Abstract

Classification is a very interesting problem in supervised learning field. This report explains about Rule based classification and the use of Decison Trees to generate rules for classification of the dataset. Applying Decison trees to an important domain like public health care and mortality statistics helps in improving the public health and monitoring by the government. The classification proved that the cause of mortality is actually dependant on various aspects like a person's age, education, gender, race and marital status. Thus, by harnessing the power of data mining, this analysis succesfully predicts the cause of death of an individual with a sound accuracy of around 92%.

1 Introduction

Rule based classification is a supervised learning technique which generates rules to classify the tuples in a dataset. In this study we used the mortality data of United States during 2005 to 2015 provided by "Centers for Disease Control and Prevention" (CDC). We generated the classification rules to predict the cause of death using certain attributes of the individual.

Mortality is particularly an important domain to analyze due its applications in health and human development sectors. Evaluating an individual's personal details helps in order to predict his current condition. Data mining techniques like Decison trees have been implemented to assess the outcome of an individual.

2 Dataset

2.1 Description :

Every year the CDC releases the country's most detailed report on death in the United States under the National Vital Statistics Systems. This mortality dataset is a record of every death in the country for 2005 through 2015, including detailed information about causes of death and the demographic background of the deceased.

It's been said that "statistics are human beings with the tears wiped off". This is especially true with this dataset. Each death record represents somebody's loved one, often connected with a lifetime of memories and sometimes tragically too short.

Putting the sensitive nature of the topic aside, analyzing mortality data is essential to understanding the complex circumstances of death across the country. The US Government uses this data to determine life expectancy and understand how death in the U.S. differs from the rest of the world.

2.2 Overview :

This dataset is a collection of CSV files each containing one year's worth of data and paired JSON files containing the code mappings, plus an ICD 10 code set. The CSVs were reformatted from their original fixed-width file formats using information extracted from the CDC's PDF manuals.

2.3 Attributes :

The chosen dataset has lot of categorical attributes. It comprises of 34 attributes describing each individual deceased. The attributes are as shown below :

1. resident status
2. education as per 1989 revision
3. education as per 2003 revision
4. education reporting flag
5. month of death
6. sex
7. detail age type(flag)
8. detail age
9. age substitution flag
10. age recode 52
11. age recode 27
12. age recode 12
13. infant age recode 22
14. place of death and decedents status
15. marital status
16. day and week of death
17. current data year
18. injury at work
19. manner of death
20. method of disposition
21. autopsy
22. activity code
23. place of injury
24. 358 cause recode
25. 113 cause recode
26. 130 infant cause recode
27. 39 cause recode

28. race
29. bridged race flag
30. race imputation flag
31. race recode 3
32. race recode 5
33. hispanic origin
34. hispanic originrace recode

For our analysis, we chose 7 attributes which serve best for our purpose. They are :

1. **manner of death :** This represents the cause of death of the individual.
2. **education as per 1989 revision :** This represents the education details of the individual according to the 1989 revision.
3. **detail age :** The age of the individual at the time of death.
4. **place of death and decedents status :** The place at which the individual has expired.(either at work, home or hospital)
5. **month of death :** The month of death of the individual.
6. **race :** The origin of the individual.
7. **sex :** This represents the gender of the individual.
8. **marital status :** The marital status of the individual.

2.4 Size of the dataset :

The dataset is of 10 years from 2005 to 2015 with a size of 4.1 GB and each year we have the records of 1015771 individuals. For all the analyses, we considered the data of the year 2005.

2.5 Inferences :

In this experiment, we trained a Desison tree classifier to predict the manner of death(7 classes) based on their education, age, race, gender, marital status, month of death and palce of death.

The manner of death is a categorical attribute comprising of seven levels :

1. Accident
2. Suicide
3. Homicide
4. Pending investigation
5. Could not determine
6. Self-Inflicted
7. Natural

By studying the above mentioned attributes of an individual, we try to predict the manner of death.

3 Background

The term rule-based classification can be used to refer to any classification scheme that make use of IF-THEN rules for class prediction.Rule-based classification schemes typically consist of the following components:

1. **Rule Induction Algorithm :** This refers to the process of extracting relevant IF-THEN rules from the data which can be done directly using sequential covering algorithms or indirectly from other data mining methods like decision tree building or association rule mining.

2. **Rule Ranking Measures :** This refers to some values that are used to measure the usefulness of a rule in providing accurate prediction. Rule ranking measures are often used in the rule induction algorithm to prune off unnecessary rules and improve efficiency. They are also used in the class prediction algorithm to give a ranking to the rules which will be then be utilized to predict the class of new cases.

3. **Class Prediction Algorithm :** We use specific set of algorithms to classify the values using the generated rules.

3.1 The Classifier :

A rule-based classifier uses a set of IF-THEN rules for classification.An IF-THEN rule is an expression of the form:

***IF** condition **THEN** conclusion.*

where ,

- Condition(or LHS) is rule antecedent or precondition.
- Conclusion(or RHS) is rule consequent.

