San Fransisco Crime

Introduction

Task

The data used in this project is a repository of incidents of crimes that occurred in San Fransisco between January 2003 and May 2015. It is taken from data analysis competition website, Kaggle. It originally has a test set and training set but for prompt testing and validation, I have used the training set, `train.csv` for my work. With 878,049 observations, the datasets provide more than 12 years of crime reports. Broadly speaking, a useful classification task is to be able to predict the `Category` (the class variable) of the crime given time and location. The source of the data is in the link below.

https://www.kaggle.com/c/sf-crime/data

Running the .Rmd file

Keeping the `train.csv` file in the directory of the .Rmd file should run the code given. Due to use of `doMC` library of the `caret` package, the .Rmd file needs to be run in a UNIX machine if not the lab VMs. Just so this code can be easily run, I have set a variable `m` in the first code block which assigns a value of the size of subset to be used. Given this condition, the file takes less than 5 minutes to produce this report. For what it's worth, a bigger value of `m` would make a better report. The following packages were used for this report: `dplyr`, `lubridate`, `nnet`, `caret`, `doMC`, `data.table` and `phyclust` and need to be installed in R.

Explanatory variables

In the original data, time is given as timestamp in the `Dates` variable and as day of the week in `DayOfWeek` variable. Geographical location is given as logitude and latitude in the `X` and `Y` variables respectively. Location in terms of Police Department District is given as `PdDistrict`.

Variables discarded

For the purposes of this task, I have discarded the `Descript` and `Resolution` variables because they are more useful after prediction is done. I have also taken out the `Address` variable because with 23228 levels, handling it gets intricate. I intend to work on it with geocoding in the future.

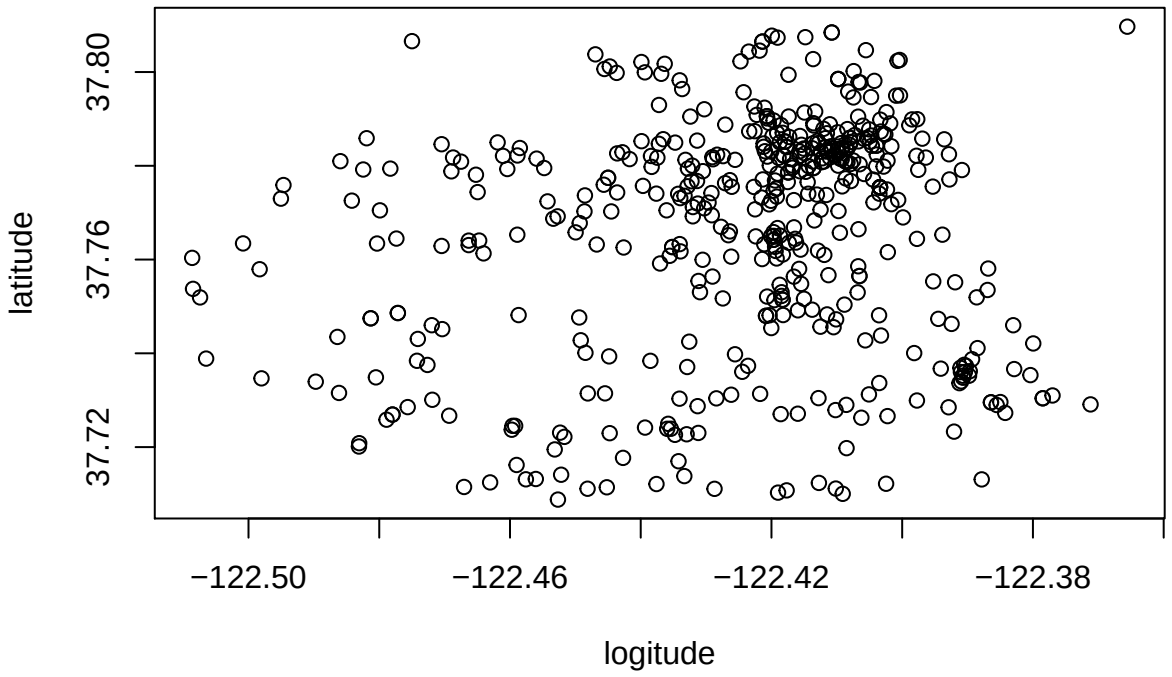
Data Preprocessing

Dealing with date

Since they are just timestamp values, the `Dates` variable in itself is not very useful, so, I have created new seperate variables, `Year`, `Month`, `Day` and `Hour`; discared the `Dates` variable. The new variables are also stored as factors. This decision and the right variable selection has come about after a lot of failed attempts of model training.

