|  |  |
| --- | --- |
|  | pREDICITING THE SENTIMENT OF AN INMATES LAST STATEMENT ON TEXAS’ DEATH ROW  Ciara Edwards 12308356 Harry Quigley 12318661  CA4010 Data Warehousing & Data Mining |

**Section 1: Idea and dataset description**

Our project idea is to predict the sentiment of an inmate’s final statement on Texas’ death row for the years 2012, 2013, 2014 and 2015. We are going to try and predict the emotion which an inmate will express in their final statement by building a classifier using every previous years statement and a range of attributes for each inmate. We created our database using information which we retrieved from the “Texas Department of Criminal Justice” website. Our dataset consists of 519 rows and holds information for every prisoner who has received capital punishment dating from 1986 to 2012. The attributes we have for each of the prisoners are as follows:

* Execution number,Last name,First name,Texas department of criminal justice number,Age at time of execution,date of execution,Race,County,Final statement,Date of birth,Education level,Date of crime,Age at the time of the crime,Height,Weight,Prior occupation,Prior criminal record,Crime committed,Victims race,Number of victims they killed

We also created our own attributes: remorse, religion, denial, anger and acceptance – The sentiments which we wished to predict. These attributes held either a 1 or a 0 depending on if the user had expressed that emotion in their final statement (1 meaning they did express that emotion, 0 meaning they didn’t). We populated these attributes using a Python script which we created to gauge the sentiment of a statement. We explain this further in section 3 of the report.

**Section 2: Data preparation**

As stated above, all the information which we needed to create our dataset was available on the Texas Department of Criminal Justice website at: <https://www.tdcj.state.tx.us/death_row/dr_executed_offenders.html>



We created a webscraper in Python using the Beautiful Soup library. In the code, we travserse through the site’s main table (shown above), storing each inmate’s execution number, name, jail number, age, date of execution, race and county into a separate array for each attribute. We also store the links for each inmate’s last statement and further information into arrays by getting their href’s. The scraper then loops through each link stored in the array containing the links for the last statements. It parses each page until it finds the last statement heading, then gets the text and appends it to a new array. The same is done for the array containing the offender information, which holds most of the attributes that we used in our analysis such as education, prior record, prior criminal record, crime & victims. This particular part of the scraper was tricky as the html tags within which the information we required was located were not consistent , so we had to account for various different index and attribute errors. Another exception that we had to take into consideration was that some of the “Offender Information” links simply led to a JPG image containing the information which obviously could not be scraped. At this point the script had managed to get all the scrapable information on the site into separate arrays for each attribute. Finally these arrays were looped through and written to a CSV file. Afterwards we imported this csv file into a PostgreSQL database. We did this as it was easier for us to view our data and it helped highlight any discrepancies in our dataset. It was useful as we could then perform SQL queries to manipulate and order our data.

Even after we scrapped the data we still had a lot of missing values. Most of the missing values were due to the older records. In these records the offender information was stored as a JPG as opposed to in a table on the website page. Thankfully this was only the case for roughly sixty inmates. The only way we could fill in these missing attributes was by manually inputting them as you cannot infer personal information by running calculations on other inmates. Even after we did this we still had some missing attributes for older inmates - This was because they had changed the information that was stored about inmates in the nineties. The attributes missing were mostly prior occupation and their victims’ race. In this case we just replaced the missing values with a global constant: N/A.

Our dataset had a lot of inconsistencies as we were mostly dealing with language instead of numbers. The first inconsistency was our date attributes. We needed to put these in a standard date format using a data cleaning tool called Trifacta. Some of the inmates gave their final statement in Spanish, so we used a Java language detection library to highlight these. Since we couldn’t accurately translate the statements we had to remove them as it would have introduced bias to our dataset. The last inconsistency was with our crime attribute – We had scraped a detialed description of the crime for this attribue however what we needed was just the actual crime(s) committed. To fix this we ran a Python script that would run through each inmate’s crime description and replace it with the category for the crime committed. For example if it read: “the inmate shot two men dead” it would replace this with murder. We also ran a similar script for “prior record” as the same problem existed for that attribute also.

Our dataset contained noisy data, the main culprit being Unicode characters that had been transferred from the scrape. We removed these from attributes by using the same data cleaning tool as before. Inmates who didn’t give a final statement didn’t add any value to our sentiment analysis as you cannot find the sentiment of a null statement. We removed these inmates when running our sentiment analysis algorithms as they skewed the results. However we kept them in our original dataset as we thought they would add value to our overall analysis.

We also had to perform some data enhancement on our dataset. Originally when we calculated the sentiment score, every time a person showed remorse we added one to their remorse score, also if they used words to emphasise something for e.g. “very” or “extremely” the score of the following word would get doubled. – “very sorry” would get a score of two instead of one. This left us with a lot of different numbers for the remorse attribute and it was causing problems with our decision trees. We decided to replace all these numbers with either one or zero, one meaning the inmate did express the emotion in their statement, zero meaning they didn’t. This gave a better structure to our tree and which resulted in more accurate and structured results.

When implementing our algorithms, we had to decide which attributes to use. We kept the ones that we thought woud have an impact on the inmate’s sentiment. This meant that many attributes which had been very useful when preparing our datasets for carrying out SQL queries and identifying inmates, were now removed as they had absolutely no impact on the sentiment. E.g. first name, date of execution, etc.