3.2 Assesment of Rule :

1. **Coverage of a rule :** The percentage of instances that satisfy the antecedent of a rule (i.e., whose attribute values hold true for the rules antecedent).

$$coverrage(Rule) = \frac{n_{covers}}{|D|}$$

2. **Accuracy of a rule :** The percentage of instances that satisfy both the a ntecedent and consequent of a rule.

$$accuracy(Rule) = \frac{n_{correct}}{n_{covers}}$$

where,

- D :Class labelled dataset.
- $|D|$:number of instances in the dataset.
- n_{covers} : The number of instances a rule covers.
- $ncorrect$: The number of instances correctly classified by the rule.

3.3 Executing a rule set:

Once generated, there are two ways to execute a rule set.

- Ordered set of rules : Order is important for interpretation.
- Unordered set of rules : Rules may overlap and lead to different conclusions for the same instance.

Let X be an instance of the dataset that satifies the rule R, then the rule is said to be triggered. The potential problems could be :

- If more than one rule is satisfied by X : *This is solved using the ”conflict resolution strategies discussed below.*
- If no rule is satisfied by X : *This is solved using the ”Default class”*.

3.3.1 Conflict resolution strategy:

These strategies are employed if an instance triggers more than one rule in the derived rule set. These are of two types :

1. **Size Ordering :**
   - Assign the highest priority to the triggering rules that is measured by the rule precondition size.(i.e., the most attribute test)
   - The rules are unordered in this strategy.
2. **Rule Ordering:**
   (a) *Class-based ordering :* This is the most popular strategy used for conflict resolution. It orders the rules in the decreasing order of frequnecy i.e., all of the rules for the most frequent class come first, the rules for the next most frequent class come next, and so on. This results in decreasing order of ”misclassification per class”.

   (b) *Rule-based Ordering :* In this technique, the rules are ordered into one long priority list based on measures such as accuracy, coverage or some other measure chosen by the experts.

3.3.2 The Default class :

If an instance X is not covered by any rule, we follow the below steps to assign a default rule for that instance.

- A default rule can be set up to specify a default class,based on the training set.
- This may be the class in majority or the majority class of the instances that were not covered by any rule.
- The default rule is evaluated at the end, if and only if no other rule covers the instance X.
- The condition in the default rule is empty.
- In this way, the rule fires when no other rule is satisfied.

3.4 Building the classification rules :

We start building our classification rules directly or indirectly.

- **Direcct methods :** In these methods, we extract the rules directly from the dataset. The common algorithms are ”1R” algorithm and ”sequential covering algorithms” like PRISM, RIPPER, CN2, FOIL, and AQ.
- **Indirect Methods :** In these methods, we extract the classification rules from other classification models like Decision trees.

3.4.1 Sequential covering algorithms :

Sequential algorithms are the most widely used aprroach to generate the classification rules. They learn the classification rules sequnetially to generate the rule set. A typical sequnetial algorithms uses below procedure :

1. Rules are learned one at a time.
2. Each time a rule is learned, the instances covered by the rules are removed.

3. The process repeats on the remaining instances unless there are no more training examples or the quality of a rule returned is below the user specified level.

General sequential algorithms include PRISM, FOIL, CN2, AQ, RIPPER. When compared to a Decision tree induction, training a sequnetial algorithm is slower since, the decison tree model learns the rules simultaneously.

3.4.2 Rule extraction from a Decision tree :

One of the indirect methods is using a Decision tree classifier model to generate the rule set. There are few points to be noted in using a decison tree :

- Decision trees can become large and difficult to interpret.

    1. Rules are easier to understand than large trees.
    2. One rule is created for each path from the root to a leaf.
    3. Each attribute-value pair along a path forms a precondition and the leaf holds the class labels.
    4. The order of the rules does not matter.

- Rules are :

    1. **Mutually exclusive :** No two rules will be satisfied for the same instance.
    2. **Exhaustive :** There is one rule for each possible attribute-value combination.

3.5 Pruning the rule set :

The resulting set of rules generated by a Decison tree can be quite large and difficult to follow. Also, there could be inefficient(not sufficient accuracy can be achieved by the rule) rules generated. In order to avoid this, we prune the rule set before interpreting them.For a given rule any condition that does not improve the estimated accuracy of the rule can be pruned (i.e., removed).

C4.5 algorithm extracts rules from an unpruned tree, and then prunes the rules using an approach similar to its tree pruning method.

3.6 Dealing with numeric attributes :

When building a decision tree, we discretize the numeric attributes. We divide each attribute's range into intervals and follow the below steps:

1. Sort instances according to attributes values.

2. Place breakpoints where class changes (majority class).

3. This minimizes the total error.

3.7 The problem of Overfitting :

Whenever we train a statistical model, we should make sure that the model does not fit too accurately for the training data. Because, this results in inaccurate prediction when a new instance is tested.

