Project 01 Summary

Tamique de Brito

July 2019

Abstract

A brief description of my first project exploring machine learning and data science. The purpose of this first project was for me to try applying basic machine learning algorithms I implemented (decision tree, KNN, perceptron) to a data set ("Absenteeism at Work"—obtained from UCI Machine Learning Repository) as well as to learn about data manipulation and visualization. The purpose of writing this paper is to give me experience with paper-writing and to communicate relevant experience in this type of work. I first did some pre-processing and visualization of the data (images included here), then I applied the algorithms to the data set to predict one of the attributes (hours absent) from the others. Numpy was used for some of the numerical computations and matplotlib was used for visualizations. No libraries were used for data manipulation.

1 Data preprocessing and visualization

1.1 Preprocessing

The preprocessing consisted mostly of writing and applying functions to do the following:

- Remove labels by name
- Convert data into list of (feature, label) pairs
- Put each attribute value into a category based on which of a specified set of numerical ranges it falls in.

The following features were removed:

- ”ID” (removed for all algorithms, as it would not be expected to predict).
- ”Reason for Absence” (removed for perceptron and KNN, as the distance measure for this is not a good indicator of similarity)
- ”Hit Target” (removed for all—unsure what it meant).

1.2 Visualization

The data visualization consisted mainly of a set of histograms for each attribute. This can be seen in figures 1 and 2 below. These were used to choose the cutoffs for the categorization of attributes as well to just get a feel for the data.

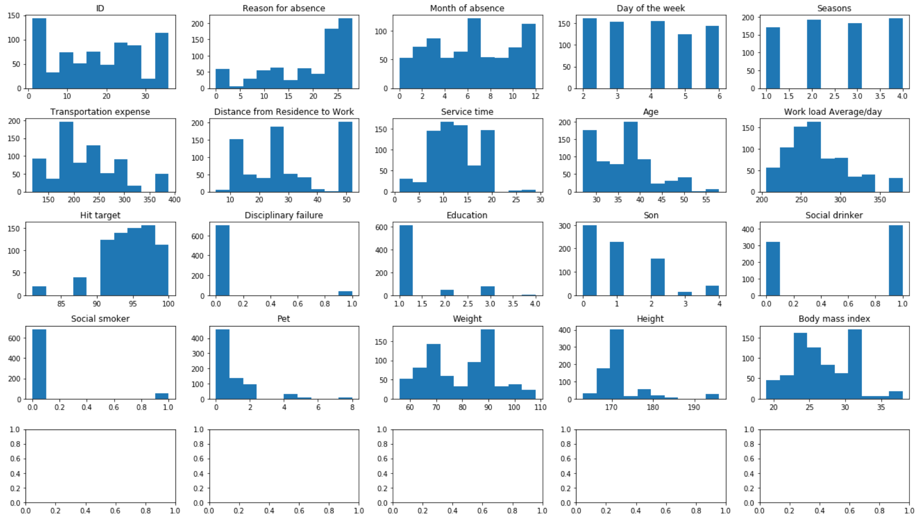

Figure 1: Histogram of attributes

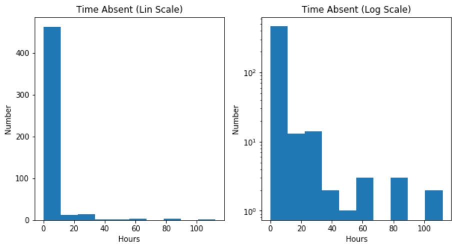

Figure 2: Histograms of time absent

Other visualizations were done, but those were for results of algorithms, and they are shown in the corresponding sections.

2 Perceptron

This is the least interesting, so it's going first. Vanilla perceptron on the numerical feature vectors (no categorization of features, though labels were of course categorized into a binary value). No visualizations/graphs to show. Average error rate of around 45%.

3 Decision Tree

3.1 Basic Decision Tree

To apply this algorithm, cutoffs for each feature were selected (manually) and applied to the data to produce binary features and labels. The error rate of this algorithm when the cutoff was selected to divide the data into two groups of equal size was around 30%. A graph of the error rate for across different cutoff values is shown in figure 3.

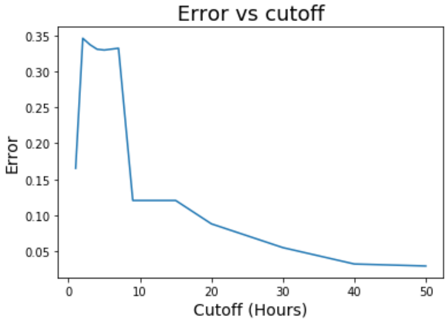

Figure 3:

3.2 Multiclass Decision Tree

This was similar to the binary decision tree, but the labels were categorized into many discrete classes rather than just two. This one actually performed worse than the binary one, at an error rate of 60%, however this is probably because the categories are more finely divided, so it predicts a class that is close to, but not the same, as the correct one. At the least, this model mapped the test data to a distribution pretty close to the actual distribution, as seen in figure 4.