In order to get insightful results from our analysis, we had to be able to deal with continuous data before running our algorithms. We decided to use Binning to discretize these attributes – binning structures continous values into intervals.

Finally, our remaining dataset was split into two parts – The training set, which our classifier would be built with and the test set, which would test the classifier. We split the dataset using a 90:10 ratio. The training set consisted of all inmates from 1986 to October 2012. The testing set held all the data from October 2012 to October 2015.

**Section 3: Algorithm description**

In order to predict the sentiment of every inmate’s final statement for 2012, 2013, 2014 and 2015 we first had to get the sentiment of all the previous statements into our training set. To do this we created a script in Python. We chose this programming language as we found it very suitable for natural language processing. We extracted the final statements from our database and copied these into their own csv file. Our program then opened and read all the tokens in this file. It compared each token to the content of our dictionary files. Our dictionary files contained words and phrases which were associated with a certain sentiment, for example “I’m sorry” would be linked to remorse. When the program found a match for a token in the dictionary file it would add the linked sentiment as a tag to the token. We also had a dictionary file for inverting a phrase so if a person put ‘not’ or ‘didn’t’ in front of a word we would not count the score for the sentiment following. After reading a full statement the program would add the scores for each sentiment tag and it would write this score to an output file. The output was in the form of numbers, any number above one stated that the sentiment was present in that statement. If the sentiment was absent it placed a zero in the file. We then added the output file’s values into our database under their respective attributes. We later exported this database into a csv file as it was easier for us to use in our Java programs.

The first algorithm which we decided to implement in our effort to predict the sentiment of an inmate’s final statement was **Naïve Baye’s Classifier**. Naïve Baye’s Classifiers are a family of simple probabalistic classifiers based on applying Baye’s theorem with strong independence assumptions between the predictors. This simply means that the presence/absence of a particular attribute of a class is unrelated to the presence/absence of another attribute. We opted to implement Naïve Bayes because it’s simplicity, efficiency and robustness seemed like a good starting point for us in understanding our data and discovering trends. The below equation represents the algorithm.



In our java code, we first create a Naïve Baye’s Classifier model with the use of our training data set, before testing the model with our testing data set. We made use of the java machine learning library Weka when implementing the algorithm. The steps involved in building a model are as follows, for each known class: Calculate the probabilities for each attribute, conditional on the class value. Using a simple example from our dataset, we can represent these probabilities for the class “Remorse” with attribye “Prior criminal record” in a likelihood table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Likelihood Table | | Show Remorse | |  |
| Yes | No |
| Prior Record | Yes | 106/199 | 87/183 | **193/382** |
| No | 93/199 | 96/183 | **189/382** |
|  | | **199/382** | **183/382** |  |

The Posterior Probablilites can then be calculated, e.g: P(Remorse = Yes | Prior Record = Yes) = (106/199 \* 199/382) ÷ 193/382 = .553

We have to work out probabilities based on multiple attributes – this is where the naïve assumption comes in. Seeing as the attributes don’t depend on each other, the probability of seeing them together is simply the product of their probabilities. So the probability of an inmate showing remorse when they have a prior criminal record and they are of age 45 is simply P(Remorse = Yes|Prior Record = Yes) \* P(Remorse = Yes|Age > 40).

When all the prior probabilities have been calculated, we can then classify the test data. To identify what class the new data belongs in, the algorithm simply uses the Naïve Baye’s Classifier equation on each class and whichever class gives the higher probability is assigned to the new data. So continuing on from the very basic above example, if the classifier was asked to classify a 45 year old inmate who had a prior record to determine whether or not they showed remorse, the result would be calculated for: P(shows remorse|Prior record = yes and age > 40) = P(prior record = yes|shows remorse) \* P(age > 40|shows remorse) /(P(prior record = yes)\*P(age > 40)). This result is then compared with the same equation applied to the class “doesn’t show remorse”. If the outcome of class “shows remorse is higher”, then it is determined that the inmate does show remorse.

(Notes: We carried out binning on the numerical attributes in order to implement Naïve Baye’s theorem. Weka uses a Laplace estimator, which adds 1 to the count for a attribute-value-class combination when an attribute value doesn’t occur with the class value. This is necessary as if not the classifier would classify some situations as being impossible – rather than just very unlikely).

Not happy with the results we had been getting with the above algorithm, and having learned from implementing it, we decided to try again using a different algorithm. **J48** is Weka’s open source implementation of the **C4.5** algorithm. C4.5 generates decision trees which can be used for classification. This algorithm appealed to us as decision trees are useful for identifying important attributes and the relationships between them. The algorithm uses the concept of information entropy. The classifier is created by recursively implementing the following sequence:

1. Check if the termination criteria (base cases) have been satisfied. Examples of base cases: All instances belong to the same class, None of the attributes provide any information gain, & a previously unseen class is encountered. (In the last 2 cases, a decision node is created higher up the tree).
2. Compute the information gain for all attributes. Information gain being equal to the total entropy for an attribute if for each of the attribute values a unique classification can be made for the result attribute. Entropy is a common way to measure impurity of a dataset. The higher the entropy the more informative the content is.
3. Choose the best attribute according to the information gain.
4. Create a decision node based on the attribute selected in the previous step.
5. Split the dataset based on the decision node just created.
6. Make a recursive call on the sub-set of data split in step 5 – We get a sub-tree.
7. Attach the sub-tree to the decision node created in step 4.
8. Return the tree.