Overfitting is very likely whenever an attribute has very large possible values. This procedure is very sensitive to noise since, even one incorrect class label can lead to a seperate interval. It results in attributes having zero errors. The simplest solution to avoid overfitting is to enforce minnimum number of instances in the majority class per interval.

4 Algorithm (Decision tree learning with C4.5)

4.1 Overview :

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data and the resulting classification tree can be an input for decision making.

4.2 Metrics :

In order to build a statistical model, we use some metrics to evaluate and improve our model. In case of Decison trees, we have two metrics:

1. **Gini impurity :** Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Gini impurity can be computed by summing the probability p_i of an item with label i being chosen times the probability $\sum_{k\neq i} p_k = 1 - p_i \sum_{k\neq i} p_k = 1 - p_i$ of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

2. **Information gain :** Information gain is based on the concept of entropy from information theory.

   Entropy is defined as below
   $H(T) = I_E(p_1, p_2, ..., p_J) = -\sum_{i=1}^{J} p_i \log_2^{p_i}$
   where $p_1, p_2, ...p_1, p_2, ...$are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.

   $$\overbrace{IG(T,a)}^{InformationGain} = \overbrace{H(T)}^{Entropy(parent)} - \overbrace{H(T|a)}^{WeightedSumofEntropy(Children)}$$

   Information gain is used to decide which feature to split on at each step in building the tree. Simplicity is best, so we want to keep our tree small. To do so, at each step we should choose the split that results in the purest daughter nodes. A commonly used measure of purity is called information which is measured in bits, not to be confused with the unit of computer memory. For each node of the tree, the information value ”represents the expected amount of information that would be needed to specify whether a new instance should be classified yes or no, given that the example reached that node”.

4.3 C4.5 algorithm :

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. Authors of the Weka machine learning software described the C4.5 algorithm as "a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date".

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, ...$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where the x_j represent attribute values or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sublists.

4.4 Advantages of Decision trees :

1. Simple to understand and interpret.
2. Able to handle both numerical and categorical data.
3. Requires little data preparation.
4. Uses a white box model.
5. Possible to validate a model using statistical tests.
6. Performs well with large datasets.
7. Mirrors human decision making more closely than other approaches.
8. Robust against co-linearity, particularly boosting.
9. In built feature selection. Additional irrelevant feature will be less used so that they can be removed on subsequent runs.

4.5 Limitations :

1. Trees do not tend to be as accurate as other approaches.

2. Trees can be very non-robust. A small change in the training data can result in a big change in the tree, and thus a big change in final predictions.

3. The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts.

4. Decision-tree learners can create over-complex trees that do not generalize well from the training data. (This is known as overfitting) Mechanisms such as pruning are necessary to avoid this problem.

5. There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems. In such cases, the decision tree becomes prohibitively large.

6. For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of those attributes with more levels.

5 Experiment and Results :

5.1 Goal and metrics:

For the major part of the analysis, we used the data corresponding to 2005. The rest of the data is similar to this and can be used in trend and growth analytics. It is particularly interesting to study the manner of death since in many a cases, it helps in investigations and help in better catering of health care services.We used Decision trees to generate our rule set and classified our dataset into seven classes, each describing the manner of death of an individual.

Python library **sklearn** provides many data mining algorithms out of the box which makes our analysis easier and efficient. The **tree** module of *sklearn* provides the implementation for a ”DecisionTreeClassifier” model. **Pandas** library is used to work with the data. The missing and invalid values have been handled using it. The metrics of the trained model are as described below :

1. **Accuracy :** This metric represents the performance of our trained model on unseen test data.
2. **Loss :** It represents the amount of error the model experiences in fitting the data.

We choose a model which has the best accuracy and the least loss. In our experiment, we used **Root mean square loss**(RMS) as the evaluating loss function. RMS is the square root of sum of differences between the predicted class and the actual class. It is the most widely used technique to evaluate a model. We tuned the model by changing the minimum samples per leaf from 1 to 50 to choose the optimal smaple population.

The accuracy and loss values are as shown in *figure.* 1 , *figure.* 2 below:

