implies that there is a lot of overfitting in the model. Another indication of overfitting is the training error of 0% in the data throughout all values of the split % of the training dataset. Despite the cross-validation accuracy of 77%, and improved model would reduce overfitting and different parameters.

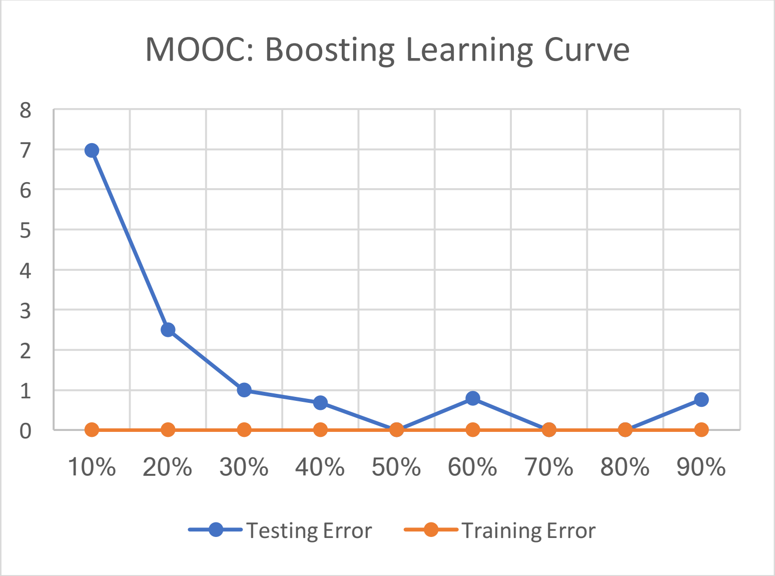




Iterations	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time(sec)	Test Time(sec)
1	26.9879	73.0121	27.9577	72.0423	0.13	0
3	26.8094	73.1906	25.1157	74.8843	0.5	0.01
5	26.1154	73.8846	25.8427	74.1573	0.8	0.01
10	25.9766	74.0234	26.5697	73.4303	1.81	0.02
20	24.9851	75.0149	26.2393	73.7607	4.12	0.04
25	24.1523	75.8477	24.9835	75.0165	4.46	0.04
50	23.3591	76.6409	24.1904	75.8096	9.4	0.09
75	23.0617	76.9383	23.926	76.074	13.92	0.33
100	22.903	77.0970	23.3972	76.6028	19.59	0.29

MOOC EEG Signals. The best boosting model for the MOOC dataset was obtained with 20 iterations. The learning curve shows the test and training error for 20 iterations. The learning curve shows that overall the testing error reduces as the training data split % increases. However, the training error is 0% throughout the curve. This is an indication of overfitting. With a training data split at and above 50%, the optimal model is obtained. The optimal model can be trained with 35% of the original dataset. Boosting results in overfitting for both datasets, but it has a higher accuracy for the MOOC dataset.





Iterations	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
1	0.1405	99.8595	0.4163	99.5837	0.37	0
3	0.2342	99.7658	0.4684	99.5316	1.55	0.01
5	0.1951	99.8049	0.5725	99.4275	3.59	0.01
10	0.0781	99.9219	0.2602	99.7398	5.88	0.03
20	0.0703	99.9297	0.1041	99.8959	15.28	0.07
25	0.0859	99.9141	0.1821	99.8179	20.66	0.09
50	0.0859	99.9141	0.4684	99.5316	57.87	0.2
75	0.1015	99.8985	0.2082	99.7918	102.05	0.33
100	0.1249	99.8751	0.1041	99.8959	148.34	0.38