When the classifier has been created, we display it in a graphical format. This allows us to easily interpret the results and to predict the class of instances in our test set, simply by traversing down the tree.

**Section 4: Results and analysis**

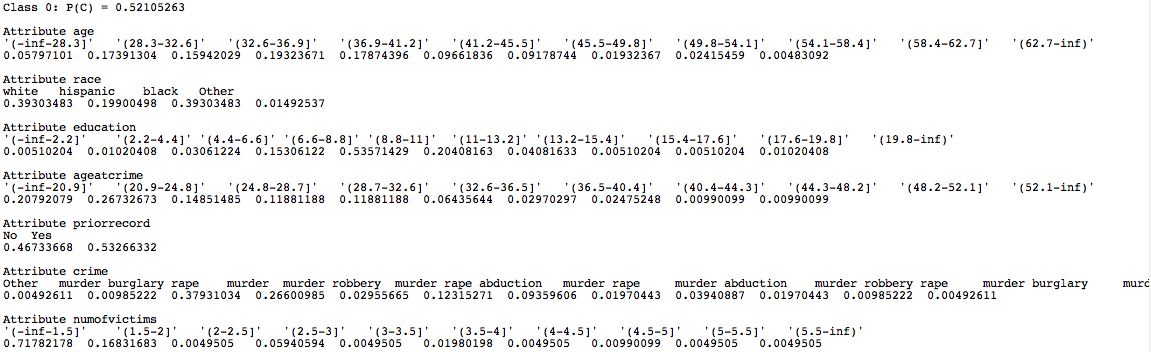
The results produced by our algorithm’s and our insisghts based on them are as follows:

**Naïve Baye’s Classifier:**

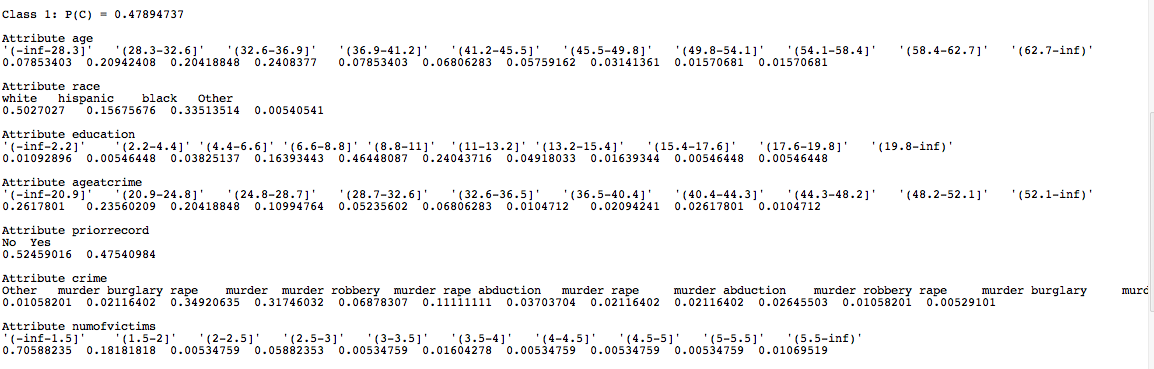
The 7 attributes which we used were: age, race, education, age at crime, prior record, crime and number of victims. All the numerical classes were binned before running the algorithm. We ran the classifier using these for 5 separate classes: Remorse, Religion, Denial, Anger, Acceptance. The output of the java program and an explanation is given for each of the classes.

1. Target Attribute: **Remorse**

Class 0 (Don’t Show Remorse)

****

Class 1 (Show Remorse)

****

**Interesting probabilities**

* We see that 48% of all inmates display remorse in their final statement.
* If an inmate shows remorse the probability of them having a prior criminal record is 0.47540984, but if the inmate doesn’t show remorse the probability of them having a prior record is .53266332. This suggests that repeat offenders have less chance of showing remorse.
* If an inmate shows remorse, the probability of them having murdered 3 or more people is 0.10695186. If an inmate doesn’t show remorse, the probability of them having murdered 3 or more people is 0.10891091. These probabilities are basically the same, which we found surprising as we thought people who didn’t show remorse might have had a higher probability of murdering more people.
* If the inmate shows remorse, the highest probability for their age at time of crime is 17-20 years old with probability 0.2617801. If the inmate does not show remorse, the age at time of crime with the highest probability is 21-24. Maybe because of extremely reckless decisions when younger which they now regret?
* The biggest difference in attribute values between the two classes is for race. If the inmate shows remorse, the probability of them being white is 0.5027027. If they don’t show remorse, the probability of them being white is 0.39303483.

