Homework 3 - Genetics

DATA 901 - Austin Smith

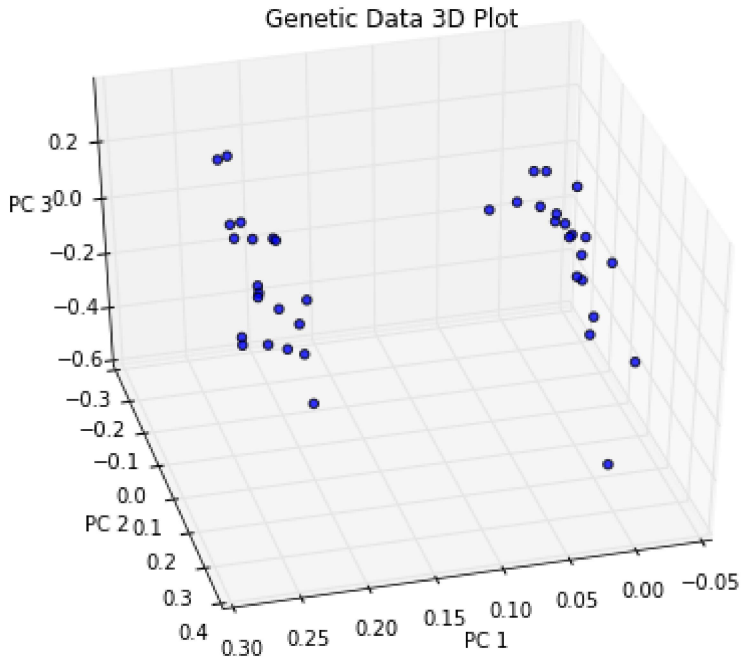
Section 1:

In this part we use gene expression data from 40 tissue samples. The first twenty samples are from patients with cancer and the other twenty are from patients without. In this data, there are 1,000 columns of gene expressions. Below is the first five rows of the data to get an understanding for what it looks like. The Purpose of this section is to compare machine learning algorithms and their ability to determine whehter or not a patient has cancer from the gene expressions.

	0	1	2	3	4	5	6	7	8	9
0	-0.961933	-0.292526	0.258788	-1.152132	0.195783	0.030124	0.085418	1.116610	-1.218857	1.267369
1	0.441803	-1.139267	-0.972845	-2.213168	0.593306	-0.691014	-1.113054	1.341700	-1.277279	-0.91834
2	-0.975005	0.195837	0.588486	-0.861525	0.282992	-0.403426	-0.677969	0.103278	-0.558925	-1.25350
3	1.417504	-1.281121	-0.800258	0.630925	0.247147	-0.729859	-0.562929	0.390963	-1.344493	-1.06711
4	0.818815	-0.251439	-1.820398	0.951772	1.978668	-0.364099	0.938194	-1.927491	1.159115	-0.24063

5 rows × 1001 columns

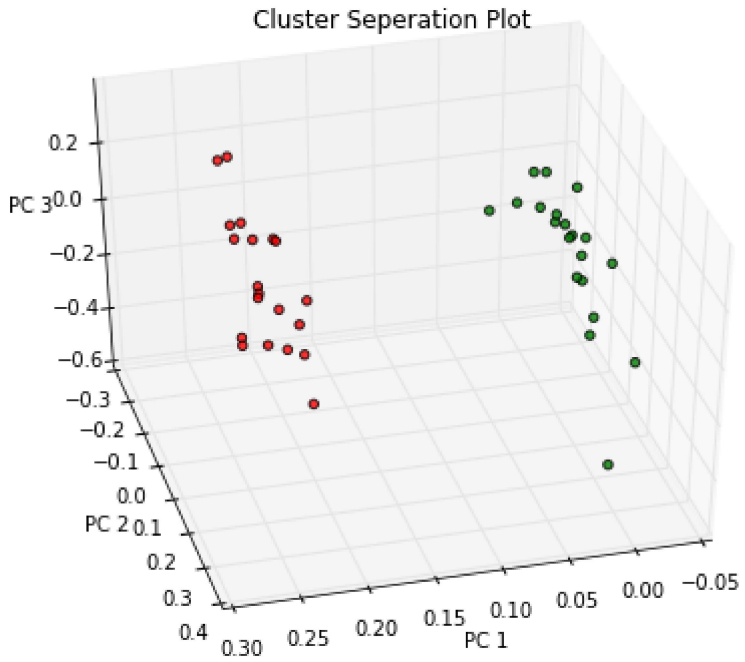
To get an idea of what the data looks like, I created the following 3-dimensional scatter plot. It is clear that there are two distinct groups in the data.