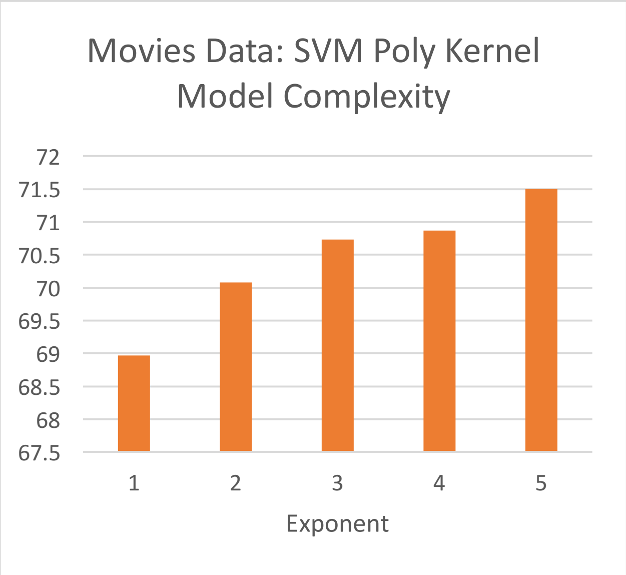
Support Vector Machines

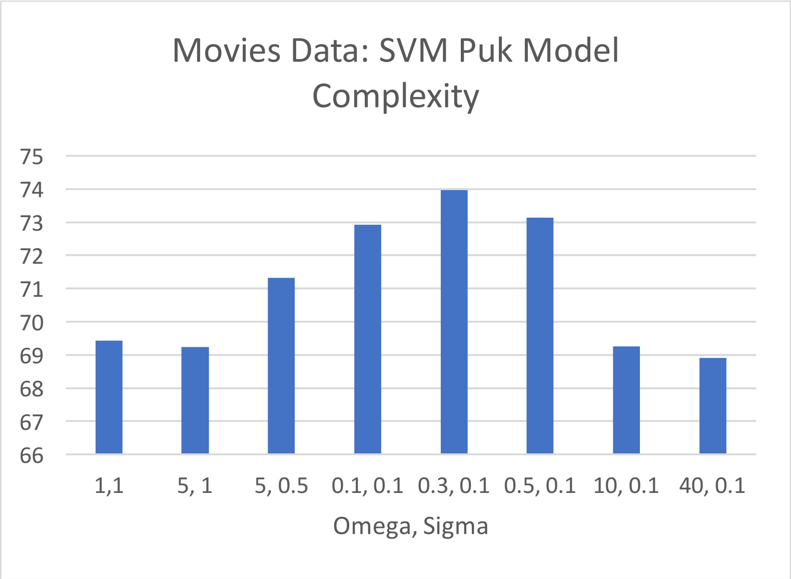
The hyperparameter for support vector machines depended on the kernel used. For both datasets, the Polynomial kernel (Poly) and the Pearson VII function-based universal kernel (Puk) were run. For the polynomial kernel the values changed are the exponent. For Puk, the values changed are the sigma and omega values. The training times for the SVM algorithms are much higher than the testing times. This is expected given that the algorithm is an eager learning algorithm.

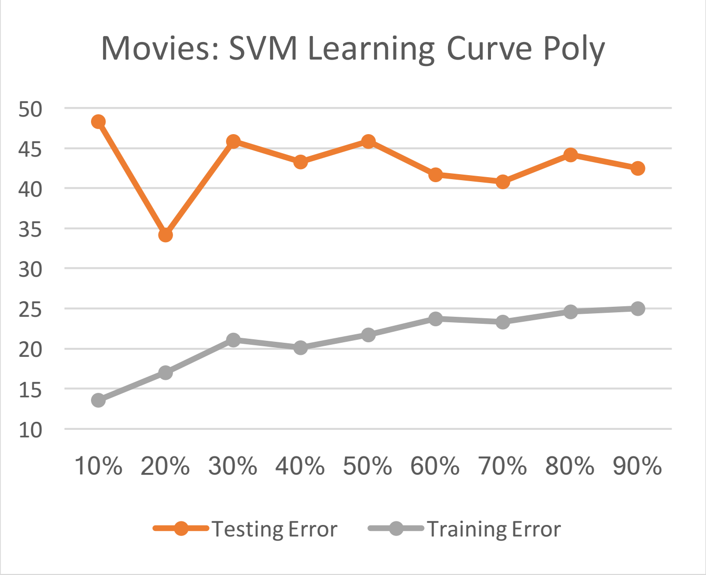
Movies. For the movies, the Polynomial kernel achieved the highest cross-validation accuracy with an exponent of 5. The higher the exponent, the higher the accuracy for the Polynomial, and this can be explained by the fact that a higher degree polynomial function can separate the classes better.