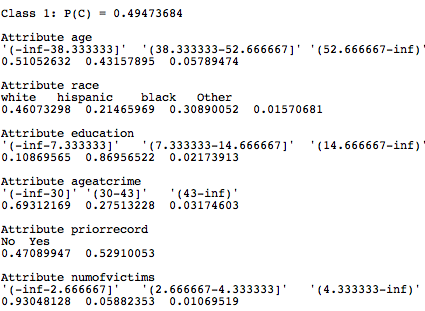
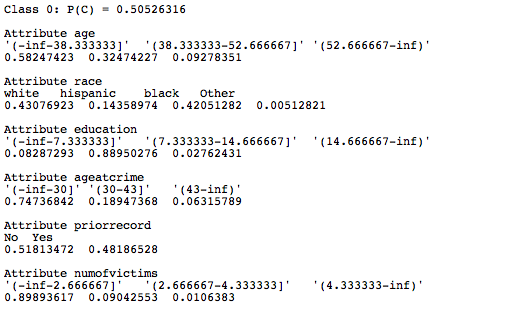
**Evaluation on test instances**

We used the above probabilities to predict the class for each instance in our test data, using the method shown in section 3. We found that for this class, the classifier had quite a low accuracy. It classified just 20 instances correctly out of the 42, meaning that it predicted the incorrect class for 22 of them. This gives it a 48% success rate. 13 of the incorrect predictions were when an inmate who didn’t show remorse was predicted as showing remorse.

Target Attribute: **Religion**

We found that we got better results for religion by removing the crime attribute and binning the numerical attributes into 3 bins rather than 10.

Class 0: Non-Religious Class 1: Religious

****

**Interesting probabilities**

* 51% of inmates display a religious faith in their final statement.
* If the inmate is not religious the probability of them being under the age of 38 when executed is .58. If they are religious, the probability of them being under 38 is less at .5. There is a higher probability of them being in the 39 – 52 age group if they are religious than if they are not. (.43 compared to .32). This made sense to us as usually religious people tend to be older. The last age group (53 and over) goes against this theory however.
* Religious inmates have a .52 probability of having a prior prison record, compared with .48 for non-religious inmates. These figures are very close, so it probably doesn’t mean anything.
* Those who express religion have a higher probability of murdering under two victims (.93) compared with non-religious inmatees (.89). Suggests that non-religious are slightly more likely to carry out crimes with a higher victim count.
* Again, the biggest difference in attribute values between the two classes is for race. If an inmate is religious the probability that they are hispanic is .2 and black .3 – if they aren’t religious this probability drops to .14 for hispanic and raises to .42 for black.
* We thought there might have been a difference in the type of crimes commited by inmates in each class but their probabilties were basically the same – this led us to deleting the “crime” attribute in hope that our classifier would make more accurate predictions.

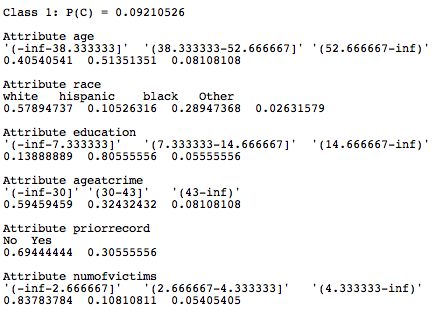
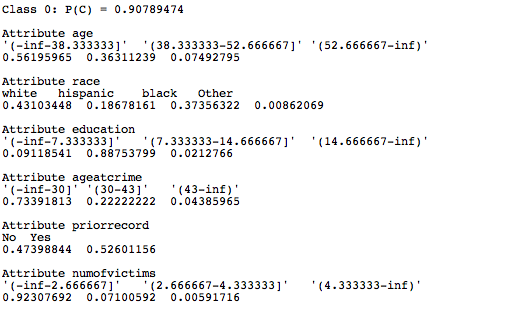
**Evaluation on test instances**

When we used the class to predict whether the death row inmates from 2012 – 2015 would mention religion in their final statement, we found that it had a low accuracy also, although better than the “remorse” classifier (probably due to the removal of the crime attribute). It correctly predicted the class of 22 of the test instances, meaning it predicted 20 incorrectly – giving us a success rate of 52%. The incorrect guesses were split down the middle regarding the class involved.

Target Attribute: **Denial**

Once again, we found we obtained better results by removing the crime attribute and binning numerical value’s in 3 categories.

Class 0: Not in denial Class 1: In denial



**Interesting probabilities**

* .09 probability that an inmate will express denial their final statement. Almost 1/10th of the training set denying that they committed the crime was higher than we thought it would be.
* A trend which we found interesting was that inmates in denial had a much higher chance of having a clean previous criminal record compared to those who didn’t show denial. Those that denied commiting the crime have a 70% probability of not having a prior criminal record, compared to just 47% for those who weren’t in denial.
* There was a noticeable difference in the probability of the inmate’s age between the two classes. Those denying the crime have a probability of .6 for being under 30 at the time of crime compared to .7 for the other class (not denying).
* Once again, there was quite a difference in the probabilities for race. The probability of the inmate being white increased substantially when denial was expressed while it went down for hispanic and black.

**Evaluation on test instances**

When we calculated the posterior probabilities of both classes for each of the test instances, we found that the class was predicted correctly 39 times and incorrectly 3 times. This gives a success rate of 93%, however we would get the same rate by simply predicting class 0 (no denial) every single time.

