Project
Cash Class

Predicting Income

University: Sofia University, FMI
Author: Dian Todorov
Programme: Software Engineering
Faculty Number: 61670
Course: Intelligent Systems

Lecturer:
Prof. Ivan Koichev, Ph. D.

Date:.....

Assessed by:.....
(Prof. Ivan Koichev, Ph. D.)

Contents

Cash Class

University project for the Intelligent Systems course @ FMI

Motivation

The aim of this project is to compare different methods for classification on a big dataset with missing and conflicting values.

Introduction to the Task

Prediction task is to determine whether a person makes over 50K a year. Due to noise, conflicts and missing values the dataset is cleaned. Afterwards, different classification algorithms are applied and compared in order to find which one is the most efficient for solving the task at stake.

Short Review of the Problem

The problem of classification has been wiedly explored. In machine learning and statistics, classification is a problem of identification. It is a version of supervised learning where the output values for each instances are supplied. The goal is to find to which of a set of categories a new observation belongs. This is done on the basis of a training set of data containing observations (or instances) whose category membership is known.

My solution

The initial goal was to use Naive Bayes Classifier, Decision Tree Classifier and KNN classifier as simple algortihm and easiest to iterate and tune. Before jumping into implementation there was a need to clean the data

Program Code

The whole project is written in Python 2.7 because of the maturity and the ease of use of the language. To set up the program

```
virtualenv virtualenv
source env/bin/activarate
pip install -r requirements
fab knn # show results from knn classification
fab bayes # show results from bayes valssification
```

Data

The data has the follwoing distribution:

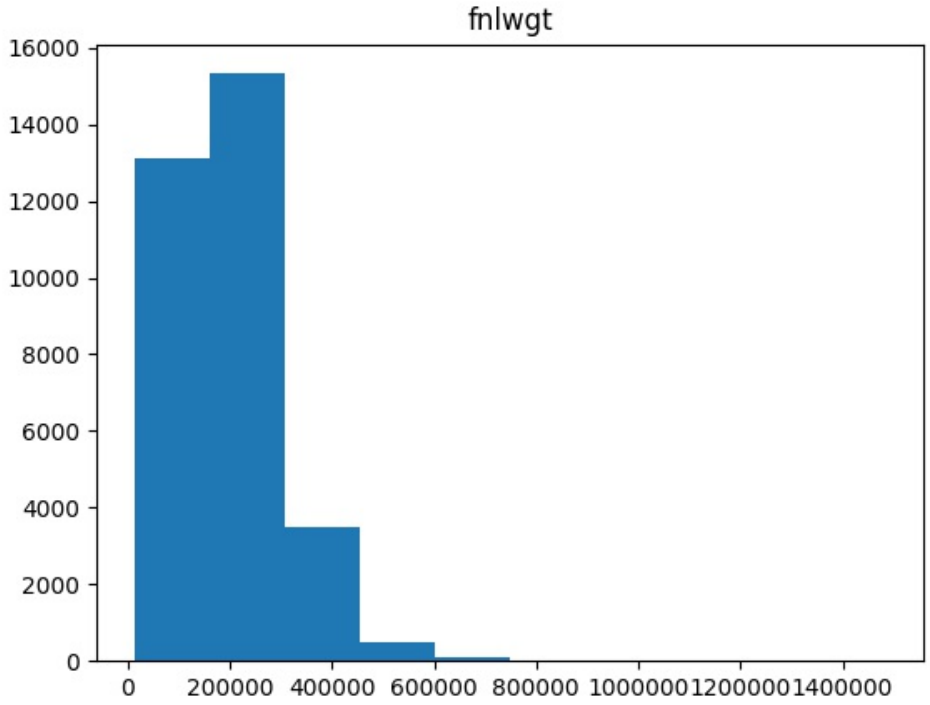
Probability for the label '>50K' : 23.93% / 24.78% (without unknowns)
Probability for the label '<=50K' : 76.07% / 75.22% (without unknowns)

1. Missing Values

All instances in the dataset are 32561. There are some with missing values. After their removal there are 30162 left.

2. Trimming Outliers

After plotting histograms for the different continious attributes it is obvious that there are some instances which are outside any sane boundries. The applied strategies for accomadating for these outliers is to trim any



After the trimming there are left 30136 to work with.

3. Discretization of Continious values

The following attributes are discretized and split into buckets when doing the Naive Bayes Method:

- capital gain:
- capital loss
- capital gain
- age
- flnwgt

4. Nomalization

The following attributes are discretized and split into buckets when doing the Naive Bayes Method:

- capital gain:
- capital loss
- capital gain

- age
- flnwgt

Validation of the program and subsets

It is a standard practice to split the dataset int three sets:

- validation 25%
- training 25%
- testing 50%

After splitting the dataset into several i

Algorithms and Results

Naive Bayes

- Intro to the algorithm:

  In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. As a first attempt Naive bayes is used as a really simple method for classifications.

- Details for the implementation:

  It was necesssary to discretize the data in order to make it possible to assess the probabilities for the different buckets,

- First Iteration

  On the first attemp got results of 82%

- Tuning:

  After examing the validation set and experimenting with increasing the number of buckets for the validation set, the results improved.

- Second Iteration

  On the second attempt got increase to 85%

KNN

- Intro to the algorithm

  k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. It was highly ineffective due to the big complexity in time.

- Details for the implementation:

  Used only the following subset of attributes to make a prediction:

  - 'capital-gain'
  - 'capital-loss',
  - 'hours-per-week',
  - 'fnlwgt',
  - 'age'

  Also did some normalization

- First attemp

  On the first attemp got results of 70%. Sadly this is below the proportion of above vs below instances.

- Tuning

  Play around with K

- Second attempt Reach 76% for with a K=6

Ideas for future development

Naive Bayes improvements

Try using different bucket size to pin point the most useful.

KNN improvements

Exp[eriment with different Ks]

Implementation of other algorithms

Seems appropriate to try out the following Decision tree algorithms

- ID3
- C4.5
- C5.0

ALso experiment try

- neural Netoworks

Ensemble learning

The final big idea is to Build an ensemble of the the methods and experiment with it.

Literarture

- Artificial Intelligence A Modern Approach, Stuart J. Russell and Peter Norvig, 1995
- [Classifiers in python] http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparis
- [K nearesr neighbour] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- [K nearesr neighbour] scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsCl
- [Decision Tree Classifiers] http://scikit-learn.org/stable/modules/tree.html#tree

- [C4.5] https://en.wikipedia.org/wiki/C4.5_algorithm
- [ID3] https://en.wikipedia.org/wiki/ID3_algorithm