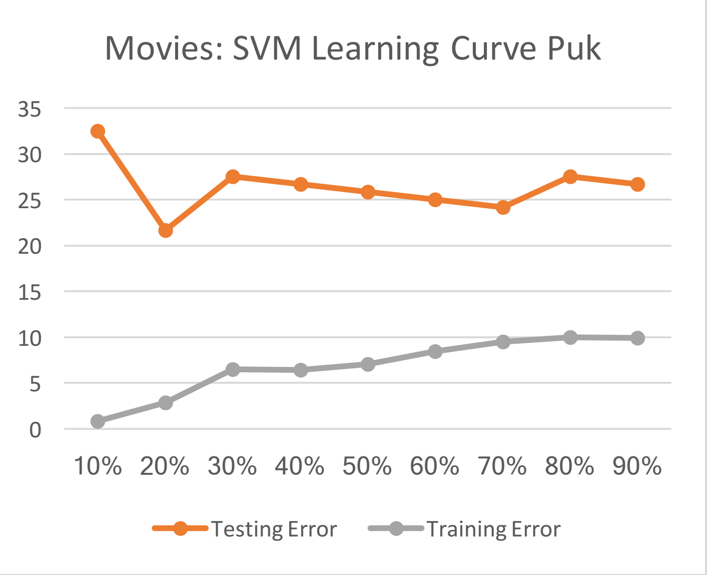
The SVM algorithm with Puk performed slightly better than with the Polynomial kernel. The optimal model had an omega of 0.3 and a sigma of 0.1. It can be observed that a lower sigma value yields better cross-validation and test accuracy by comparing the run with values (5, 1) and (5, 0.5). A smaller sigma leads to a narrow steep decay function which also implies that for the same dataset a high power of Polynomial kernel can classify the data better. With high values of omega, the model does not perform as well as with low values. This indicates that the data has an intrinsic linear relationship. The lowest test error in the learning curve is obtained when the 14% of the training data is used. Using more data than 20% of the training split results in overfitting.

Kernel	Exponent, Omega/Sigma	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
Poly	1	31.0331	68.9669	31.2624	68.7376	1.67	0.04
Poly	2	29.9227	70.0773	30.7997	69.2003	55.26	0.56
Poly	3	29.2683	70.7317	30.1388	69.8612	132.02	0.87
Poly	4	29.1295	70.8705	30.4032	69.5968	162.93	1.31
Poly	5	28.4949	71.5051	29.7422	70.2578	239.47	1.02
Puk	1,1	30.577	69.423	30.4032	69.5968	135.91	3.11
Puk	5, 1	30.7555	69.2445	31.2624	68.7376	72.12	1.8
Puk	5, 0.5	28.6734	71.3266	28.6186	71.3814	121.41	3.25
Puk	0.1, 0.1	27.0672	72.9328	27.4289	72.5711	103.42	3.99
Puk	0.3, 0.1	26.0361	73.9639	26.5697	73.4303	89.78	3.49
Puk	0.5, 0.1	26.8689	73.1311	26.5697	73.4303	135.08	4.09
Puk	10, 0.1	30.7365	69.2635	30.6604	69.3396	1.7	0.11
Puk	40, 0.1	31.0907	68.9093	32.0755	67.9245	2.07	0.1









The learning curves for the movies with the Polynomial kernel and Puk look similar in terms of trends. The learning curve for the Polynomial kernel shows an increase in test and training error. The test error oscillates a lot more for the Polynomial kernel than for the Puk. The test and training error graphs show a general upward trend.