This is because the test dataset is unbalanced with only 2 out of 42 showing remorse. It’s very hard to predict this class using Naïve Bayes when there is such an inbalance in classes.

Target Attribute: **Anger**

Our sentiment analysis did not pick up any instances of anger in the training set, therefore there was no point in applying Naïve Baye’s Theorem with anger as the target attribute. Correctly detecting it would require a more complex sentiment analysis as it can be implied without the person directly stating their anger.

Target Attribute: **Acceptance**

Similarly, there is only one instance of acceptance picked up by our sentiment analysis script. Acceptance, like anger is usually implied by the speaker’s tone rather than through specific keywords or phrases, which made things difficult for us to detect.

**Naïve Baye’s Theorem Conclusion**

Ultimately, we would consider our implementation of Naïve Baye’s Theorem a failure. It’s accuracy was not high enough in it’s predictions of the test data. We also failed to uncover any really insightful trends or reliable rules using it, although it was very useful for finding very basic statistics in our dataset. We suspect the reason the algorithm wasn’t very useful for predicting the sentiment of inmate’s statements was because of it’s “Naive” independence assumption – that all the attributes are uncorrelated to each other. After implementing it we realised that there are (probably) relationships between the attributes we were using, but we we would have to try use another algorithm to find them.

**C4.5 Algorithm:**

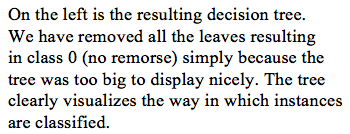
Using the same attributes as for Naïve Baye’s and for the same 5 classes of sentiment, our results were the following:

**Remorse:**

Using the same 8 attributes as earlier:

J48 pruned tree

------------------

age = '(-inf-28.3]': Remorse (25.0/11.0)

age = '(28.3-32.6]'

| priorrecord = No: Remorse (48.0/20.0)

age = '(32.6-36.9]'

| numofvictims = '(-inf-1.5]'

| | crime = Other: Remorse (1.0)

| | crime = murder burglary rape : Remorse (0.0)

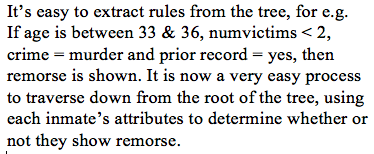
| | crime = murder

| | | priorrecord = Yes: Remorse (9.0/2.0)

| | crime = murder robbery : Remorse (18.0/7.0)

| | crime = murder rape abduction : Remorse (3.0)

| | crime = murder rape

| | | education = '(6.6-8.8]': Remorse (1.0)

| | | education = '(8.8-11]': Remorse (2.72/1.0)

| | crime = murder robbery rape : Remorse (0.0)

| | crime = murder robbery abduction : Remorse (0.0)

| | crime = murder robbery rape abduction : Remorse (0.0)

| | crime = murder robbery burglary : Remorse (0.0)

| numofvictims = '(1.5-2]'

| | education = '(-inf-2.2]': Remorse (0.0)

| | education = '(2.2-4.4]': Remorse (0.0)

| | education = '(4.4-6.6]': Remorse (0.0)

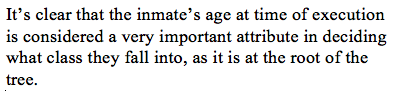
| | education = '(8.8-11]': Remorse (7.78/1.59)

| | education = '(11-13.2]': Remorse (1.08/0.08)

| | education = '(13.2-15.4]': Remorse (1.08/0.08)

| | education = '(15.4-17.6]': Remorse (0.0)

| | education = '(17.6-19.8]': Remorse (0.0)

| | education = '(19.8-inf)': Remorse (0.0)

| numofvictims = '(2-2.5]': Remorse (0.0)

| numofvictims = '(3-3.5]': Remorse (0.0)

| numofvictims = '(3.5-4]': Remorse(1.01)

| numofvictims = '(4-4.5]': Remorse (0.0)

| numofvictims = '(5-5.5]': Remorse (0.0)

| numofvictims = '(5.5-inf)': Remorse (1.01)

age = '(36.9-41.2]'

| race = white: Remorse (45.0/16.0)

age = '(54.1-58.4]': Remorse (8.0/3.0)

age = '(62.7-inf)': Remorse (2.0)

**Evaluation on test instances**

Using the above 8 attributes allowed us to create a classifier which produced the most accurate results for the test set. We experimented by adding back in extra attributes however it’s accuracy then went down. We tried it again using less attributes as we thought overfitting might have been occuring, but the results were less accuarate again. Instead of binning numerical attributes into 10 categories, we tried 3 but to no avail. We also experimented with different levels of post-pruning but were unable to improve the classifier by doing so. (Pruning being a technique which reduces the size of decision trees by removing parts of the tree that don’t help in classifying an instance in a hope to improve predictive accuracy). The decision tree did however work better than our Naïve Baye’s Classifier as it correctly identified 57% of the test data correctly, in comparison with the previous algorithm’s 48% for detecting instances of remorse.