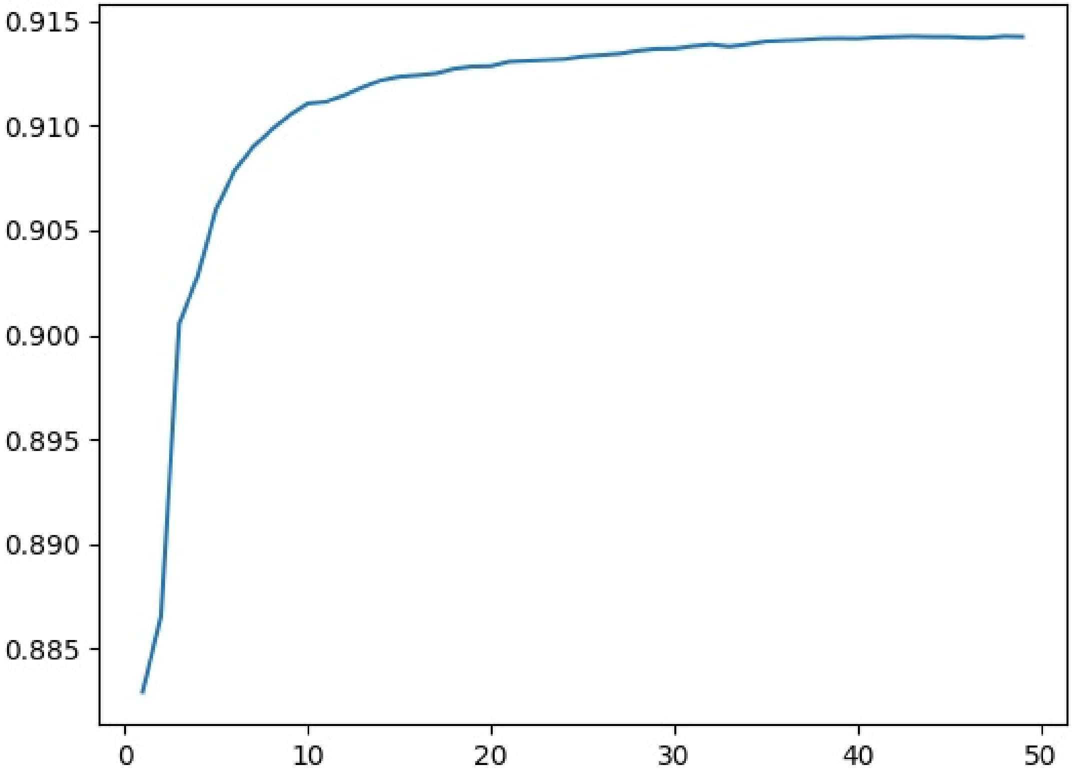


Figure 1: Accuracy vs Minimum samples per leaf node.