Sampling

For quick evaluation and visualisation, arrangements for sampling has been made using a function called `makeSample`. While it is convenient to use sample, it's a good idea to estimate how much memory it would take to work on the entire dataset. Hence, I have also implemented a `getBigMemory` function for the purpose of such evaluation. Given below is a plot of 500 incidents of crime based on latitude and longitude. Under the plot, it shows how much memory (in megabytes) it would take if we were to use the entire dataset.




```
## Memory for entire dataset: 14.4 Mb
```

The models (Supervised)

Before going into the models, we do an evaluation of variable importance using AIC. It turns out, that the location variables and even intercept itself are doing a good job.

```
variable.selection[1]
```

```
## $coefficients
## (Intercept)           Y
##  -422.17230    11.24717
```

Logistic regression

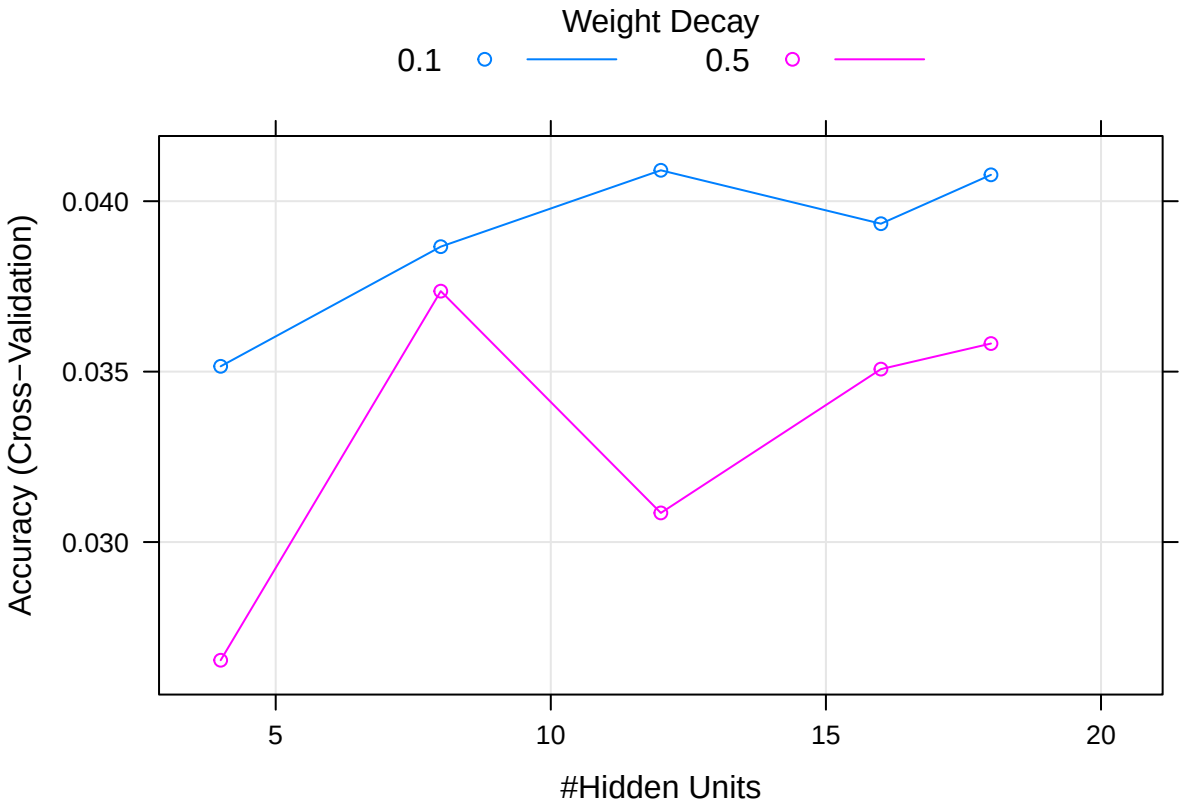
Building training models for a class of 39 levels has proven to be extremely challenging. The first model I have used is logistic regression using `multinom` call of the neuralnet package. It is essentially a neural network with no hidden layers. For evaluating the accuracy of the approach, I have split the data in `7:3` ratio for training and testing respectively. At this point, I have also started to make use of `proc.time()` function to evalute the efficiency of the model generation.

```
## The model took  1.397  seconds to generate
##  Out of  600  test cases, it got 114  right
```

