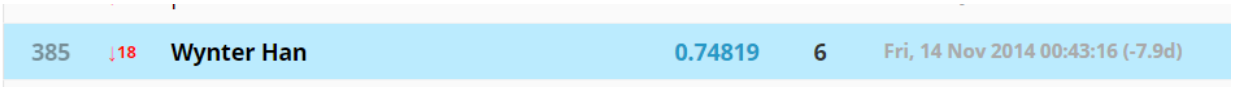
Forest Cover Prediction Report

Dong Han

CSE 581

Nov. 18, 2014

The score on the kaggle.com is 0.74819, which is also the precision rate on test dataset.



Introduction

This is a report for the forest coverage prediction [1]. The training set and testing set have been given as CSV format files. Based on these samples, a classifier should be proposed to recognize different forest cover type, i.e. the tree that covers the area is determined. The wilderness regions are divided into 30x30 meter area. In the dataset, each area is covered by one type of tree. In the report, an artificial neural network (ANN) model is used for the task. I will consider how to visualize the data, and get some insights from the data. Based on the analysis, the data is preprocessed to get appropriate features. Then I build a classifier from these features. I will discuss how to adjust different parameters to improve the performance of ANN classifier. I also consider to use multiple ANN classifiers, which can get better result than just use one ANN classifier.

The following content are arranged as four parts. In the first section, the dataset is introduced. I will discuss some important features of the dataset and show some interesting visualization of the dataset. In the second section, the ANN is introduction. The features extracted from the dataset is discussed in the section. In the third section, the experiment setup and result is introduced. The performance corresponding to different parameters are discussed. In the fourth section, the conclusion is given and some possible improvement is concerned.

1. Dataset analysis

The data of the forest type study is from the region of Roosevelt National Forest. In the forest area, it has four distinct wildernesses. These places are considered without human influence. Therefore, the distribution of tress are nature. The wildernesses area is divided into sub-area. The size of each sub-area is 30x30 meter. US forest Service has investigated the forest. In all sub-areas, their tree types have been determined. The task here is whether we can use information [2] from geographic information system (GIS) to predict the type of tree for each sub-area. The GIS information have been processed and saved as different fields, which include elevation, aspect, and slope. It also has horizontal and vertical distance to hydrology, horizontal distance to roadways and fire points. Moreover, the hill shades information are given for different times, which are 9am, noon, and 3pm, respectively. In addition, the type of wilderness are given. There are four different wildernesses. Each sub-area belongs to one of these wildernesses. The GIS data also considers soil type. Each sub-area has one soil type from 40 different soil types. The above described GIS data has been extracted and provided as CSV file. Each field in the CSV file is one type of GIS data.

The provided dataset has two CSV files. One is train.csv, which has Cover_Type field. The Cover_Type field labels the type of tree in the area. The second one is test file, that is test.cvs, which does not have Cover_Type field. We need to predict the Cover_Type in the test file. After the prediction, we can upload the predicted result to the kaggle.com server. The server can judge the prediction by precision rate of the result.

The training dataset has 15120 samples. The type of trees in the samples are evenly distributed in the training dataset. As shown in Figure 1. Each row in the training dataset is a description of one area. For example a typical sample description in the training data set is:

