

AML PA1

ajchavan

March 2017

Run program like => e.g. - `python2.7 pa1_logistic.py`

1 Data preprocessing

1. Data preprocessing is executed in a method called `preprocess_data()`
2. Appended train and test file together and converted it to `adult.csv`
3. Read csv file using pandas by specifying `na_values=['?']` to consider for missing values and later on drop those columns using `dropna()` method in pandas.
4. From the data, I removed work-class, race and native-country as they didn't form a substantial logic for the income prediction. After reading the data description, I still didn't understand what `fnlwgt` meant, so I looked up online and I don't think it will have any credibility for income prediction, so dropped that too
5. From the data available, I found online how to convert ordinal and nominal variables to convert to boolean and feed it to sklearn estimation. For continuous values too, e.g. if data is in the range, say different numbers from 0 to 1000, then we do the same procedure as ordinal variables i.e. Go through entire dataset, if a value corresponding to the current value is found, then make it true else False. Repeat this for every unique value. Apply this same procedure for each algorithm.
I have explained this in the code too for better understanding.
6. Since we are predicting whether income is greater than \$50K or not, I use `> $50K` as the parameter and drop `<= $50K`

2 Training the data

1. After the procedure for preprocessing data, we train the data for the 4 Machine Learning algorithms
2. In training, I split train/test to 0.8/0.2.
3. Using sklearn, fit the X and y from the training data from preprocessing data, and then predict using Xtest and ytest
4. Using `classification_report` from `sklearn.metrics`, we get precision, recall and f1 score and support.
5. After that, using `predict_proba` for the given estimator to get the yscore and getting roc value using it.

3 Choice of parameter = ROC

I chose ROC as a better performance measure as opposed to F1 score, Precision, Recall because of 2 reasons:

1. Data is not askewed. Every attribute of a given category isn't equally distributed and so accuracy cannot be considered as a reliable measure
2. Our main aim is to maximize our prediction(TPR) or minimize the error. This can be best measured by using ROC curve by finding the ratio of True Positive rate vs False Positive Rate. So even though precision and recall are good measures, but for current case, ROC works the best.

4 Effect of Hyperparameters

4.1 Logistic Regression

For logistic regression, more the number of iterations, better the result.

Below is output for tolerance = 0.1 and 0.0001:

```
ajinkya@ajinkya: ~/Documents/AML_PA1/practise
avg / total      0.86      0.87      0.86      9207

roc: 0.86787082795
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal_logistic.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k      0.83      0.99      0.91      7662
>50k       0.40      0.03      0.05      1545

avg / total      0.76      0.83      0.76      9207

roc: 0.464844029164
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal_logistic.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k      0.92      0.96      0.94      7682
>50k       0.73      0.60      0.66      1525

avg / total      0.89      0.90      0.89      9207

roc: 0.943024272197
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
```

4.2 Decision Trees

In decision trees, when depth is very less then the result isnt accurate, for a good estimated depth, it gives minimum error, while at high depth it again gives bad accuracy.

Below is the output for depth = 2, 5, 100

```
ajinkya@ajinkya: ~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal decisiontrees.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k    0.92     0.93     0.93     7685
>50k     0.64     0.61     0.62     1522

avg / total    0.88     0.88     0.88     9207

roc: 0.810958597264
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal decisiontrees.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k    0.92     0.96     0.94     7661
>50k     0.75     0.61     0.67     1546

avg / total    0.89     0.90     0.90     9207

roc: 0.938879285263
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal decisiontrees.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k    0.84     1.00     0.91     7655
>50k     0.67     0.04     0.08     1552

avg / total    0.81     0.84     0.77     9207

roc: 0.806173572626
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
```

4.3 KNN

Works similar to Decision trees, for smaller k, less accuracy. For very high k, it again drops.

Below is the output for: k=2, 25, 1000

```
ajinkya@ajinkya: ~/Documents/AML_PA1/practise$ python2.7 pal_knn.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
precision    recall  f1-score   support

<=50k       0.88     0.96     0.92     7653
>50k        0.65     0.37     0.48     1554

avg / total       0.84     0.86     0.84     9207

roc: 0.80104024616
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal_knn.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
precision    recall  f1-score   support

<=50k       0.91     0.94     0.92     7644
>50k        0.65     0.54     0.59     1563

avg / total       0.87     0.87     0.87     9207

roc: 0.895034614564
ajinkya@ajinkya:~/Documents/AML_PA1/practise$ python2.7 pal_knn.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
precision    recall  f1-score   support

<=50k       0.86     0.98     0.92     7713
>50k        0.65     0.18     0.29     1494

avg / total       0.83     0.85     0.81     9207

roc: 0.835253282459
ajinkya@ajinkya:~/Documents/AML_PA1/practise$
```

4.4 Naive Bayes

For threshold (binarize) = 0, it gives proper output, as threshold goes close to 1, the accuracy decreases.

Below is output for - binarize = 0, 1

```
ajinkyaa@ajinkyaa: ~/Documents/AML_PA1/practise$ python2.7 pa1_nb.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
      precision    recall  f1-score   support

<=50k      0.95      0.86      0.90      7654
>50k       0.53      0.77      0.63      1553

avg / total      0.88      0.84      0.86      9207

roc: 0.912948605757
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$ python2.7 pa1_nb.py
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of
the model selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed in 0.20.
/usr/local/lib/python2.7/dist-packages/sklearn/metrics/classification.py:1113: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples.
('precision', 'predicted', average, warn_for)
      precision    recall  f1-score   support

<=50k      0.84      1.00      0.91      7704
>50k       0.00      0.00      0.00      1503

avg / total      0.70      0.84      0.76      9207

roc: 0.61036951711
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
ajinkyaa@ajinkyaa:~/Documents/AML_PA1/practise$
```

5 Performance comparison

Since the dataset is sufficiently large enough, Logistic Regression performs better than Naive Bayes.

Logistic regression without binarizing the ordinal features performs almost similar to Decision trees but after applying the feature binarization, Logistic regression performs better than Decision trees but Decision trees still perform better than Naive Bayes, simply because of the simple model of Decision trees even for larger values, and no consideration of generative model requirements like conditional independence.

Since there are many features, Naive Bayes would be preferred over KNN with respect to performance and time taken.

Logistic Regression > Decision Trees > Naive Bayes > KNN = for adult dataset