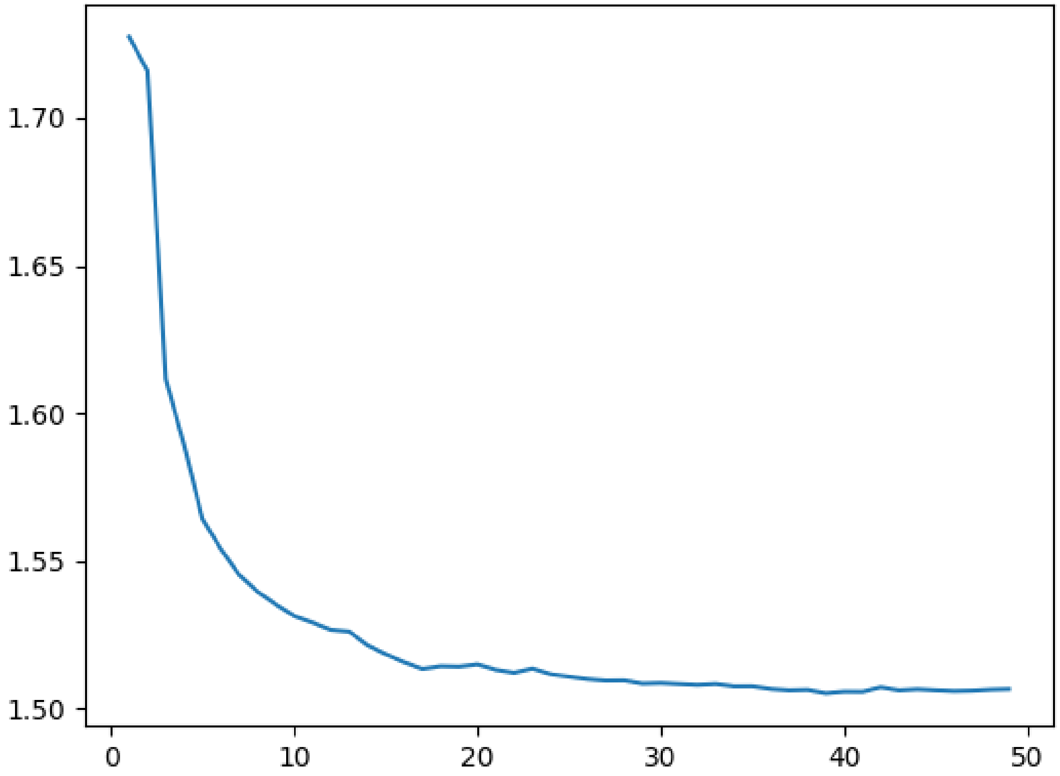
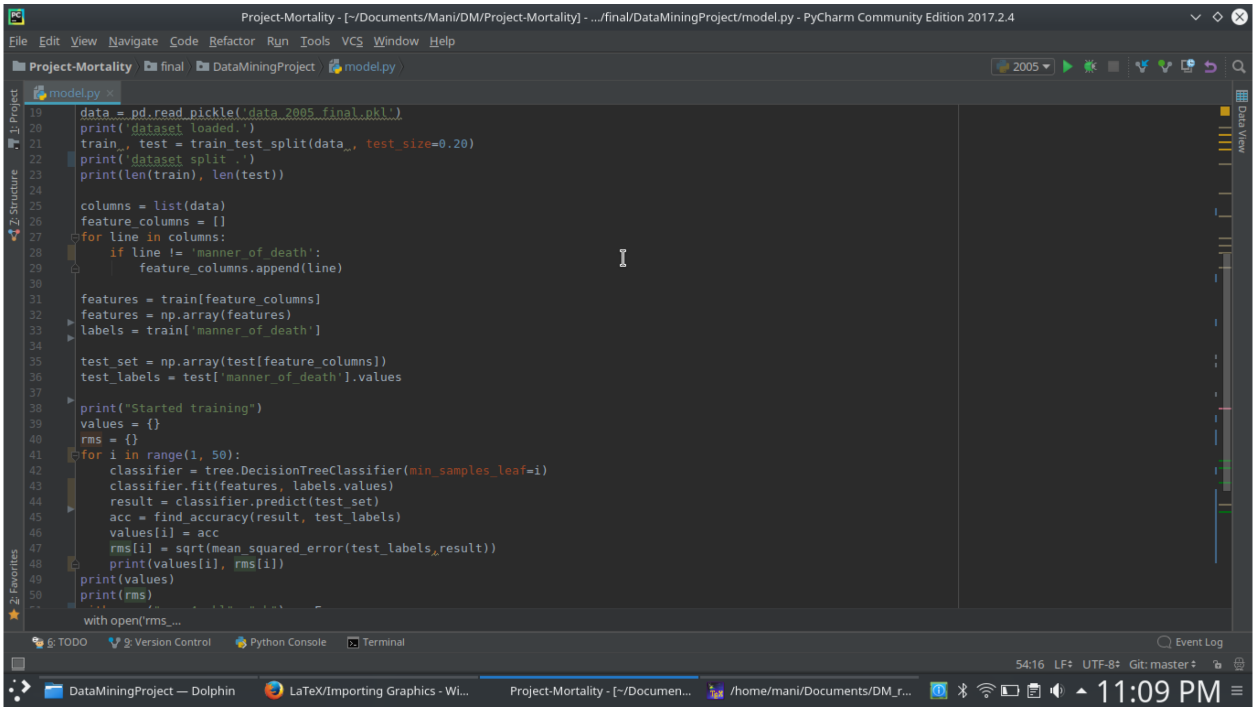


Figure 2: Loss vs Minimum samples per leaf node.

We plot accuracy on the Y-axis and minimum samples per leaf node on X-axis in *figure* .1. It is evident that the acccuracy keeps increasing till one point and saturates after that. Similarly, in *figure* .2 we plot Loss on Y-axis and minimum samples per leaf node on X-axis.

We tuned the model to obtain the most accurate metrics. As evident from *figure*.1 and *figure*.2, the most optimal model for classification corrersponds to the one with minimum samples per leaf node as 45. Thus, we select this model for our final analysis.

With this model, we predict the *manner of death* of the test set containing 20% of the dataset. Our model achieves a test accuraccy of 91.5% and a RMS loss of 1.50 . Thus, the rule based classification has proven to be a good classification algorithm for the selected dataset.


Project-Mortality - [~/Documents/Mani/DM/Project-Mortality] - .../final/DataMiningProject/model.py - PyCharm Community Edition 2017.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Project-Mortality final DataMiningProject model.py
2005
model.py
1: Project
7: Structure
2: Favorites
Data View
6: TODO
9: Version Control
Python Console
Terminal
Event Log
54:16 LF UTF-8 Git: master
DataMiningProject — Dolphin
LaTeX/Importing Graphics - Wi...
Project-Mortality - [~/Documen...
/home/mani/Documents/DM_r...
11:09 PM

```
data = pd.read_pickle('data_2005_final.pkl')
print('dataset loaded.')
train_, test = train_test_split(data_, test_size=0.20)
print('dataset split .')
print(len(train), len(test))

columns = list(data)
feature_columns = []
for line in columns:
    if line != 'manner_of_death':
        feature_columns.append(line)

features = train[feature_columns]
features = np.array(features)
labels = train['manner_of_death']

test_set = np.array(test[feature_columns])
test_labels = test['manner_of_death'].values

print("Started training")
values = {}
rms = {}
for i in range(1, 50):
    classifier = tree.DecisionTreeClassifier(min_samples_leaf=i)
    classifier.fit(features, labels.values)
    result = classifier.predict(test_set)
    acc = find_accuracy(result, test_labels)
    values[i] = acc
    rms[i] = sqrt(mean_squared_error(test_labels,result))
    print(values[i], rms[i])
print(values)
print(rms)
```

with open('rms_...