Kmeans Clustering

The first algorithm I will use is Kmeans clustering. This is a technique which is used to find groups of similar points in the data, called clusters. In this example, we know we are looking for two groups, cancerous and not cancerous. The K-Means algortihm is an iterative process where it initializes K (in this case 2) centroids and goes through to detirmine which points are closest to the centroids. The points are grouped by proximity to the centroid. This process is an example of unsupervised learning, since we typically do not know which groups these points belong to prior to clustering. In this case, since we know that the first twenty rows contain healthy patients and the last twenty rows contain patients with cancer, we can determine the accuracy of Kmeans.

The 3-dimensional plot below shows the outcome of KMeans clustering on our data. The algorithm returned 100% accuracy of clustering the groups. Red points are part of the diseased group and green points are part of the healthy group.



Non-Linear Models

Next, we look at the following non-linear models:

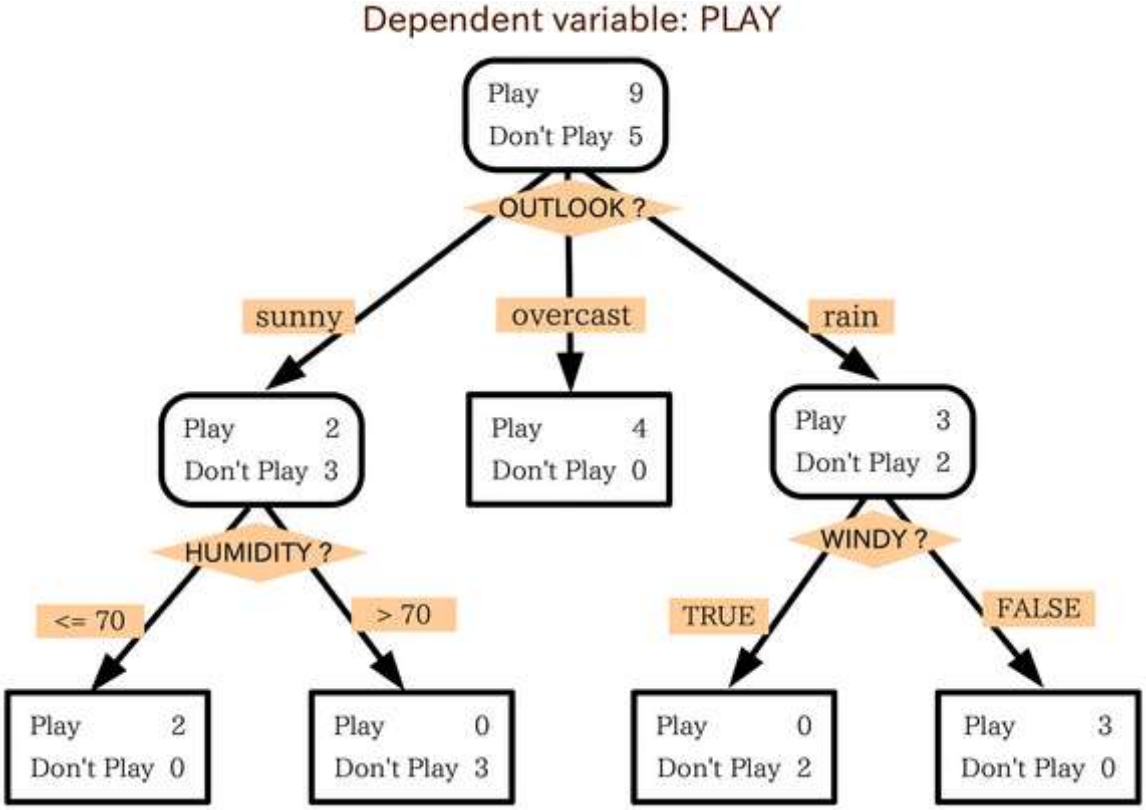
- Decision Tree
- Random Forest
- Boosted Trees

In the execution of these models, I will use two forms of cross validation.