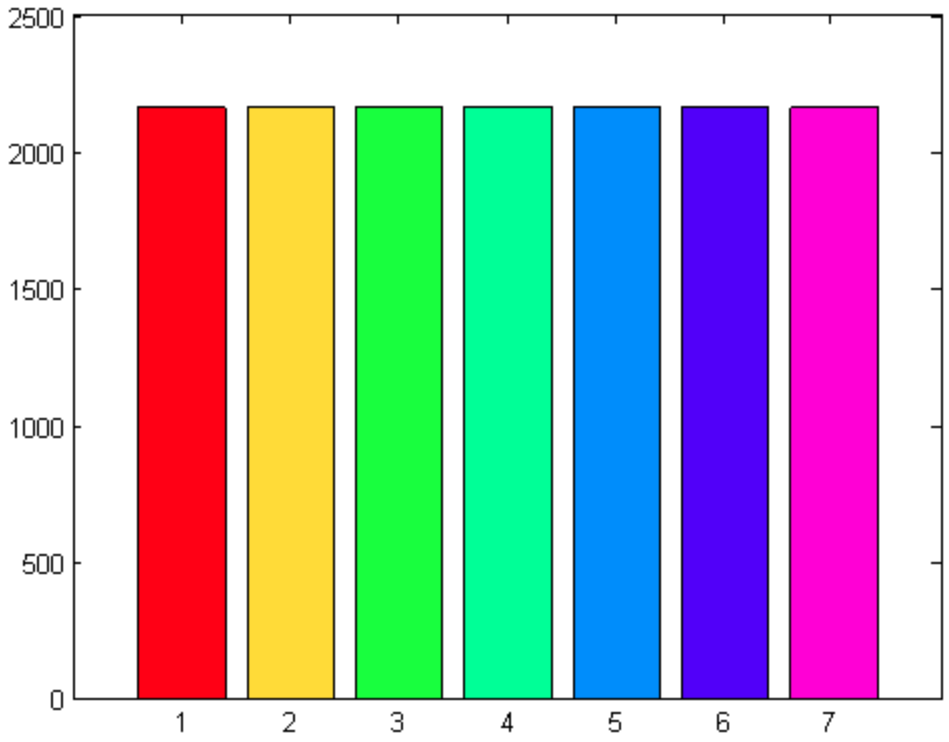


Figure 1: the samples for the types of tree distribution in training dataset. The X axis 1 to 7 specifies seven different type of trees in samples. The Y axis is the total number of samples for a specific type of tree.

Id	**Elevation**	**Aspect**	**Slope**	**Horizontal Distance To Hydrology**	**Vertical Distance To Hydrology**	**Horizontal Distance To Roadway**
1	2596	51	3	258	0	510
Hillshade 9am	**Hillshade noon**	**Hillshade 3pm**	**Horizontal_Distance_To_Fire_Points**	**Wilderness_Area1**	**Wilderness_Area2**	**Wilderness_Area3**
221	232	148	6279	1	0	0
Wilderness_Area4	**Soil_Type1**	**Soil_Type 2 to 39**	**Soil_Type 40**	**Cover_Type**		
0	1	0	0	5		

In the sample data, we can get that the variables like elevation, aspect, and slope are numerical data. Therefore, we can use numerical variable to describe them. Some variables like wilderness area, soil type are categorical data, we can use binary vector to describe them. For example, the sample area belongs to the first wilderness area, we use binary [1 0 0 0] to describe the property. The forest has 40

different soil type, the sample has the first soil type. We use binary [1 0 … 0], which is one with 39 zeros to specify that the sample has the first soil type. Another feature is hills shade. It is a kind of index variable, the value of hills shade is from 0 to 255, each value describe a kind of shade. However, to simplify the input feature, we assume the hills shade is a numerical variable. The output feature cover type is also a categorical data. The value is from 1 to 7. Each cover type is described by a number. When we use ANN model to train the dataset, we should convert the cover type to a 7 binary described vector. If the cover type belong to the 5th type of tree, we use vector [0 0 0 0 1 0 0] to describe the output.

Since there are 54 different input features, it is better for us to visualize these data, then we can consider to generate some new features and consider which feature is more important than others. In the project, I plot all plots for a pair of features from 54 different features. Here we just show a few most interesting ones. All other figures can be found in Appendix.