Neural networks

The next approach used is neural networks. Before running the model, I have preprocessed the data with feature scaling. This has resulted in slightly higher accuracy. I have used the `caret` package for it, 2 most important reasons being:

- It has built in implementation for using multiple cores of the machine. The same function without the use of the cores is more than 3 times slower. This fact is reflected in the out below. The number of cores to be used is set to 8 on the basis of a . It is set to 20 when the code is run on the VM.
- It has a built in cross-validation which self-evaluates and builds the right model. In the process, it selects the right value of the regularization term, `decay` from a list that is provided. This helps in prevention of overfitting. Another parameter of neural networks that this model automatically selects is `size` which is the number of units in the hidden layer. Since multiple copies are made during cross validation, it has proven to be a better idea to run this in the VMs where memory is ample. With more RAM available, more parameters can also be taken into account with more interations (i.e. splits and repeats).




```
## The model took  187.659  seconds to generate.
##  But the machine time was  522.941 seconds, indicating parallisation.
##  Out of  600  test cases, it got 29  right
```

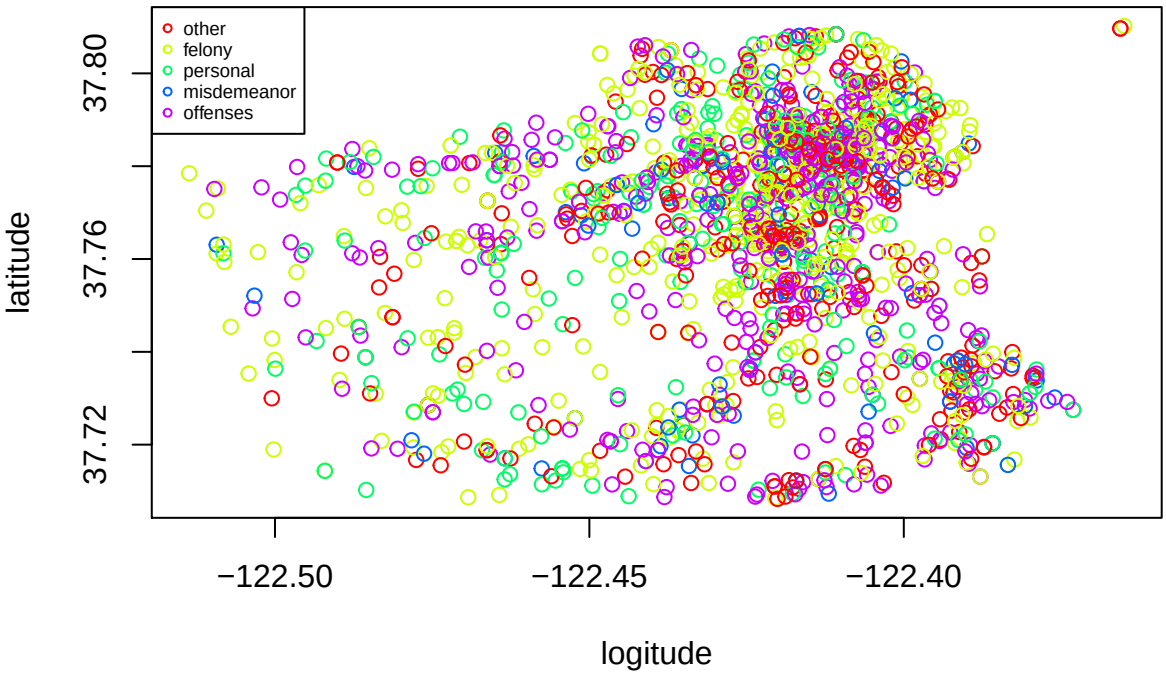
Random forest

For the standards of neural networks, the accuracy of prediction is very low. But the story is not that joyful for random forest either. As it turns out, it failed to even run. Unless I am mistaken, this is most likely because the trees did not find a good split to divide the data in the first node. Nonetheless, the attempt was to generate a model of 100 trees. The values of `mtry`s that were used are 3, 6 and 9. In an ideal result, the value of `mtry` that performed best during cross validation would have been the training model.

