Final Project - CSC 4850

Yash Shah
yshah4@student.gsu.edu

1. Multi-Class Classification:
For the first problem, the first thing that was done was to replace the missing values with the mean of the values. The easiest way to do this was by implementing the following code :

```
# fill missing values
if X.isnull().any().any():
    # change to nan
    X = X[X < 1e99]
    # fill linear-ly
    X = X.interpolate()
    # fill outside values with mean
    X = X.fillna(X.mean())
```

The next step was to use classifiers to see which one has the best accuracy. I chose to use three classifiers. The three classifiers I use are k-nearest neighbors, support vector machine, and linear regression.

All the classifiers split the data into 90% train and 10% testing. I also used a random_state = 20 so the test can be replicated again. For KNN each dataset ran through with 5-15 neighbors and it chose the best model.

- The first dataset the best classifier was tied with KNN (n_neighbors = 5) and SVM the accuracy was 0.8666666666666667.
- The second dataset the best clssifer was SVM and the accuracy was 0.8.
- The third dataset the best clssifer was knn('n_neighbors': 14) and the accuracy was 0.33174603174603173
- For the fourth dataset the best classifier was linreg: 0.7795576185903385
- For the fifth dataset the best classifier linreg: 0.4847238516155669
- For the sixth dataset the best classifier was linreg: 0.4936204763073814

Classes for the 6th dataset
Classes: [925000. 2250000. 8000000. 3500000. 1750000. 1500000. 950000.
842500. 1250000. 800000. 600000. 1000000. 680000. 590000.
650000. 5000000. 5250000. 1600000. 717500. 900000. 3750000.
792500. 6000000. 5500000. 724500. 725000. 4000000. 4250000.
742500. 750000. 4500000. 3000000. 630000. 2200000. 1300000.
700000. 1200000. 2500000. 1050000. 4750000. 767500. 3100000.

2350000. 6750000. 2600000. 667500. 2100000. 7000000. 3900000.
625000. 2750000. 3400000. 575000. 5750000. 715000. 3250000.
825000. 3575000. 5850000. 660000. 13800000. 727500. 7450000.
5400000. 735000. 640000. 892500. 7250000. 10900000. 832500.
692500. 705000. 1850000. 740000. 9000000. 2950000. 875000.
675000. 1100000. 3850000. 635000. 3650000. 6075000. 11000000.
775000. 1700000. 1025000. 7875000. 2000000. 2570000. 10000000.
2900000. 840000. 2075000. 5800000. 1150000. 2800000. 4350000.
3275000. 3667000. 1650000. 620000. 3800000. 4050000. 4400000.
2400000. 3550000. 2700000. 2825000. 3600000. 874125. 1800000.
4100000. 645000. 817500. 7500000. 5600000. 3200000. 4875000.
2300000. 632500. 4300000. 8750000. 1125000. 595000. 4838000.
5200000. 867500. 850000. 722500. 615000. 6500000. 12000000.
2725000. 670000. 1900000. 3700000. 9500000.]

2.

Setup

The first thing to approximate accuracy of the values of the missing values, 100 known values from each of the dataset were removed. This is on top of the missing values that were already given. All these values are set to NaN.

We ran each dataset through two interpolation strategies, linear and slinear, as well as two imputation strategies, mean imputation and knn imputation. The weighted knn imputation method was coded using numpy and pandas, to find 4 nearest neighbors using euclidean distance, and weight their value contributions accordingly.

The results of each method were compared to the original dataset, with the initial unknown values replaced by predicted values to reduce error, and compared using sklearn's mean squared error method.

Mean Squared Error for MissingData1.txt:

RUN	LINEAR INTERPOLATION	SLINEAR INTERPOLATION	MEAN IMPUTATION	KNN IMPUTATION
1	0.002445	0.002526	0.004307	0.003237
2	0.004930	0.005700	0.005264	0.005264
3	0.003634	0.003089	0.002595	0.002595
4	0.002804	0.002731	0.003249	0.002348
5	0.003209	0.003947	0.003594	0.002534
AVG	0.0034044	0.0035986	0.0038018	0.0031956

Final Project - CSC 4850

Mean Squared Error for MissingData2.txt:

RUN	LINEAR INTERPOLATION	SLINEAR INTERPOLATION	MEAN IMPUTATION	KNN IMPUTATION
1	0.000780	0.000829	0.000709	0.000362
2	0.000857	0.000468	0.000561	0.000441
3	0.000961	0.000658	0.000528	0.000412
AVG	0.000866	0.000651	0.000599	0.000405

For the first dataset weighted knn imputation method performed slightly better than other methods, but in dataset 2 clearly separated itself from the other methods. This might be because the second dataset is substantially larger, which is better for knn. Thus the weighted knn classifier was utilized for dataset three.

4. Time Series Classification & Prediction Deep Learning

I tried setting up an SVM to predict the sales for November 25th.