February 20, 2018

Problem Set 3

Sanchit Singhal

INF 385T – Introduction to Machine Learning with Danna Gurari

Spring 2018

School of Information

1. **Data Clean-Up**
    a. **One-Hot Encoding**

Dataset	Type_Comedy	Type_Drama	Length_Short	Length_Medium	Length_Long	IMDB	Liked
Train	1	0	1	0	0	7.2	Yes
Train	0	1	0	1	0	9.3	Yes
Train	1	0	0	1	0	5.1	No
Train	0	1	0	0	1	6.9	No
Train	0	1	0	1	0	8.3	Yes
Train	0	1	1	0	0	4.5	No
Train	1	0	1	0	0	8.0	Yes
Train	0	1	0	1	0	7.5	Yes
Train			0	1	0	6.7	Yes
Train	1	0				3.6	Yes
Train	0	1	0	0	1		Yes
Test	0	1	0	1	0	8.1	Yes
Test	1	0	1	0	0		Yes
Test			0	0	1	7.5	Yes

    b. **Imputing Missing Values**

Mean IMDB , Class Liked=Yes : (7.2+9.3+8.3+8.0+7.5+6.7+3.6)/7 = 7.2
Mean IMDB, Class Liked=No : (5.1+6.9+4.5)/3 = 5.5

Mean Type_Comedy, Class Liked=Yes : (1+0+0+1+0+1+0)/7 = 0.43; Class Liked=No : (1+0+0)/3 = 0.33
Mean Type_Drama, Class Liked=Yes : (0+1+1+1+0+1)/7 = 0.57; Class Liked=No : (1+1+0)/3 = 0.66

Mean Length_Short, Class Liked=Yes : (1+0+0+1+0+0+0)/7 = 0.29; Class Liked=No : (0+0+1)/3 = 0.33
Mean Length_Medium, Class Liked=Yes : (0+1+1+0+1+1+0)/7 = 0.43; Class Liked=No : (1+0+0)/3 = 0.33
Mean Length_Long, Class Liked=Yes : (0+0+0+0+0+0+1)/7 = 0.14; Class Liked=No : (1+0+0)/3 = 0.33

Dataset	Type_Comedy	Type_Drama	Length_Short	Length_Medium	Length_Long	IMDB	Liked
Train	1	0	1	0	0	7.2	Yes
Train	0	1	0	1	0	9.3	Yes
Train	1	0	0	1	0	5.1	No
Train	0	1	0	0	1	6.9	No
Train	0	1	0	1	0	8.3	Yes
Train	0	1	1	0	0	4.5	No
Train	1	0	1	0	0	8.0	Yes
Train	0	1	0	1	0	7.5	Yes
Train	**0.43**	**0.57**	0	1	0	6.7	Yes
Train	1	0	**0.29**	**0.57**	**0.14**	3.6	Yes
Train	0	1	0	0	1	**7.2**	Yes
Test	0	1	0	1	0	8.1	Yes
Test	1	0	1	0	0	**7.2**	Yes
Test	**0.43**	**0.57**	0	0	1	7.5	Yes

c. Feature Scaling

Min = 3.6, Max = 9.3
Mean = 6.7, Variance = 2.9

Dataset	IMDB x	IMDB min-max scaled (x-min) / (max-min)	IMDB standardized (x-mean)/variance	Liked
Train	7.2	0.63	0.17	Yes
Train	9.3	1.00	0.90	Yes
Train	5.1	0.26	-0.55	No
Train	6.9	0.58	0.07	No
Train	8.3	0.82	0.55	Yes
Train	4.5	0.16	-0.76	No
Train	8.0	0.77	0.45	Yes
Train	7.5	0.68	0.28	Yes
Train	6.7	0.54	0.00	Yes
Train	3.6	0.00	-1.07	Yes
Train	**7.2**	**0.63**	**0.17**	Yes
Test	8.1	0.79	0.48	Yes
Test	**7.2**	**0.63**	**0.17**	Yes
Test	7.5	0.68	0.28	Yes

2. Dimensionality Reduction

a. Uses for dimensionality reduction techniques

- **i.** Faster training/testing of model when timing is crucial
- **ii.** Reduced risk of overfitting/ reduction of noise when data might not be ideal
- **iii.** Reduced need for expensive hardware for memory/diskspace when budget is low

b. PCA vs LLE

Principle Component Analysis and Locally Linear Embedding are both dimensionality reduction techniques. I would use PCA when I am assuming that the data is linearly separable. It is most useful when most of the data lies close to a linear subspace and therefore helps to eliminate irrelevant features through separating linearly. However, when the best representation of the data is not linear, PCA will not be able to help us because it will try to find a planar surface that describes the data and might miss out on important information with this approach. Locally Linear Embedding is a type of manifold learning algorithm that I would use when I do not think the underlying structure is linear. It works by assuming that the data points are low-dimensional manifolds that are embedded into a higher dimensional space and therefore attempts to describe the structure as a function of only a few parameters. Using this method, LEE are able to be much more effective at dimensionality reduction of non-linear data structures while not losing insight.