- **Train Test Split:** This is a simple method of validating a model, it involves leaving out a present amount of the data. The model is trained on some data and then prediction accuracy is tested on the data that was left out. In this case 1/4 of the data is left out.
- **Leave One Out:** Runs a train test split using one row as a the test and the rest of the data as training. It iterates through using each row once and generates in this case, 40 seperate accuracy scores. The mean of all of these scores is the cross validated score.

Decision Tree

Decision trees for classification or regression, except, instead of being linear in nature, they form a tree structrure. It does this by breaking the dataset into smaller and smaller partitions and develops a tree. The independent variables can be categorical or continuous. A simple example would be a child deciding if they want to play as shown below.



The algorithim starts at the top and works its way down using greedy search through all options. Decision trees can easily overfit data if they are allowed to run deep enough and split every piece of data.

Below is a confusion matrix for the results of a decision tree run on the data to determine whether or not a patient has cancer, as we can see, it misclassifies almost a quarter of all patients.

```
Decision Tree Cross Validated Mean Squared Error: 0.225
Decision Tree Cross Validated Accuracy: 0.775
```

Random Forest

A single decision tree has shortcomings, mainly the concept of overfitting the data and not predicting out of sample data well. The random forest combats this by fitting a large number of decision trees with randomly selected variables and observations. It is known as an ensemble method since it is aggregating a large amount of trees. The random component of it is in its selection of variables and observations. In each tree, the algorithm randomly selects variables, with replacement. This means that if a variable is selected, it is not excluded for future selection. Thus about 2/3 of the data is selected each time. Next, a random subset of variables is selected for each tree. In a random forest, the accuracy of each tree is calculated by checking the misclassifcation of the observations which were not randomly selected, aka the remaining 1/3. This is known as the out of bag error rate.

Below is the resulting confusion matrix of a random forest used on this data in order to classify patients. As you can see in this case, it achieved 100% accuracy.

	Predicted Cancerous	**Predicted Healthy**
Actual Cancerous	5	0
Actual Healthy	0	5

```
Random Forest Cross Validated Mean Squared Error: 0.0
Random Forest Cross Validated Accuracy: 1.0
```

Adaptive Boosting

In general terms, boosting is an ensemble method similar to random forest, however, it grows many simple decision trees (weak learners) and each successive tree is created for the prediction errors of the previous tree. The adaptive boosting algorithm used below uses the following progression:

1. Start with a small tree, which isn't a very strong predictor.
2. Create a second tree to predict the residuals from the first.
3. Find the residual values of first two trees combined and fit third tree to predict those values.
4. Any observations which are incorrect are weighted more heavily in the next iteration.
5. Classifiers which are accurate predictors recieve more weight and ones which are not accurate predictors recieve less weight in the final model.
6. The output of the model is a weighted average of all the decision trees.

As shown below in this example, this method produced 100% accuracy.

	Predicted Cancerous	**Predicted Healthy**
Actual Cancerous	5	0
Actual Healthy	0	5

```
AdaBoost Cross Validated Mean Squared Error: 0.0
AdaBoost Cross Validated Accuracy: 1.0
```

Linear Models

Next, we look at the following linear models:

- Logistic Regression
- Ridge Regression
- Lasso
- ElasticNet

In the execution of these models, I will use the same two forms of cross validation as used above.