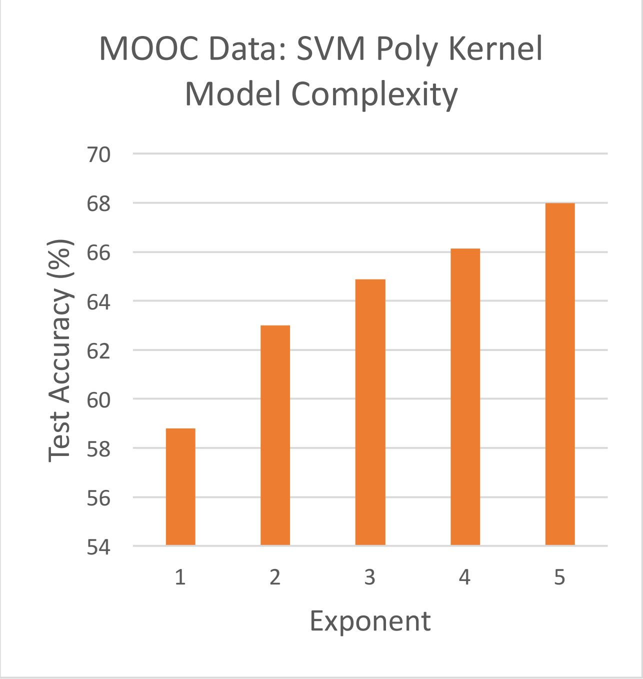
However, Puk yields a higher accuracy for the test and cross-validation. The Puk learning curve tends toward a consistent gap between the train and test errors, which could mean that there is bias in the training data. The test error for the Puk shows a general downward trend with some variation towards the end. This indicates that there is a low chance of overfitting.

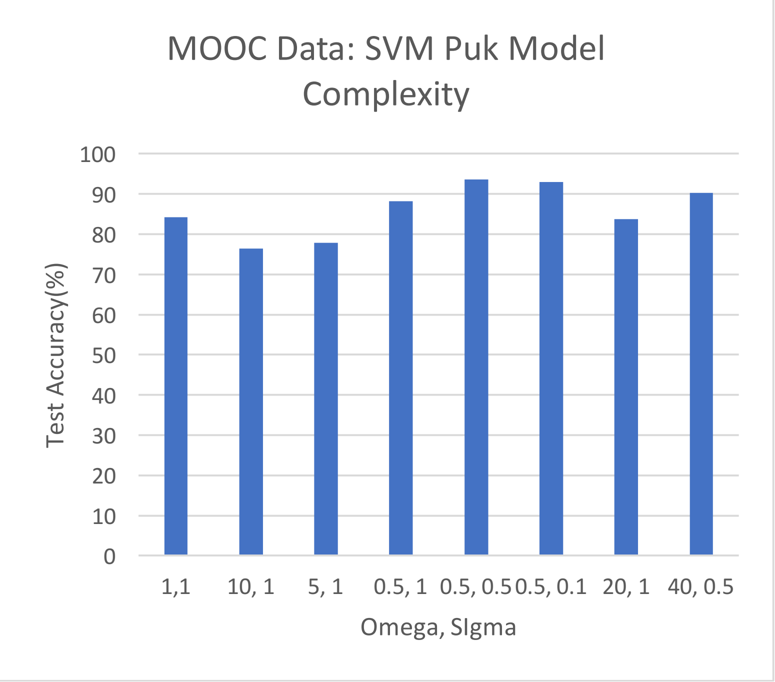
Based on cross-validation accuracy and learning curve results, the SVM algorithm with the Puk provides the most reliable results for the Movies dataset.

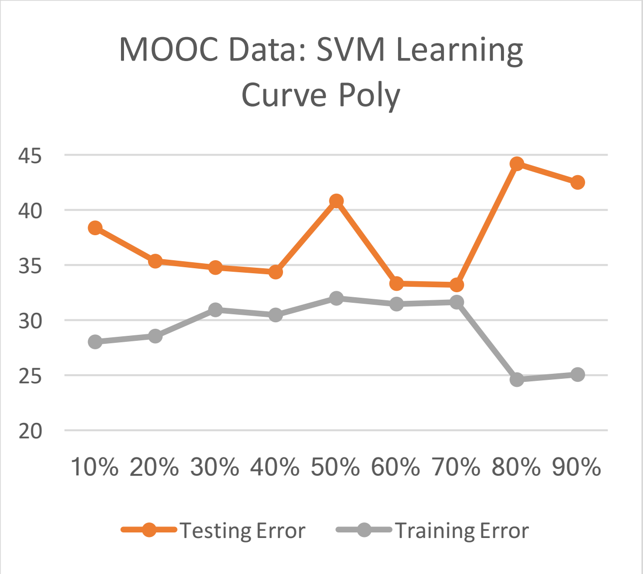
MOOC EEG Signals. For the MOOC dataset, the SVM algorithm with a Polynomial kernel performed best with an exponent of the degree 5. A higher exponent, as mentioned before, allows for the algorithm to classify the data more accurately.