3. Classification Evaluation

True Positive (TP)	False Positive (FP)
False Negative (FN)	True Negative (TN)

a. PR Curve when designing a ML system

Precision = TP/(TP+FP) vs Recall = TP/(TP + FN)
A precision-recall (PR) curve shows the tradeoff between precision and recall for different thresholds. Since precision is the fraction of retrieved instances that are relevant and recall is the fraction of relevant instances that are retrieved, it is very useful to gage both measures to evaluate the information retrieval system. A large area under the curve indicates both high precision and high recall whereas a low area indicates low levels for both. Typically, it is difficult to return all the outcomes but only avoid bad results and therefore the two are generally inversely related. The designer needs to find a balance between the two based on the application of the model. The PR curve can be used to analyze and compare various parameters to achieve optimal results for the designer's task. The PR curve is useful when the positive class samples are very rare and the problem involved "searching for a needle in a haystack.

b. PR Curve and a Confusion Matrix

The confusion matrix are all the evaluation values for a given threshold. Although it is the basis for the PR curve, a confusion matrix is only one data point on the curve and therefore a PR curve can be thought of as an aggregation of confusion matrices on different thresholds. It is important to note that the PR curve does not factor in the TN values of each value.

c. ROC Curve when designing a ML system

FPR = FP/(FP + TN) vs TPR = TP(TP + FN)
A receiver operating characteristic (ROC) curve shows the tradeoff between sensitivity, also known as recall, and specificity. Recall is still the fraction of relevant instances that are retrieved but specificity is a measure of how many of the negative samples are identified as negative. ROC analysis also provides a method of selecting optimal models and disregard suboptimal thresholds so the right balance needs to be found based on the designer's targeted application of the model. Because a ROC curve remains the same regardless of the probability of the positive class, it is a better tool to apply when positive class samples are not very rare.

d. ROC Curve and a Confusion Matrix

Again, the confusion matrix is all the evaluation values for a given threshold that defines the points on the ROC curve and the curve is an accumulation of the various confusion matrices. However, the ROC curve considers all the values of the confusion matrix and therefore is a more holistic representation of the evaluation from the confusion matrix.

4. Challenge Analysis

Challenge 1 – Mercari Price Suggestion Challenge

a) Motivation

Product Pricing is hard to judge because it varies based on season, brand, specs, and a multitude of other factors. Developing an algorithm that automatically suggests the right prices will help online sellers determine competitive valuations of the products that they well

b) Machine Learning Task

Using user-inputted text descriptions that includes details about category, brand, and item condition, the algorithm must predict the price of the item in dollars.

c) Dataset Partitioning

The dataset is split into two partitions – train and test using a 50/50 split. The competition is divided into two stages though where stage 1 has 700K testing rows and stage 2 has the complete test dataset of 3.5M rows. The complete training data is provided at the start though – approximately 3.5M rows as well.

d) Evaluation Metrics Used

The evaluation metric for this competition is RMSLE (Root Mean Squared Logarithmic Error). The submission with the lowest score will win the competition.

Challenge 2 – House Prices

a) Motivation

Predicting prices for houses is difficult because there are so many factors that influence the results that it is hard for home buyers to estimate features that affect the sale price. Developing an algorithm that predicts house prices will help buyers estimate and evaluate their options when purchasing property.

b) Machine Learning Task

Based on the exact specifications of houses such as number of bedrooms, bathrooms, garage space, amenities, property condition, and so on, the algorithm must predict the price of the house.

c) Dataset Partitioning

Again the dataset is splits into two partitions – train and test using a 50/50 split. There are 1460 rows in each partition of the data.

d) Evaluation Metrics Used

The submissions are evaluated by the Root-Mean-Squared-Error (RMSE) between the log of the predicted value and the log of the observed sale price. Taking the log means that the errors in predicted both expensive and cheap houses will affect the results equally.