Logistic Regression

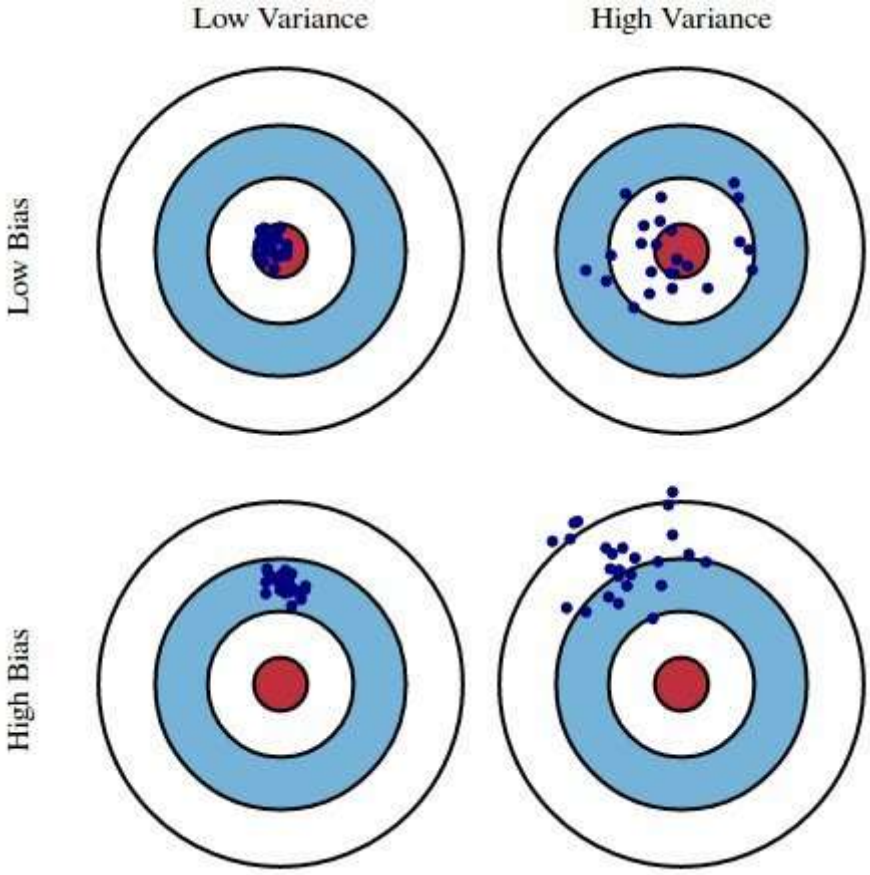
A logistic regression a for of an ordinary least squares linear regression where. Except, the dependent variable is a binary categorical variable, either 0 or 1. The function assigns weights to each variable and produces an output of a decimal between zero and one. Rounding it to the nearest digit yields the classification.
The following table shows a confusion matrix for the logistic regression run with all variables available on the test data after training the model. Additionally, I used Leave One Out cross validation to compute the Mean Squared Error and Accuracy.

	Predicted Cancerous	**Predicted Healthy**
Actual Cancerous	5	0
Actual Healthy	5	0

```
Logistic Regression Cross Validated Mean Squared Error: 0.4
Logistic Regression Cross Validated Accuracy: 0.6
```

Ridge Regression

This is a different type of a linear model that applies a penalty to shrink the coefficients toward zero. In reducing the beta vector there is a variance-bias-tradeoff depicted below:



Variance: is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting, the modeling the random noise in the training data, rather than the intended outputs.

Bias: is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The penalty of the betas is the l2 penalty (sum of betas squared). A tuning parameter lambda is used to tune the model in order to penalize the Betas as little as possible.

Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity. In particular, the ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

Bottom Line:

It is mainly used to prevent overfitting. Since it includes all the features, it is not very useful in case of exorbitantly high number of features, say in millions, as it will pose computational challenges.

It generally works well even in presence of highly correlated features as it will include all of them in the model but the coefficients will be distributed among them depending on the correlation.

In the case of this example, we see that ridge regression was able to get 100% accuracy in both our train/test split and our cross validation, as depicted below.