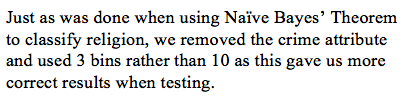
**Religion:**

J48 pruned tree

------------------

race = white

| numofvictims = '(-inf-2.666667]'

| | age = '(-inf-38.333333]'

| | | education = '(-inf-7.333333]'

| | | | priorrecord = No: Not Religious (5.1/2.0)

| | | | priorrecord = Yes: Religious (3.03/0.11)

| | | education = '(7.333333-14.666667]'

| | | | priorrecord = No: Religious (35.82/16.82)

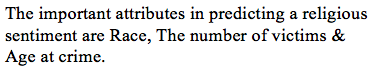
| | | | priorrecord = Yes: Not Religious (30.88/13.0)

| | | education = '(14.666667-inf)': Religious (1.03/0.03)

| | age = '(38.333333-52.666667]': Not Religious (65.85/25.93)

| | age = '(52.666667-inf)': Not Religious (15.93/5.0)

| numofvictims = '(2.666667-4.333333]': Not Religious (11.33/4.13)

| numofvictims = '(4.333333-inf)': Not Religious (1.03/0.01)

race = hispanic: Religious (67.0/27.0)

race = black

| ageatcrime = '(-inf-30]': Not Religious (113.82/42.82)

| ageatcrime = '(30-43]'

| | priorrecord = No: Not Religious (4.0/1.0)

| | priorrecord = Yes: Religious (18.16/5.0)

| ageatcrime = '(43-inf)': Not Religious (3.02/1.02)

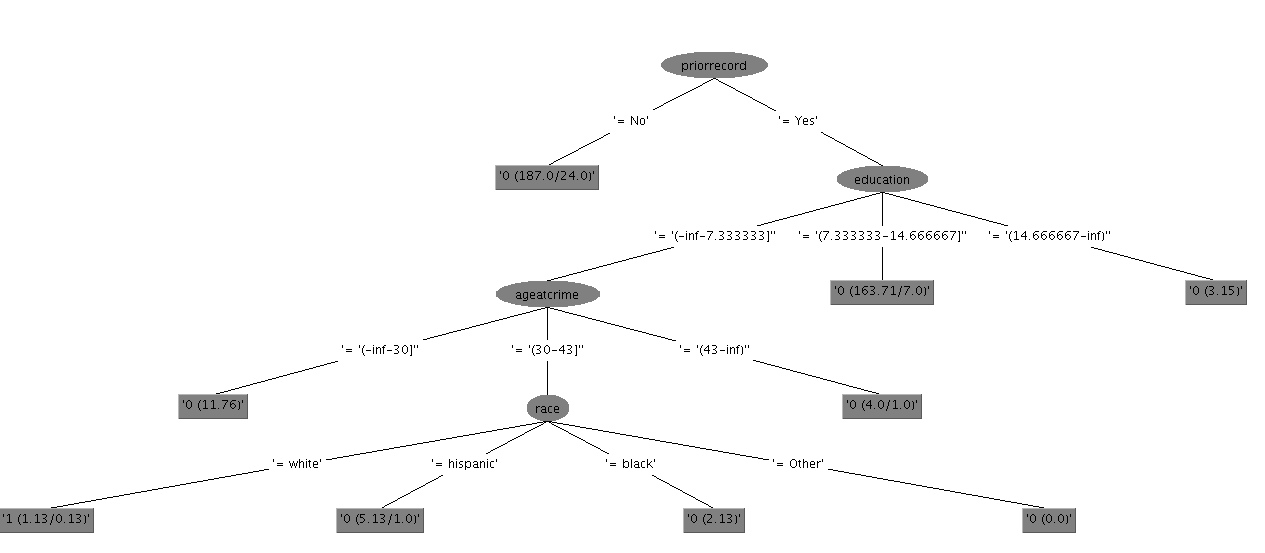
race = Other: Religious (2.0)

**Evaluation on test instances**

Using the rules defined in the above decision tree, we classified 24 instances of the test set correctly, meaning that 18 were incorrect. This gives us an accuracy of 57%. While it still leaves a lot to be desired, the C4.5 algorithm continued to out perform Naïve Bayes’.

**Denial**

J48 pruned tree



Again the crime attribute has been removed and binning into 3 categories has been caried out on the numerical attributes. Prior Record & Education are the attributes which hold the biggest influence over whether or not an inmate will express Denial.

**Evaluation on test instances**

Because our dataset for the “denial” sentiment is so unbalanced – only 32 inmates out of the 467 in the training set express denial – we had to reduce massively the pruning being carried out on the decision tree, as the classifier was simply predicting that the inmate would not show denial 100% of the time. As It does have a high accuracy as it classifys the test instances correctly 93% of the time, however the accuracy would be rated roughly the same if it were to simply pick class 0 (No denial) every single time. Neither of the 2 inmates in the test set who expressed denial received the correct prediction.

