Machine Learning Report

Kevin Nguyen

Kevin Horning

Goal and Approach

The objective of the project was to predict different types of data using data that was given. We had to estimate the value of missing data, determine the classes of samples and predict future sales of stores. To do this we decided to conduct extensive tests on different machine learning models to determine which models performed the best for the given task.

For both the single and multilabel datasets, the given data was split into train and test data so that numerous models could compare their predicted classifications on the test data on the actual test labels. The accuracy of these predictions was shown in a comprehensive analysis report for each model.

For the sales prediction dataset, we decided to use a Long Short Term Memory Neural Network to determine the most likely store sales for the next month based on previous month sales.

Preprocessing

In order to run the different models on the given data, we had to solve the problem of missing data. In several of the datasets, there were certain outliers that did not fit in with the rest of the data. An example of this is shown below:

```
1       5       3       5       1       2       1       1       0
1       5       4       3       4       2       1       2       0
2       1.00000000000000e+99    5       4       1       4       3
1       3       4       4       1       5       1       1       1
1       5       4       3       4       5       1       1       0
1       5       2       5       1       2       1       1       0
2       1       5       3       5       5       3       4       0
2       3       4       4       5       5       1       1       0
1       5       2       3       1       3       1       2       0
2       5       2       3       6       1.00000000000000e+99    1
2       1       3       5       4       5       2       3       1
1       5       3       4       2       5       1       1       0
2       3       5       3       1       5       1       1       0
2       5       2       5       3       5       1       4       0
1       1       5       4       3       5       2       5       2
2       1.00000000000000e+99    7       3       8       5       1
1       5       3       5       1       2       1       1       0
```

These values are problematic because classification the models are not capable of processing numbers this large. Therefore, we either needed to remove the samples that included an outlier or impute these missing values. We decided on the latter to preserve the numerous samples that included missing data.

It was settled that filling a sample's missing values based on the values of the most similar samples would be the best method of imputation. We used the KNN model from the fancyimpute library to calculate the most similar samples, average their values for the feature at hand, and substitute the missing values with these averages. After all missing data values were replaced by appropriate estimations, the data was complete and ready for the classification models to use.

Choice of Models

We were uncertain of which model and parameters would produce the best prediction, so we conducted extensive experiments with different models. The Logistic Regression, Random Forest, Support Vector Machine, and K-Nearest Neighbors models were chosen to give a variety of classifier types. The parameters for each model were then optimized as we processed the models with different parameter values and selected the one that resulted in the best prediction accuracy.

With optimal parameters selected, the models split the dataset with 30% reserved for testing. After the training of the models, a full report on each models' prediction accuracy is created. The reports include confusion matrix metrics of precision, recall, fi-score, and support. The report also prints a graph of the ROC curve to show the relationship between the True Positive

Rate and the False Positive Rate. To test for overfitting, the models also conduct cross validation tests and print the scores of the different folds. An example of a model's report is shown below:

```
Random Forest Classifier with 2 estimators
Model Score:  0.8888888888888888
Classification Report:
              precision    recall  f1-score   support

           1       0.85      1.00      0.92        29
           2       1.00      0.83      0.91         6
           3       1.00      0.60      0.75         5
           4       1.00      0.75      0.86         4
           5       0.00      0.00      0.00         1

    accuracy                           0.89        45
   macro avg       0.77      0.64      0.69        45
weighted avg       0.88      0.89      0.87        45

ROC Curve:


Cross Validation Scores:
3 folds:  [0.47058824 0.66666667 0.69230769]
5 folds:  [0.36363636 0.77777778 0.88888889 0.77777778 0.71428571]
Hamming Loss:  0.1111111111111111
```

Dimensionality reduction via Principle Component Analysis is also carried out. For each of the datasets, the 2 features with the strongest correlation to the labels are determined. All datapoints are then graphed based on these 2 features.

Analysis

There were 6 different single label datasets and 3 different multilabel datasets that the classification models were run on. Each dataset had varying ranges of values and this caused models to perform better on some datasets than others. For the single label datasets, the model score was the primary factor for model grading. Here are the results for each dataset:

Dataset	Best Model	Score
Dataset 1	SVM	.97777
Dataset 2	Random Forest	.5
Dataset 3	KNN	.95555
Dataset 4	Logistic Regression	.99346
Dataset 5	SVM	.70537
Dataset 6	SVM	.36957

For the multilabel datasets, we applied a One-Vs-Rest strategy when using the same classifier models. The Hamming Loss was chosen to be the principle factor for assessment of the models for the multilabel datasets. Here are the results:

Dataset	Best Model	Score
Dataset 1	SVM	.94667
Dataset 2	Logistic Regression	.75253
Dataset 3	KNN	.89648

Future Sales

For the task of predicting the next month's sales of various stores, we had to take a completely different approach though. We wanted a model that would take all previous month's sales into account and would give more weight to recent months so implementing a Recursive Neural Network was the logical choice. We decided to use a Long Short Term Memory RNN specifically so that older month's sales were not forgotten. We designed the network with the following layers:

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 12, 10)            480
_________________________________________________________________
lstm_2 (LSTM)                (None, 12, 6)             408
_________________________________________________________________
lstm_3 (LSTM)                (None, 1)                 32
_________________________________________________________________
dense_1 (Dense)              (None, 10)                20
_________________________________________________________________
dense_2 (Dense)              (None, 10)                110
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 11
```

After a bit of preprocessing, we fed the previous months data through the LSTM network and produced satisfactory results.