Computer Science

Coms3007

Machine Learning 2019 assignment

Students: 1073978, 1741606, 1619542, 1325952, 1653526

## **STEP 1: DESCRIPTION OF THE DATASET**

Reference of the dataset is UCI Machine Learning Repository Dataset Name: Adult Data Source: <https://archive.ics.uci.edu/ml/datasets/Adult>

**- What are the attributes? What values do they take on?**

* **age**: continuous.
* **workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
* **fnlwgt**: continuous.
* **education**: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
* **education-num**: continuous.
* **marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
* **occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
* **relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
* **race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
* **sex**: Female, Male.
* **capital-gain**: continuous.
* **capital-loss**: continuous.
* **hours-per-week**: continuous.
* **native-country**: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

- **What are the targets?**

* The targets are the **income classes** (>50K, <=50K).
* We are predicting whether a citizen earns more than fifty thousand dollars per year given their census information.

- **How many data points do you have?**

* The whole dataset contains **48842** data points

- **Sample data points from the dataset:**

|  |  |
| --- | --- |
| 1 | State-gov,77516, Bachelors,13, Never-married, Adm-clerical, Not-in-family, White, Male,2174,0,40, United-States, <=50K |
| 2 | Private,159449, Bachelors,13, Married-civ-spouse, Exec-managerial, Husband, White, Male,5178,0,40, United-States, >50K |
| 3 | Private,45781, Masters,14, Never-married, Prof-specialty, Not-in-family, White, Female,14084,0,50, United-States, >50K |
| 4 | Self-emp-not-inc,209642, HS-grad,9, Married-civ-spouse, Exec-managerial, Husband, White, Male,0,0,45, United-States, >50K |
| 5 | Self-emp-not-inc,83311, Bachelors,13, Married-civ-spouse, Exec-managerial, Husband, White, Male,0,0,13, United-States, <=50K |

**- State what you are trying to predict with the data?**

* Prediction task is to determine whether a person makes over 50K a year or not.

## **STEP 2: DATA PROCESSING AND VISUALISATION.**

- **How was the data pre-processed**?

* As seen our dataset did not have the right headers for each column(feature) so we set the right name to the columns
* Identified missing values in our data .In our dataset the missing values are represented by a ?. We replaced "?" with NaN (Not a Number), which is Python's default missing value marker, for reasons of computational speed and convenience.
* Since our dataset is mainly categorical, we replaced our missing values in the column using the most frequent class.
* The list goes on. Please see attached **preprocessingNotebook.ipynb** which contains all the explanations and insights to the way data was processed.

**- How were the inputs/targets normalised?**

**- How was the data split into training/validation/test data?**

* The full set of the data was spilt into 2/3 for **training** and **validation** and 1/3 for **test** data. Each Algorithm furthermore splits the 2/3rds of data according to their preference, which we’ll get to later on.

## **STEP 3: ALGORITHM IMPLEMENTATION**

* For this dataset, we implement 3 types of algorithms namely Naïve Bayes, Decision Trees, And Logistic regression.

**Naïve Bayes**

* + There are 2 different implementations of naïve Bayes Classifier.

1. See **naiveBayesClassifier.ipynb**

- **Details of each implementation**

This is the "Naive" implementation of the **naive Bayes algorithm**, with all attributes made **categorical** by using python binning into equal classes. The numerics of the implementation are calculated uses the gaussian method. It uses Series and dictionaries data structures to perform operations

(**Please see the .ipynp file for full implementation explanations , selection of hyperparameters and optimisation of the algorithm)**

- **Error on the test**

The maximum accuracy reached on test was **76.6%,** therefore error is about **23%**

This version of Naïve Bayes was Could be used to improve accuracy, hence we took on a second implementation where \*\*\*\*\*\*(Alec continues)

\*

\*

\*

**confusion matrix**

|  |  |  |
| --- | --- | --- |
|  | **Predicted: >50K** | **Predicted: <=50K** |
| **Actual: >50K** | **TN=** | **FP=** |
| **Actual:**  **<=50K** | **FN=** | **TP=** |

**Decision Trees** ?

- Details of each implementation

- Error on the test

- How and why hyper-parameters were selected

- Presentation of errors in the form of confusion matrix (for e.g.)

**Logistic Regression** ?

- Details of each implementation

- Error on the test

- How and why hyper-parameters were selected

- Presentation of errors in the form of confusion matrix (for e.g.)

## **STEP 4:** A BRIEF DISCUSSION AND ANALYSIS OF RESULTS FROM THE VARIOUS ALGORITHMS.

Decision Tree(Parameter Tuning in the notebook)

* dropped most numerical value because during our dataprocesssing we find that the data points in continous form where hard to interpret for example ‘capital\_gain’ and capital\_loss and in the main algorithm during traing if l include the capital\_gain and capiital\_loss the accurancy on the test set dropped from 75 percent to 24 percent signifying the reduntancy of them in our tree.