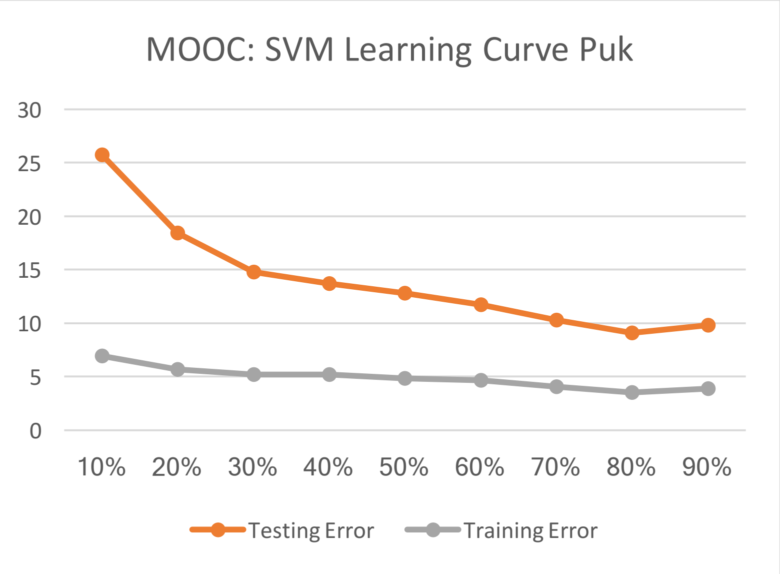
There is a higher increase in the cross-validation accuracy when using a Puk kernel instead of a polynomial kernel. The omega, sigma values (0.5, 0.5) yield the highest accuracy for the MOOC dataset. Higher omega values do not yield as high a cross-validation accuracy as 93%. As expected changing the sigma value has a higher impact than changing the omega value, if you compare change in accuracy for runs (0.5, 1) and (0.5, 0.5) with the change in accuracy for runs (1, 1) and (0.5, 1).

Kernel	Exponent	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
Poly	1	41.199	58.801	41.3999	58.6001	1.56	0.01
Poly	2	37.0073	62.9927	37.4447	62.5553	522.04	3.1
Poly	3	35.1261	64.8739	36.1436	63.8564	1162	3.36
Poly	4	33.8693	66.1307	34.9467	65.0533	1543.71	3.37
Poly	5	32.0037	67.9963	33.4634	66.5366	3919.76	4.44
Puk	1,1	15.7755	84.2245	17.3302	82.6698	336.61	6
Puk	10, 1	23.5501	76.4499	24.9805	75.0195	545.27	6.69
Puk	5, 1	22.1684	77.8316	24.0697	75.9303	655.92	8.04
Puk	0.5, 1	11.7243	88.2757	12.9066	87.0934	623.93	5.98
Puk	0.5, 0.5	6.3851	93.6149	7.9105	92.0895	313.16	5.38
Puk	0.5, 0.1	7.0252	92.9748	8.587	91.413	533.01	8.36
Puk	20, 1	16.2113	83.7887	25.9953	74.0047	461.91	7.27
Puk	40, 0.5	9.7026	90.2974	11.0851	88.9149	238.06	3.44









The learning curve for the MOOC data with the Polynomial kernel converge. Because the kernel uses a high exponent, it trains very well when a high split% of the training dataset is used. This leads to overfitting and a high test error as shown with the 80% and 90% splits of the data.

The learning curve for the MOOC data with the Puk shows a downward trend in the test and train error. This a sign of no overfitting, and it shows that the Puk algorithm generalizes well for the dataset. The optimal model can be obtained by training the algorithm with 56% of the overall dataset.