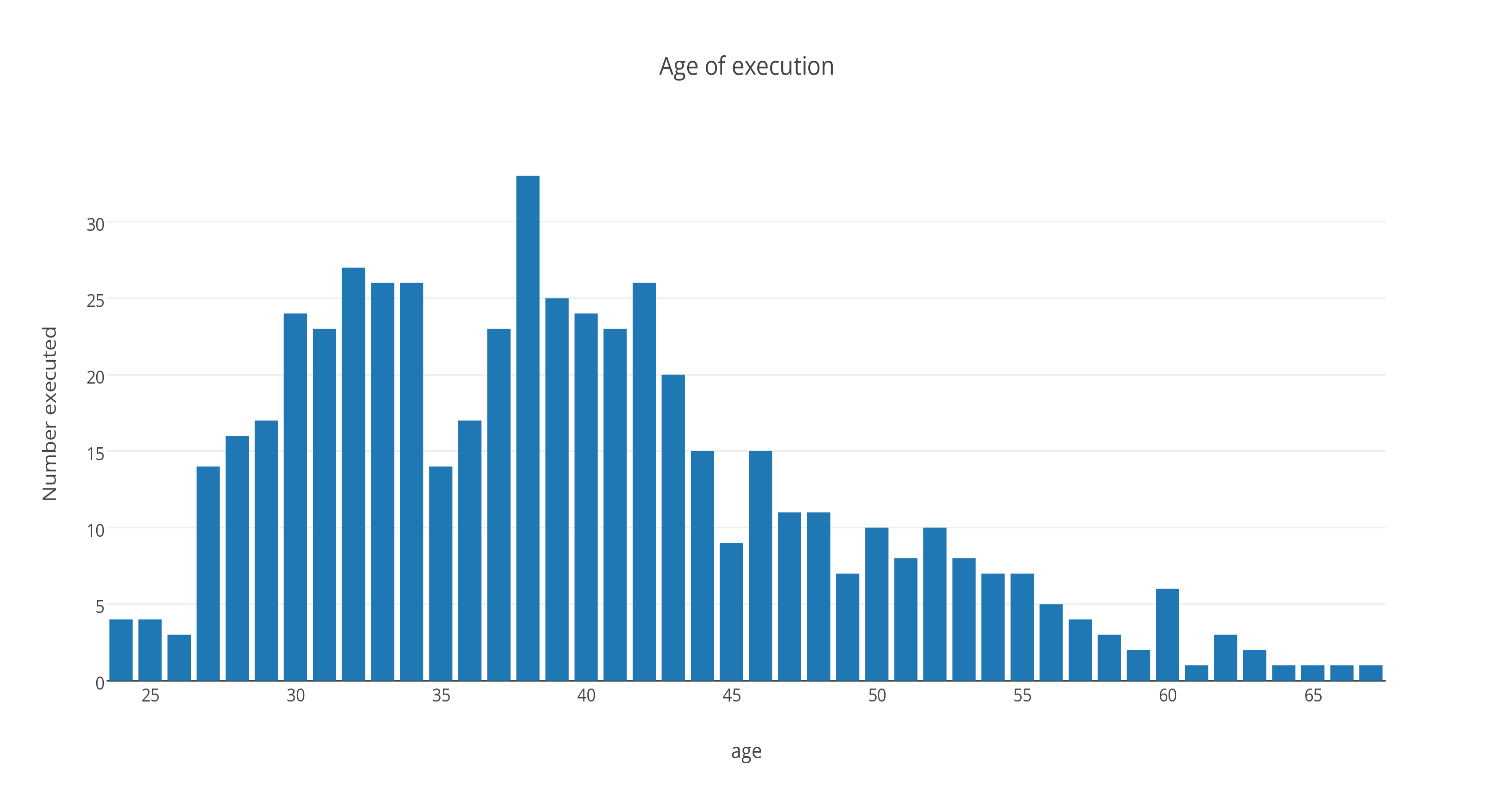
**C4.5 Algorithm Conclusion**

This algorithm was far more effective in discovering trends for attributes and the links between them. We would consider it’s implementation a success as it it predicted the correct sentiment more often than not. Also it was much easier to classify the test instances because of the tree format the results were outputted in.

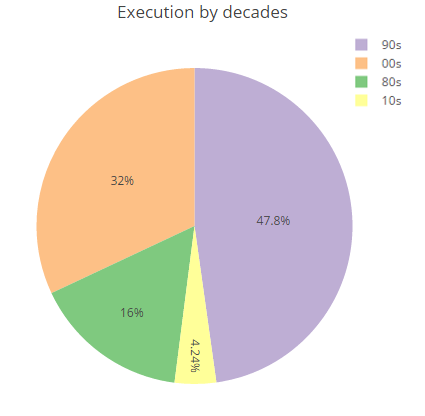
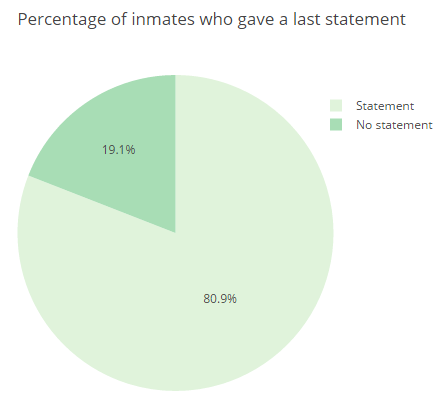
**Final Conclusion**

We realise that our results aren’t very definite and there is a lot of room for error, as we didn’t discover any trend which guarantees to be correct 100% of the time. However we accept this because we realise that predicting sentiment is a lot harder than predicting numbers. This is because the human brain is a lot more complex than eighteen attributes. Not to mention the mental instability we expect many inmates on deathrow would possess. Going on averages the next person executed on death row will be 38 years old with a prior record and less than 10 years in education. They will have commited a murder along with a burglary and their race will be white. Using these attributes along with our algorithms we predict the inmate will show remorse but they will not show denial, anger, acceptance or religion.