	Predicted Cancerous	**Predicted Healthy**
Actual Cancerous	5	0
Actual Healthy	0	5

```
Ridge Regression Cross Validated Mean Squared Error: 0.0
Ridge Regression Cross Validated Accuracy: 1.0
```

Lasso

Lasso performs the same function as the Ridge regression, however the penalty is different, in this one, it is the l1 penalty, which is the (Sum of Absolute Value of Betas). This forces some of the independent variables to zero. Aka eliminates them from the model.

A limitation of this method is that if p>n aka there are more variables than samples, it can only use an equal or lower amount of variables. It will not allow there to be more samples than variables. This is a major disadvantage in fields like genomics because the number of genes that can be regressed is limited by the number of samples. However, in this case, we hope it will narrow down to specific fatores in the genes that predict cancer in patients.

Objective = RSS + α * (sum of absolute value of coefficients)

$$Cost\ (W) = RSS(W) + \lambda * (sum\ of\ absolute\ value\ of\ weights)$$

$$= \sum_{i=1}^{N} \left\{ y_i - \sum_{j=0}^{M} w_j\, x_{ij} \right\}^2 + \lambda \sum_{j=0}^{M} |w_j|$$

Bottom Line:

Since it provides sparse solutions, it is generally the model of choice for modelling cases where the number of features are in millions or more. In such a case, getting a sparse solution is of great computational advantage as the features with zero coefficients can be ignored.

Its not hard to see why the stepwise selection techniques become \very cumbersome to implement in high dimensionality cases. Thus, lasso provides a significant advantage.

It arbitrarily selects any one feature among the highly correlated ones and reduced the coefficients of the rest to zero. Also, the chosen variable changes randomly with change in model parameters. This generally doesn't work that well as compared to ridge regression.

This disadvantage of lasso can be observed in the example we discussed above. In highly correlated data, we see that even small values of alpha give significant sparsity (i.e. high number coefficients as zero).

In the case of this example, it predicts with 100% accuracy.

	Predicted Cancerous	**Predicted Healthy**
Actual Cancerous	5	0
Actual Healthy	0	5

```
Lasso Cross Validated Mean Squared Error: 0.033162830147279374
```