After considering the learning curve and the accuracy rate, the Puk kernel for Support Vector Machines is the best performing algorithm for MOOC dataset.

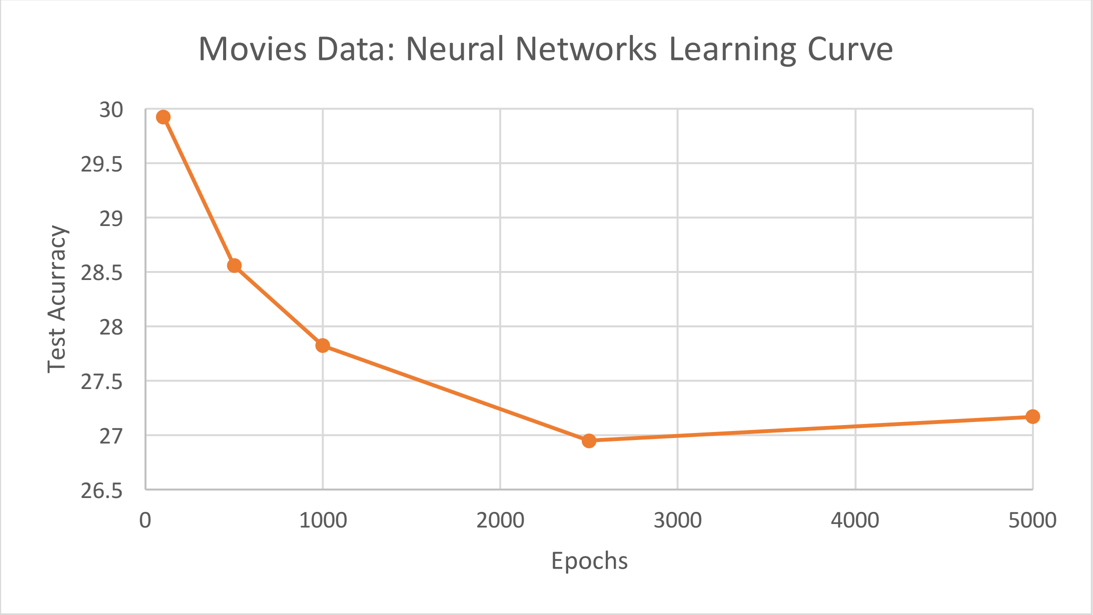
Artificial Neural Networks

The hyperparameter for the neural networks were the momentum and learning rate. The trials to find the optimal momentum and learning rate were run for a training time of 500 epochs. After finding the optimal momentum and learning rate, a learning curve with training times of 100, 500, 1000, 2500, and 5000 was run on the dataset.

The training times for the artificial neural networks are much higher than the testing times for the artificial neural networks. This makes sense since it is an eager learner.

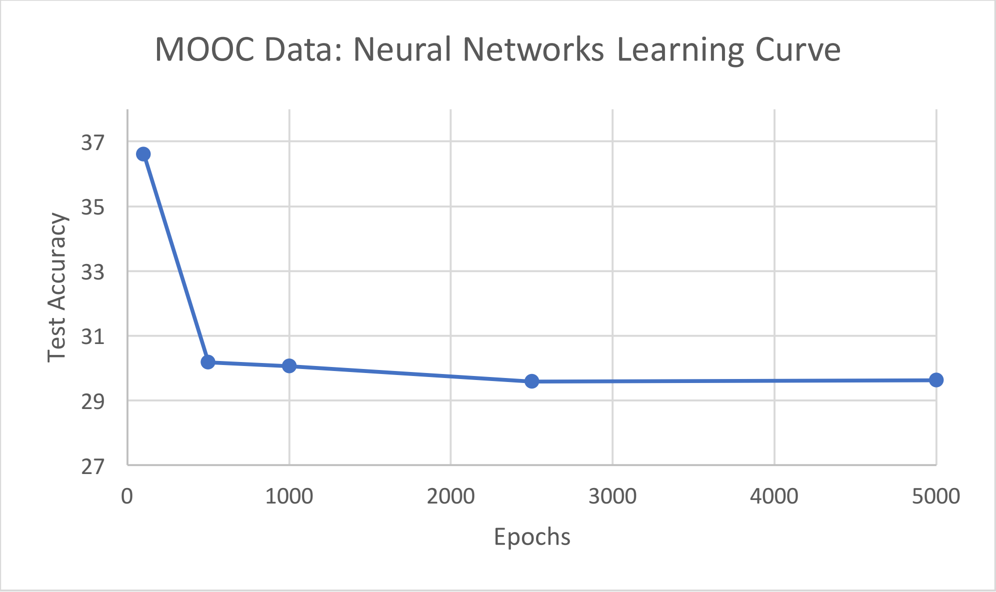
Movies. For the movies dataset, the optimal momentum and learning rate were 0.2 and 0.05 respectively. Despite these being the optimal hyperparameter values, the cross-validation error and the test error are high, close to 29%. From the learning curve, we can observe that as the number of Epochs increases, the test error reduces. At a training time of 2500 epochs, the algorithm produces the lowest error. The graph does not suggest that there is overfitting.