The models (Unsupervised)

The two approaches used are principal component analysis in combination with k-means cluster. For meaningful visualisation, the levels of `Category` are further classified into 5 major groups. How well these

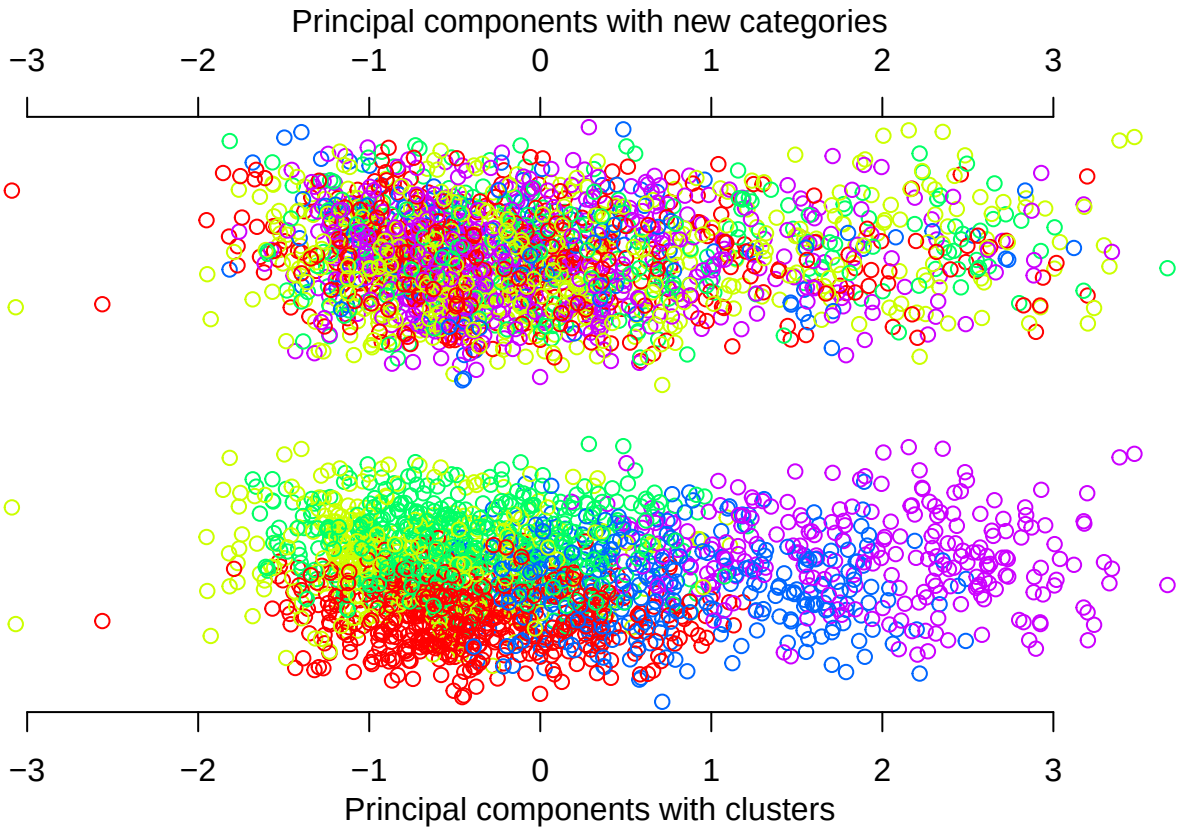
groups are chosen is debatable and certainly can be improved with expert opinions.




```
##      felony    personal misdemeanor    offenses       other
##         443         554         308         148         547
```

PCA and K-means

For unsupervised analysis, first, the 5 new categories are mapped on a 2 dimensional space. Then clusters are generated and applied on the same 2 dimensional space on a different plot. Not surprisingly actual data is not as uniform as the clusters and neither is the `RRand` value very promising.




```
##     Rand  adjRand   Eindex
## 0.655419 0.004199 0.098826
```

Email: cruison@gmail.com