Figure 3: Code for tuning the model parameters

5.2 Code :

Coming to the programming aspect of the experiment, we initially load the dataset from csv format to a *pandas dataframe* in the memory. Pandas is particularly efficient in loading, cleaning and accessing huge datasets. Once the data is loaded into the memory, we ignored all the illegal and empty tuples in the dataset. We chose to have only people whose education details are as per the norms of 1989 revison. This helps in conducting the experiment to a section of people whose age is above a certain threshold.

5.3 Results :

Using the above discussed methods, we can classify the chosen dataset using a decison tree classifier. The given data is very huge that the visualization of the decision tree generated would be computationally intensive and very complex to interpret from.

Due to this, we visualized the decision tree of a sample dataset.We sampled 5% of the dataset and trained using the same model. The resulting decision tree is as shown in *figure.* 4

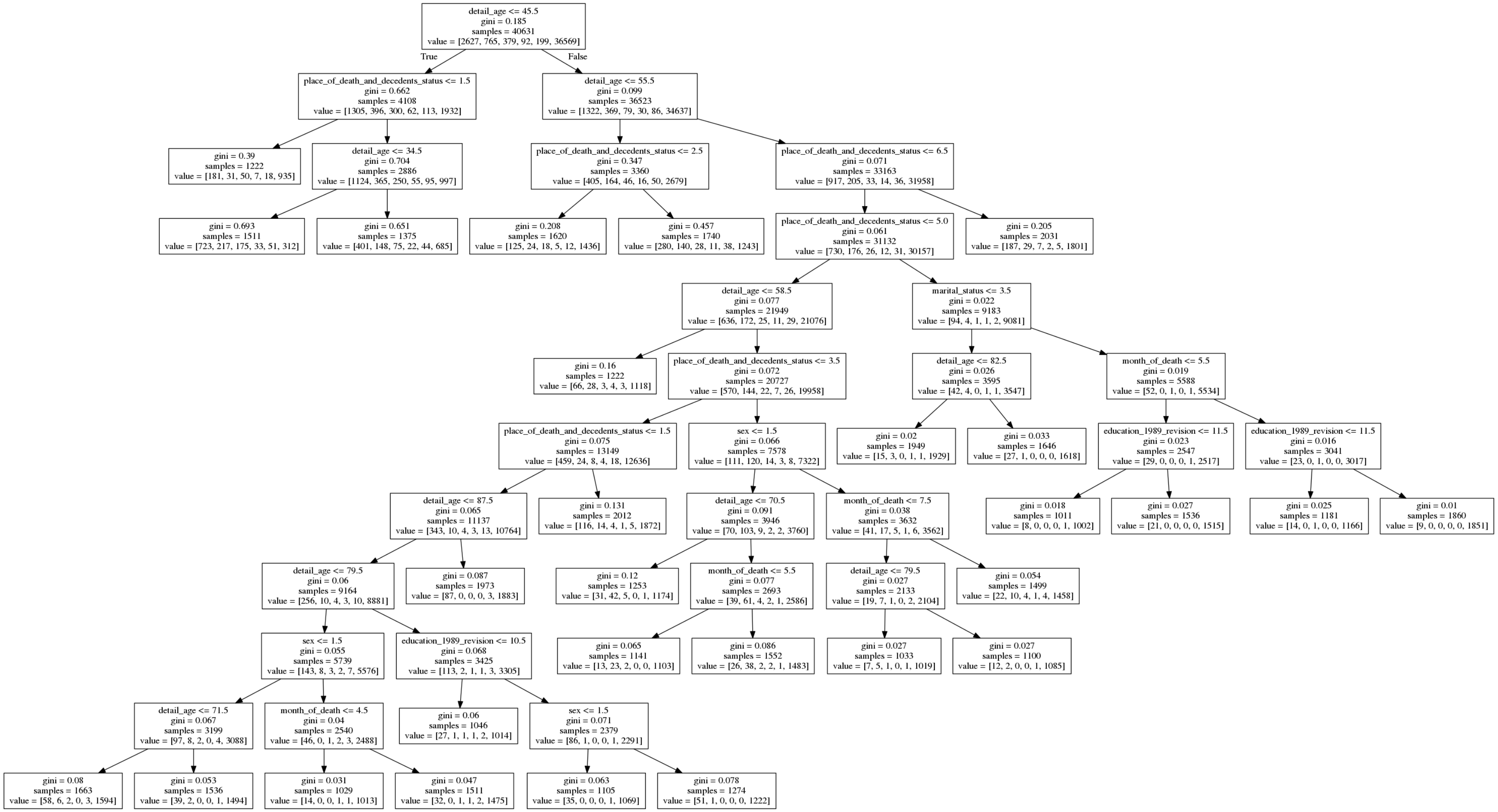

detail_age <= 45.5
gini = 0.185
samples = 40631
value = [2627, 765, 379, 92, 199, 36569]
True
False
place_of_death_and_decedents_status <= 1.5
gini = 0.662
samples = 4108
value = [1305, 396, 300, 62, 113, 1932]
detail_age <= 55.5
gini = 0.099
samples = 36523
value = [1322, 369, 79, 30, 86, 34637]
gini = 0.39
samples = 1222
value = [181, 31, 50, 7, 18, 935]
detail_age <= 34.5
gini = 0.704
samples = 2886
value = [1124, 365, 250, 55, 95, 997]
place_of_death_and_decedents_status <= 2.5
gini = 0.347
samples = 3360
value = [405, 164, 46, 16, 50, 2679]
place_of_death_and_decedents_status <= 6.5
gini = 0.071
samples = 33163
value = [917, 205, 33, 14, 36, 31958]
gini = 0.693
samples = 1511
value = [723, 217, 175, 33, 51, 312]
gini = 0.651
samples = 1375
value = [401, 148, 75, 22, 44, 685]
gini = 0.208
samples = 1620
value = [125, 24, 18, 5, 12, 1436]
gini = 0.457
samples = 1740
value = [280, 140, 28, 11, 38, 1243]
place_of_death_and_decedents_status <= 5.0
gini = 0.061
samples = 31132
value = [730, 176, 26, 12, 31, 30157]
gini = 0.205
samples = 2031
value = [187, 29, 7, 2, 5, 1801]
detail_age <= 58.5
gini = 0.077
samples = 21949
value = [636, 172, 25, 11, 29, 21076]
marital_status <= 3.5
gini = 0.022
samples = 9183
value = [94, 4, 1, 1, 2, 9081]
gini = 0.16
samples = 1222
value = [66, 28, 3, 4, 3, 1118]
place_of_death_and_decedents_status <= 3.5
gini = 0.072
samples = 20727
value = [570, 144, 22, 7, 26, 19958]
detail_age <= 82.5
gini = 0.026
samples = 3595
value = [42, 4, 0, 1, 1, 3547]
