CSC 466 Project Presentation

Frank Assumma and Zach Weinfeld

Project Pitch

We will use a dataset containing health data from thousands of subjects used to predict a patient's risk of a stroke. Our primary goal will be to create a classification model that predicts if a subject will have a stroke. We will use decision trees and neural networks as our two KDD methods. We will implement each of these methods by hand and compare accuracy and efficacy of each implementation to SciKit Learn's corresponding implementation.

Data

Target Variable

	gender	age	hypertension	heart_disease	ever_married	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Urban	228.69	36.6	formerly smoked	1
1	Male	80.0	0	1	Yes	Rural	105.92	32.5	never smoked	1
2	Female	49.0	0	0	Yes	Urban	171.23	34.4	smokes	1
3	Female	79.0	1	0	Yes	Rural	174.12	24.0	never smoked	1
4	Male	81.0	0	0	Yes	Urban	186.21	29.0	formerly smoked	1
5	Male	74.0	1	1	Yes	Rural	70.09	27.4	never smoked	1
6	Female	69.0	0	0	No	Urban	94.39	22.8	never smoked	1
7	Female	81.0	1	0	Yes	Rural	80.43	29.7	never smoked	1
8	Female	61.0	0	1	Yes	Rural	120.46	36.8	smokes	1
9	Female	54.0	0	0	Yes	Urban	104.51	27.3	smokes	1

3426 rows × 10 columns

Source: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Methods

- We implemented a single-layer neural network and a decision tree in order to create our predictions and analyze the data.
    - In our neural network implementation, a constant learning rate and logistic sigmoid activation function were used.
- Our SciKit Learn neural network involved 100 hidden layers with an inverse scaling learning rate and logistic sigmoid activation function.
- Both our SciKit Learn decision tree and our implemented decision tree were numeric decision trees with entropy as the criteria, a test size of 20%, a training size of 80%, and a minimum split count of 5.
    - Gain and gain ratios were used to generate the rules of the trees and produce predictions

Decision Tree

Our Results:

	0	1
0	635	7
1	42	2

	precision	recall	f1_score	support
0	0.937962	0.989097	0.962851	642
1	0.222222	0.045455	0.075472	44

SciKit Learn Results:

	0	1
0	610	32
1	35	9

	precision	recall	f1_score	support
0	0.945736	0.950156	0.947941	642
1	0.219512	0.204545	0.211765	44

Note: SciKit Learn's tree predicted far more strokes than our model.

This resulted in decreased F1 score for stroke presence despite similar prediction proportions between the two

Neural Network

Our Results:

	0	1
0	499	155
1	30	2

	precision	recall	f1_score	support
0	0.943289	0.762997	0.843618	654
1	0.012739	0.062500	0.021164	32

SciKit Learn Results:

	0	1
0	631	23
1	30	2

	precision	recall	f1_score	support
0	0.954614	0.964832	0.959696	654
1	0.080000	0.062500	0.070175	32

Note: SciKit Learn's neural network predicted far fewer strokes than our model.

Our model held a large amount of false positive predictions despite the same amount of false negative predictions and true positive predictions

Specific Aims

1. What are the correlations between the given variables and having a stroke?
2. Which subjects have missing data, and what insights can we get from this?
3. Do any natural clusters form in the data? What do these clusters tell us?

Specific Aim 1: Correlations

- We calculated the correlation between each of the variables with “stroke”
- As expected, age has the highest positive correlation with “stroke”
- While these are weak correlations, we would expect a correlation near zero if the variables truly were independent.

```
age                  0.242495
hypertension         0.143647
avg_glucose_level    0.140453
heart_disease        0.138553
smoking_status       0.022042
bmi                  0.011673
Residence_type      -0.006068
gender              -0.012594
ever_married        -0.071691
Name: stroke, dtype: float64
```

Specific Aim 2: Missing Data

- In the full dataset, about 5% of subjects were classified as having a stroke
- We looked at subjects with missing data for the BMI variable, and found that in this group, about 20% of subjects were classified as having a stroke
- This difference could be due to some systematic inequality or an implicit factor not measured in the dataset

Full Dataset

```
0    0.951272
1    0.048728
Name: stroke, dtype: float64
```

Missing Data Dataset

```
0    0.800995
1    0.199005
Name: stroke, dtype: float64
```

Specific Aim 3: Clustering

- We used the silhouette score to find the optimal value for *k*
  - We calculated *k=2* clusters was optimal
- The data naturally partitioned
- If we were to use this clustering as a classifier, the observations would be classified with an accuracy of 82.5%

```
n=2
kmeans = KMeans(n_clusters=n, n_init=10)
kmeans.fit(X)
X["cluster"] = kmeans.labels_
(X["cluster"] == y).value_counts(normalize=True)
```

```
True     0.824577
False    0.175423
dtype: float64
```

Conclusion

- Overall, our implementations produced similar results with SciKit Learn's implementation, except for significant variations in the amount of true negatives and false positives
  - Both of SciKit Learn's models had far fewer true negatives.
  - SciKit Learn's neural network had far fewer false positives.
- Our models were somewhat slower to train than SciKit Learn's models
- Hopefully, more analysis of this data can pave the way to more accurate models, and ultimately save lives by aiding with stroke prevention across the world.