CS6350 - Machine Learning
Mid-term Project Report

Rishanth Rajendhran
(u1419542)

We shall implement a classifier for a binary classification task.

Data

The dataset used for this project was extracted by by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the specific conditions. Prediction task is to determine whether a person makes over 50K a year given the census information. There are 14 attributes, including continuous, categorical and integer types. Some attributes may have missing values, recorded as question marks.

Background

Data Preprocessing:

1. Data imputation

It is common to have missing attribute values in real-world data. Missing values in the data set need to be handled before training a machine learning model over it.

One way to handle missing values would be to ignore data instances with missing attribute values. This method is not suitable unless we have a very large training data set with mostly data instances with complete attribute values. This technique can potentially lead to a loss of training data which in turn may affect performance of the machine learning model as we might not have enough training instances to train the model.

Another method is to replace missing attribute values with statistical measures such as mean, median and mode of all data instances for the given attribute.

2. Data standardization and normalization

Different attributes may have values in different ranges. When using linear classifiers/neural network models, this can cause some attribute values to hugely influence the outputs simply because those attributes take values larger in magnitude than other attributes even if they are weakly correlated to the target label/value. To prevent this, all attribute values can be brought within the same range. Typically, all attributes are made to have values in the range [0,1] by standardizing the data:

X' = (X-Mean)/StandardDeviation

Other ways of normalization includes:

1. Min-Max normalization: X' = (X-Min)/(Max-Min)
2. Mean normalization: X' = (X-Mean)/(Max-Min)

It is important to note that machine learning models such as decision trees do not require standardization of input data.

3. One-hot encoding categorical attributes

Most machine learning models expect numerical data. However it is common to have categorical attributes in real-world data sets. One way to convert categorical data into numerical data is the method of one-hot encoding. Given a categorical attribute, we infer distinct values the attribute can take from the available data (this can also be available as prior knowledge (e.g.) 'sex' attribute takes values "male" and "female" (world knowledge)). We replace the categorical attribute in the dataset with a set of n binary attributes where n is the number of distinct values that attribute can take. For every data instance, only the column corresponding to the value taken by that data instance for that attribute is set to one, all other columns are set to zero.

One thing to note here is that if we know that a data instance did not take (n-1) of the n distinct attribute values, we can infer that it took the remaining attribute value (known as multicollinearity). Therefore, we can do away with one of the n columns.

4. Feature selection

Choosing the right set of features to train machine learning model over is very important as irrelevant features can confuse the model. One way to find out most relevant features for a given task is by using the correlation matrix. Correlation coefficient is computed for every attribute with respect to the target label/value. Only those attributes with high correlation coefficient are chosen. Additionally, attributes which as closely correlated to each other introduce redundancy. We can choose to keep one among the highly correlated attributes to be given to the ML model.

Algorithms:

The following machine learning models were employed one after another and the respective performances were noted. The input given to these models were preprocessed; some or all the above mentioned preprocessing techniques were employed and the best choice was chosen based on the performance of the validation data.

1. Decision tree classifier

Decision tree classifier is an intuitive machine learning model that can be employed even on non-linearly separable data. One main advantage of using decision trees is its interpretability. Printing out the decision tree can be a very good indicator of why the model is working well on the given data or why it isn't. Decision to make splits can be based on several metrics such as entropy, majority error and Gini Index. It is worth experiment with these metrics along with the maximum allowed depth, to avoid overfitting on the training data, a common occurrence with decision trees.

2. Adaboost

Adaboost can help overcome the problem of overfitting. Typically decision stumps are used as the weak learners in this ensemble learning method. The weak learners are learnt sequentially focussing on those training instances on which the previous weak learners performed very poorly.

3. Bagging

This is yet another ensemble learning that can help prevent overfitting on the training data. Learners are learnt in parallel. A subset of the training data generated by uniformly sampling data instances with replacement from the training data is used for training the weak learners. However, bagging may not work any differently from the individual learners if there are strongly indicative features in the data set.

4. Random Forest

Random forest is another ensemble method and it is very similar in bare-bone structure to Bagging. To circumvent the problem of strongly indicative features, here besides randomly sampling data instances with replacement, we also randomly sample the features without replacement. This way, the individual learners get to work with different set of features and this prevents the strongly indicative features from largely influencing the predictions of the individual learners.

<u>Results</u>

Model \ Evaluation	Train Accuracy	Validation Accuracy
Decision Tree	0.99994286	0.811866
AdaBoost	0.875542857142857	0.872
Bagging	0.999942857142857	0.856
Random Forest	0.999942857142857	0.854533333333333

Going forward:
While decision trees are prone to overfitting, the ensemble learning methods employed should have solved that problem to some extent. The improvement observed was not substantial which could be because of the makeup of the data at hand. It could also be because of wrong choices on how to combine predictions from individual weak learners. Other machine learning models can be employed to get better performance:

1. Nearest Neighbours
The goal here is to predict whether the income level of a person defined by the data attributes has an income over a certain level. One would expect that the people with similar educational backgrounds and age should have similar income levels. Based on this intuition, another machine learning model that can be employed is the nearest neighbors model. This may not work very well if the test data is far away from the train data. Also outliers can pose a problem during training as this model is sensitive to outliers (What if there is a prodigy who earns way more than his contemporaries?)

2. SVM
SVM can be employed to find a separating hyperplane between the training instances. In case the performance is not great with linear SVM, non-linearity can be introduced using different kernel functions. Finding the right kernel function can be non-trivial.

5. Neural Networks
This is another model that is worth exploring. While it might not be easy to interpret the working of the neural network, they are very good at exploiting relationships between various attributes and with the target label. We can also explore a deep neural network but it is not clear if that would be necessary considering that the data at hand has a reasonable small number of features.

Conclusion

Several supervised machine learning models were employed to perform the binary classification task at hand. The performance of these models were evaluated based on the errors these models produced on the validation data. Other supervised machine learning models with non-linearity shall be employed going forward.