month_of_death <= 5.5
gini = 0.019
samples = 5588
value = [52, 0, 1, 0, 1, 5534]
place_of_death_and_decedents_status <= 1.5
gini = 0.075
samples = 13149
value = [459, 24, 8, 4, 18, 12636]
sex <= 1.5
gini = 0.066
samples = 7578
value = [111, 120, 14, 3, 8, 7322]
gini = 0.02
samples = 1949
value = [15, 3, 0, 1, 1, 1929]
gini = 0.033
samples = 1646
value = [27, 1, 0, 0, 0, 1618]
education_1989_revision <= 11.5
gini = 0.023
samples = 2547
value = [29, 0, 0, 0, 1, 2517]
education_1989_revision <= 11.5
gini = 0.016
samples = 3041
value = [23, 0, 1, 0, 0, 3017]
detail_age <= 87.5
gini = 0.065
samples = 11137
value = [343, 10, 4, 3, 13, 10764]
gini = 0.131
samples = 2012
value = [116, 14, 4, 1, 5, 1872]
detail_age <= 70.5
gini = 0.091
samples = 3946
value = [70, 103, 9, 2, 2, 3760]
month_of_death <= 7.5
gini = 0.038
samples = 3632
value = [41, 17, 5, 1, 6, 3562]
gini = 0.018
samples = 1011
value = [8, 0, 0, 0, 1, 1002]
gini = 0.027
samples = 1536
value = [21, 0, 0, 0, 0, 1515]
gini = 0.025
samples = 1181
value = [14, 0, 1, 0, 0, 1166]
gini = 0.01
samples = 1860
value = [9, 0, 0, 0, 0, 1851]
detail_age <= 79.5
gini = 0.06
samples = 9164
value = [256, 10, 4, 3, 10, 8881]
gini = 0.087
samples = 1973
value = [87, 0, 0, 0, 3, 1883]
gini = 0.12
samples = 1253
value = [31, 42, 5, 0, 1, 1174]
month_of_death <= 5.5
gini = 0.077
samples = 2693
value = [39, 61, 4, 2, 1, 2586]
detail_age <= 79.5
gini = 0.027
samples = 2133
value = [19, 7, 1, 0, 2, 2104]
gini = 0.054
samples = 1499
value = [22, 10, 4, 1, 4, 1458]
sex <= 1.5
gini = 0.055
samples = 5739
value = [143, 8, 3, 2, 7, 5576]
education_1989_revision <= 10.5
gini = 0.068
samples = 3425
value = [113, 2, 1, 1, 3, 3305]
gini = 0.065
samples = 1141
value = [13, 23, 2, 0, 0, 1103]
gini = 0.086
samples = 1552
value = [26, 38, 2, 2, 1, 1483]
gini = 0.027
samples = 1033
value = [7, 5, 1, 0, 1, 1019]
gini = 0.027
samples = 1100
value = [12, 2, 0, 0, 1, 1085]
detail_age <= 71.5
gini = 0.067
samples = 3199
value = [97, 8, 2, 0, 4, 3088]
month_of_death <= 4.5
gini = 0.04
samples = 2540
value = [46, 0, 1, 2, 3, 2488]
gini = 0.06
samples = 1046
value = [27, 1, 1, 1, 2, 1014]
sex <= 1.5
gini = 0.071
samples = 2379
value = [86, 1, 0, 0, 1, 2291]
gini = 0.08
samples = 1663
value = [58, 6, 2, 0, 3, 1594]
gini = 0.053
samples = 1536
value = [39, 2, 0, 0, 1, 1494]
gini = 0.031
samples = 1029
value = [14, 0, 0, 1, 1, 1013]
gini = 0.047
samples = 1511
value = [32, 0, 1, 1, 2, 1475]
gini = 0.063
samples = 1105
value = [35, 0, 0, 0, 1, 1069]
gini = 0.078
samples = 1274
value = [51, 1, 0, 0, 0, 1222]