**Other interesting findings**

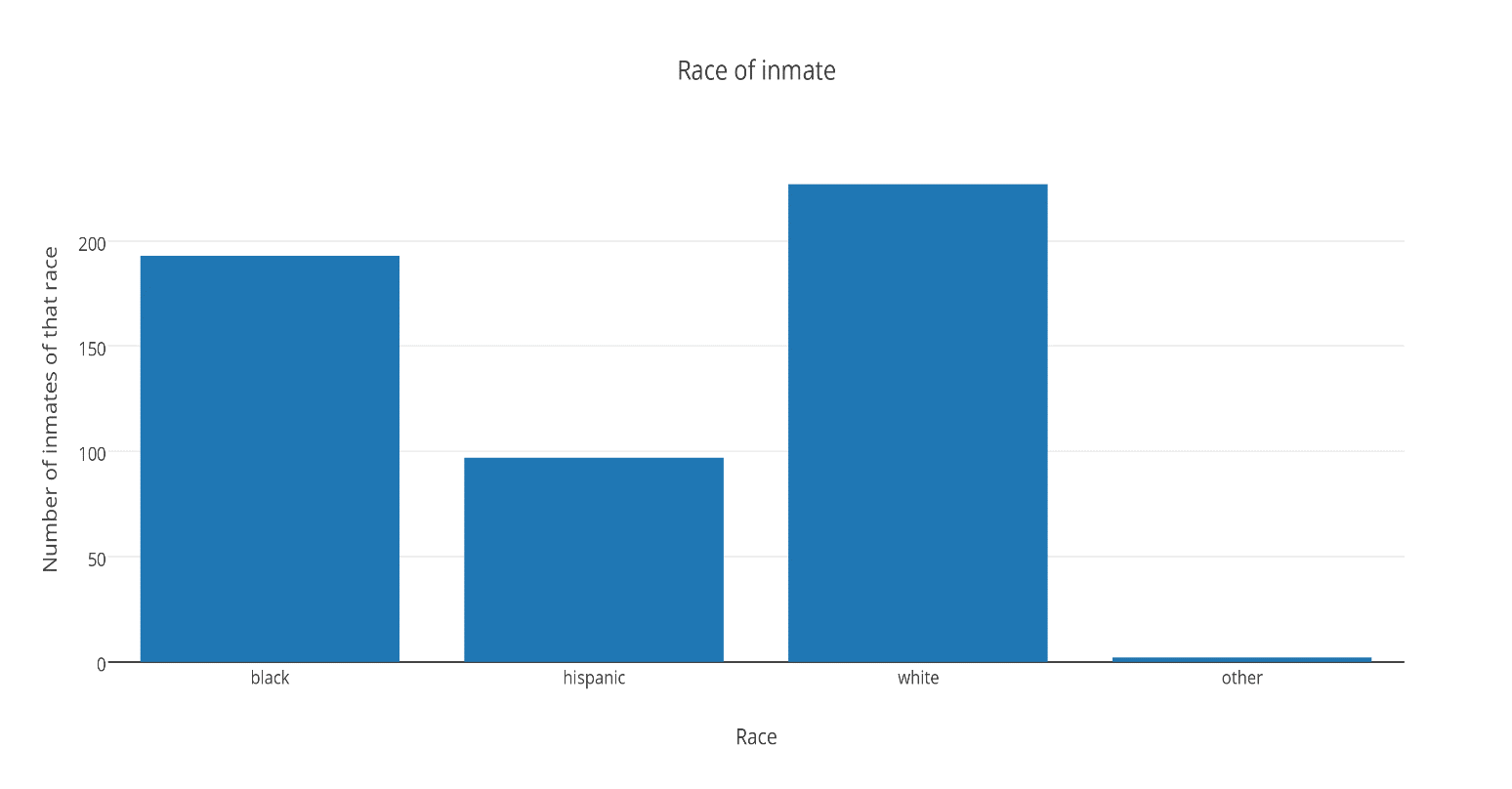
Below are some graphs we have created from attributes in our dataset that we found to be interesting.

This graph highlights that the majority of people on death row are executed by the time they are 45 years old. Given that the average amount of years spent on death row is 12.5 this indicated that most people would have committed their crime in their early twenties.



The graph on the left shows the percentage of executions by decade. We found this particularly interesting as we anticipated that the number of executions would have reduced the closer it got to the present day. We thought this would be the case because opposition to the death penalty has become increasingly popular in America with 19 states abolishing it completely in recent years. This turned out to be true, as there was a 15% drop in executions during the 00’s in comparison to the 90’s.

Our project was to predict the sentiment of an inmate who did give a final statement however we realise that not giving a final statement is also a valid result as 1 in 20 inmates refuse to give one.



From this graph we can see the majority of people killed on death row were white then black followed by Hispanic. The majority of people in Texas are white, however only 12.4% are African American. Hispanics represents 38.2% of the population in Texas.