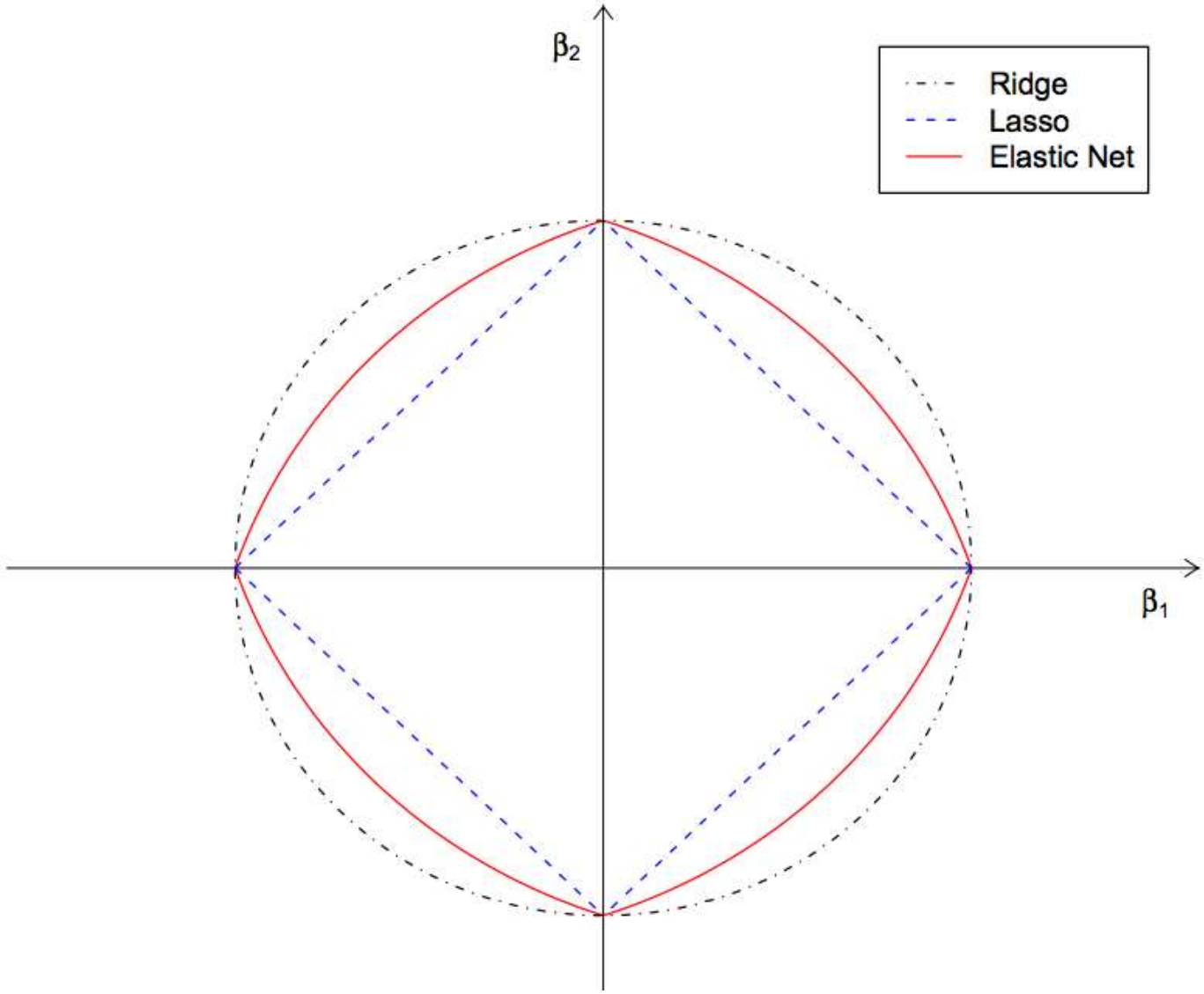
```
Below I have listed the Lasso coefficients that have not been zeroed. The algorithm kept 19
variables as the ones it found important to classify patients with cancer.
```

	Lasso Variables Used
501	-0.068024
592	-0.050167
598	-0.036151
599	-0.034991
583	-0.023773
535	-0.020728
548	-0.015815
561	-0.012308
567	-0.010019
11	-0.006720
559	-0.006600
574	-0.006403
585	-0.006178
553	-0.005507
528	-0.004517
581	-0.003133
12	-0.002400
512	-0.001576
510	-0.001201

Elastic Net

Elastic Net is another useful techniques which combines both L1 and L2 regularization. It can be used to balance out the pros and cons of ridge and lasso regression. It uses both penalties in the error measurement. If L1 is zeroed it becomes ridge and if L2 becomes zeroed it becomes lasso. Otherwise, it is a mix of the two.

The graphic below shows how the three different algorithms arrive at an answer, ridge is on the outside, lasso is on the inside and elastic net is in the middle. As it is a combination of both of them.



As with both Ridge and Lasso, Elastic Net got 100% accuracy in the train/test split.

	Predicted Cancerous	Predicted Healthy
Actual Cancerous	5	0
Actual Healthy	0	5

```
Elastic Net Cross Validated Mean Squared Error: 0.019902046696543572
```

Most Important Variables

To get an idea of what variables are most important in determining wheter a patient has cancer, we can look at the weights assigned to individual variables during the Adaboost algorithm. The higher the number, the more importance. The table below shows the 17 columns which did not have a weight of zero. These columns are the best predictors of whether or not a patient has cancer.

	Feature Importance
588	0.18
534	0.14
10	0.12
569	0.12
560	0.06
564	0.06
535	0.04
544	0.04
587	0.04
12	0.04
599	0.04
591	0.02
501	0.02
545	0.02
568	0.02
527	0.02
551	0.02