Figure 4: Decision tree for the sample dataset.

The decision tree generated above uses **gini index** for information gain. At each node, it checks for gini index and the minimum number of samples per leaf node and performs the split. The further split is terminated if the Information gain is less than the minimum or if the minimum number of samples per leaf node is reached. This basically means that the data has been classified satisfactorily and no further splitting is required.

The decision tree first chose **detail age** with a gini value of 0.185 and with split condition (detail age <= 45.5). In the second level, it chooses **place of death and decendents status** for split with the condition (place of death <= 1.5). This split leads to several subtrees and the split conditions varies within these subtrees. The split conditions are shown in *figure* .4.

This decision tree is encoded as a dot file by sklearn. This dot file basically consists of all the rules at each level of the tree. The first few lines of the dot file are as shown below :

```
digraph Tree {
node [shape=box] ;
0 [label="detail_age <= 45.5\ngini = 0.185\nsamples = 40631\nvalue = [2627, 765,
    379, 92, 199, 36569]"] ;
1 [label="place_of_death_and_decedents_status <= 1.5\ngini = 0.662\nsamples =
    4108\nvalue = [1305, 396, 300, 62, 113, 1932]"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="gini = 0.39\nsamples = 1222\nvalue = [181, 31, 50, 7, 18, 935]"] ;
1 -> 2 ;
3 [label="detail_age <= 34.5\ngini = 0.704\nsamples = 2886\nvalue = [1124, 365,
    250, 55, 95, 997]"] ;
1 -> 3 ;
4 [label="gini = 0.693\nsamples = 1511\nvalue = [723, 217, 175, 33, 51, 312]"] ;
3 -> 4 ;
5 [label="gini = 0.651\nsamples = 1375\nvalue = [401, 148, 75, 22, 44, 685]"] ;
3 -> 5 ;
6 [label="detail_age <= 55.5\ngini = 0.099\nsamples = 36523\nvalue = [1322, 369,
    79, 30, 86, 34637]"] ;
0 -> 6 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
7 [label="place_of_death_and_decedents_status <= 2.5\ngini = 0.347\nsamples =
    3360\nvalue = [405, 164, 46, 16, 50, 2679]"] ;
6 -> 7 ;
8 [label="gini = 0.208\nsamples = 1620\nvalue = [125, 24, 18, 5, 12, 1436]"] ;
7 -> 8 ;
9 [label="gini = 0.457\nsamples = 1740\nvalue = [280, 140, 28, 11, 38, 1243]"] ;
7 -> 9 ;
10 [label="place_of_death_and_decedents_status <= 6.5\ngini = 0.071\nsamples =
    33163\nvalue = [917, 205, 33, 14, 36, 31958]"] ;
6 -> 10 ;
11 [label="place_of_death_and_decedents_status <= 5.0\ngini = 0.061\nsamples =
    31132\nvalue = [730, 176, 26, 12, 31, 30157]"] ;
10 -> 11 ;
12 [label="detail_age <= 58.5\ngini = 0.077\nsamples = 21949\nvalue = [636, 172,
    25, 11, 29, 21076]"] ;
11 -> 12 ;
13 [label="gini = 0.16\nsamples = 1222\nvalue = [66, 28, 3, 4, 3, 1118]"] ;
12 -> 13 ;
```

6 Social Impact

Understanding the mortality statistics of a country lets the government imporve the medical facilities. In the growing technological world, the suicide rates are disturbing due to high stress levels, emotional distress and other causes. Understanding this could lead to achieving a solution to this problem. We predicted the cause for death of an individual with a sound accuracy, thus enabling us to have better insights about unknown details.