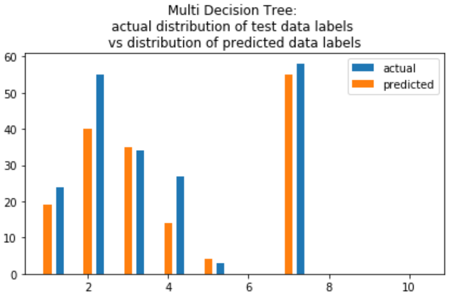

Figure 4:

4 KNN

This algorithm was applied in two ways:

- No categorization of features/labels. Just a numerical average of the K-nearest-neighbors. This got an average absolute-value loss of around 6 hours. (K = 30)
- Categorization as with the the multidecision tree, where the normal label-voting applied. This got an average error of about 60%. (K = 30)

The graphs of the average loss/error vs K are shown in figures 5 and 6.

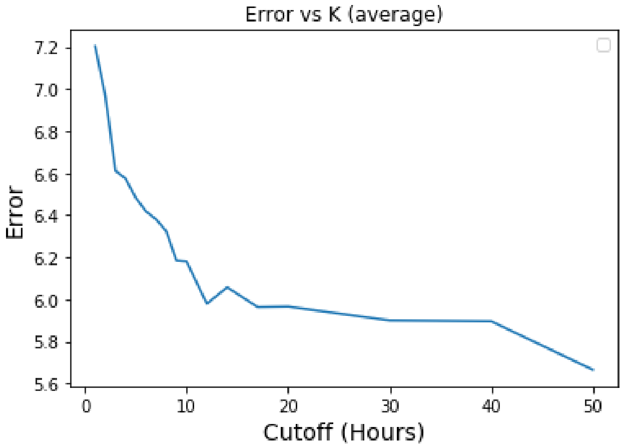

Figure 5:

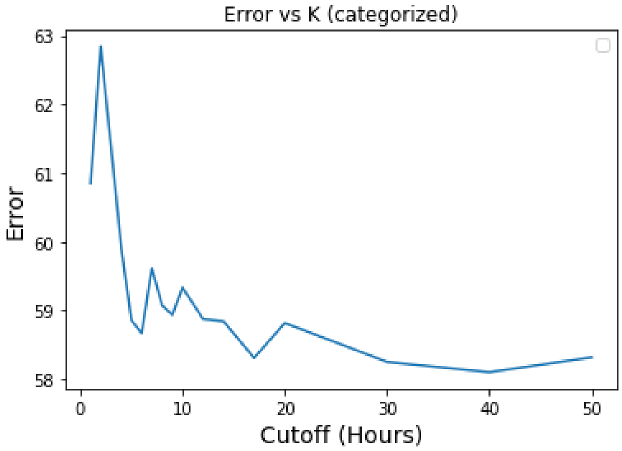

Figure 6:

Additional generated graphs shown below:

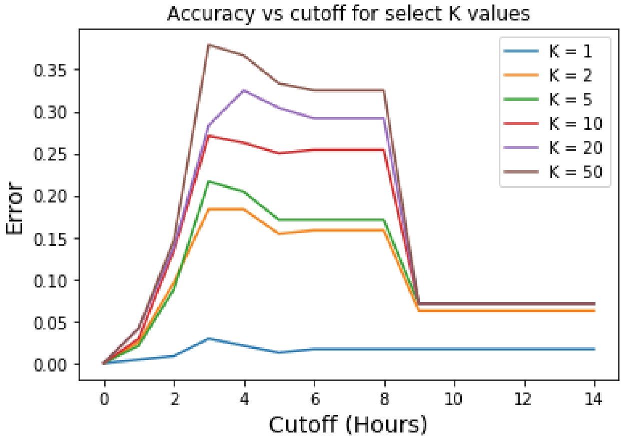

Figure 7:

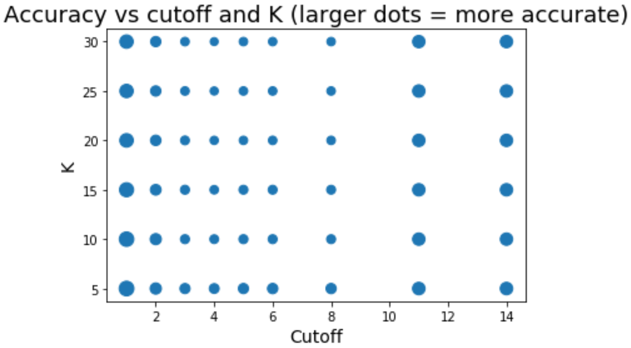

Figure 8:

5 Discussion

The prediction accuracies were not so great. They probably could have been improved by using different algorithms, or transforming the data (or by using an "out-of-the-box" algorithm, but that was not the point of this project). I'm fine with this though, as the point was mainly to get more comfortable with manipulating data and applying algorithms to it, not necessarily to get good prediction results. I intend to do projects in the future where the focus is on accuracy or a business application of information that can be gained from the data.