Momentum	Learning	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
0.2	0.3	31.7073	68.2927	32.6504	67.3496	199.92	0.07
0.2	0.4	41.741	58.259	50.1652	49.8348	230.86	0.08
0.2	0.2	29.5459	70.4541	31.1963	68.8037	291.77	0.07
0.2	0.1	29.2286	70.7714	30.0066	69.9934	271.66	0.05
0.2	0.05	28.5544	71.4456	29.9405	70.0595	121.32	0.06
0.3	0.05	29.0303	70.9697	29.7422	70.2578	1106.44	0.06



MOOC EEG Signals. For the MOOC dataset, the optimal momentum and learning rate were 0.1 and 0.1, respectively. As in the Movies dataset, the cross validation and test error are close to 30%, which is very high. However, the learning curve for the MOOC dataset looks different. A training time of 5000 epochs and 2500 epochs yield the same result. For the MOOC dataset, the highest accuracy with least training time would result when the training time is 2500 epochs. The graph does not suggest that there is overfitting. Based on cross-validation accuracy, the neural networks are the worst performing algorithm for MOOC dataset.

Momentum	Learning	CV Err (%)	CV Acc (%)	Test Error(%)	Test Acc(%)	Training Time	Test Time
0.2	0.3	40.2311	59.7689	42.9612	57.0388	33.51	0.02
0.2	0.4	42.1122	57.8878	45.2251	54.7749	25.03	0.01
0.2	0.2	39.9422	60.0578	41.7122	58.2878	25.65	0.01
0.2	0.1	30.2396	69.7604	22.8467	77.1533	31.97	0.01
0.3	0.1	31.4417	68.5583	27.8689	72.1311	37.76	0.04
0.1	0.1	30.185	69.815	28.2852	71.7148	18.22	0.01
0.2	0.15	35.0402	64.9598	40.3591	59.6409	22.2	0.03
0.4	0.1	33.924	66.076	39.5004	60.4996	21.17	0.01



References

Abakar, Khalid. "The Performance of Data Mining Techniques in Prediction of Yarn Quality." International Journal of Innovation, Management and Technology (2013): n. pag. Web.

"AKSW." SlideWiki: Authoring Platform for OpenCourseWare. N.p., n.d. Web. 05 Feb. 2017.

"EEG Brain Wave for Confusion | Kaggle." EEG Brain Wave for Confusion | Kaggle. N.p., n.d. Web. 05 Feb. 2017.

"Event Tree and Decision Tree Analysis." Risk Assessment (2012): 163-80. Web.

"Facilitating the Application of Support Vector Regression by Using a Universal Pearson VII Function Based Kernel." Facilitating the Application of Support Vector Regression by Using a Universal Pearson VII Function Based Kernel. N.p., n.d. Web. 05 Feb. 2017.

"IMDB 5000 Movie Dataset | Kaggle." IMDB 5000 Movie Dataset | Kaggle. N.p., n.d. Web. 05 Feb. 2017.

"Top Rated Movies." IMDb. IMDb.com, n.d. Web. 05 Feb. 2017.