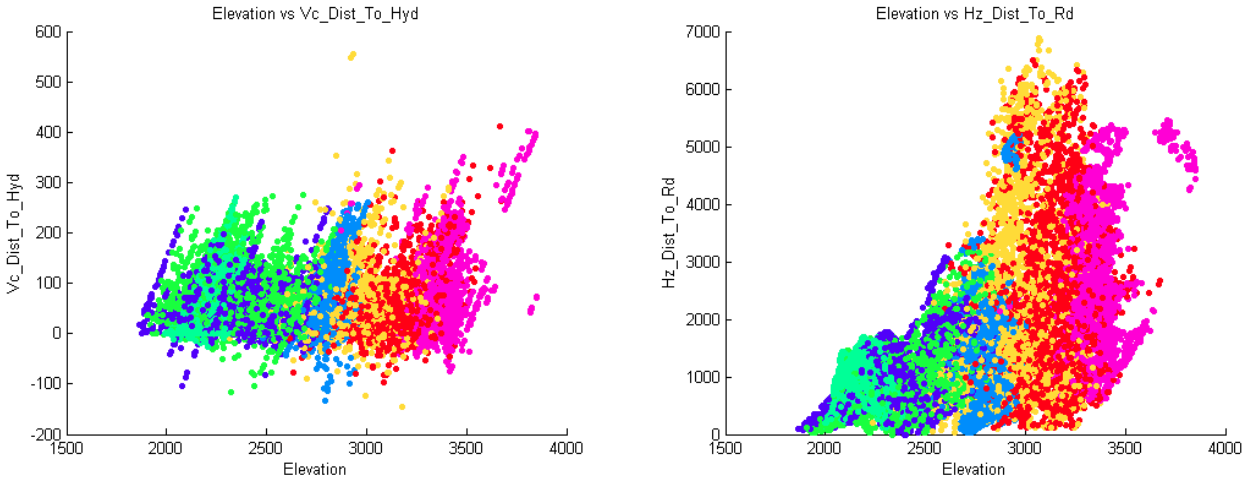


Figure 2: the plot for “Elevation” to “Vertical Distance To Hydrology”, and plot for “Elevation” to “Horizontal Distance To Hydrology”, different color points are different type of tree.

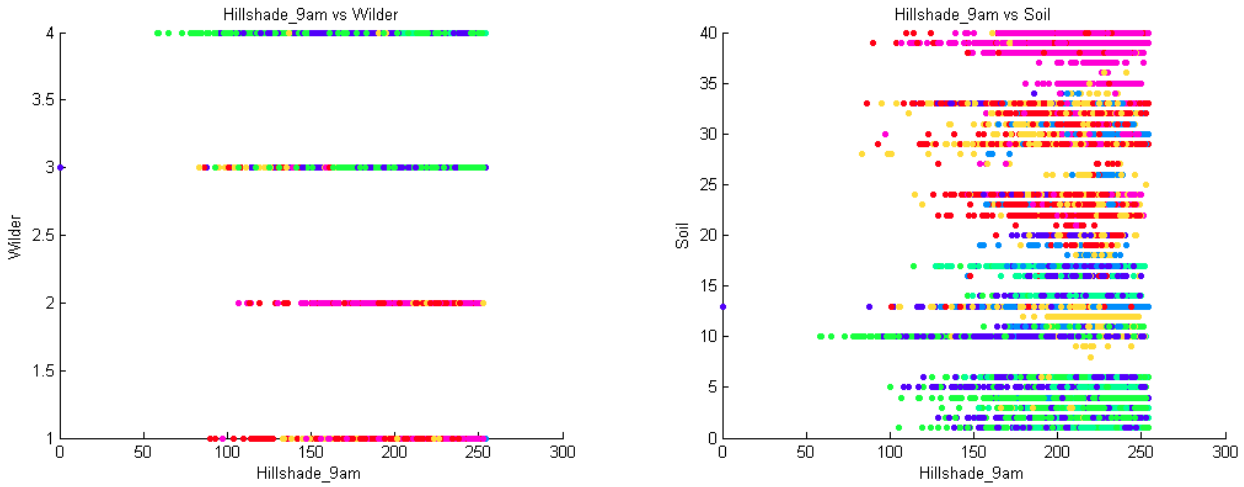


Figure 3: the plot for hill shade to wilderness and hill shade to soil.

The Figure 2 shows the relation of elevation and distance to hydrology. From the plots, we can find out that the trees are distributed following the elevation. For the different level of elevation, different trees are grown. Moreover, we can find out that with the increasing of elevation, the upper value of distance to hydrology is also increased. Also, we have some vertical distance to hydrology has negative values.

From Figure 3, we can get insight that different wilderness grow different type of trees. Also we can have similar conclusion when we view the soil type. These plots have significant features that different wilderness or soil grow different types of tree.

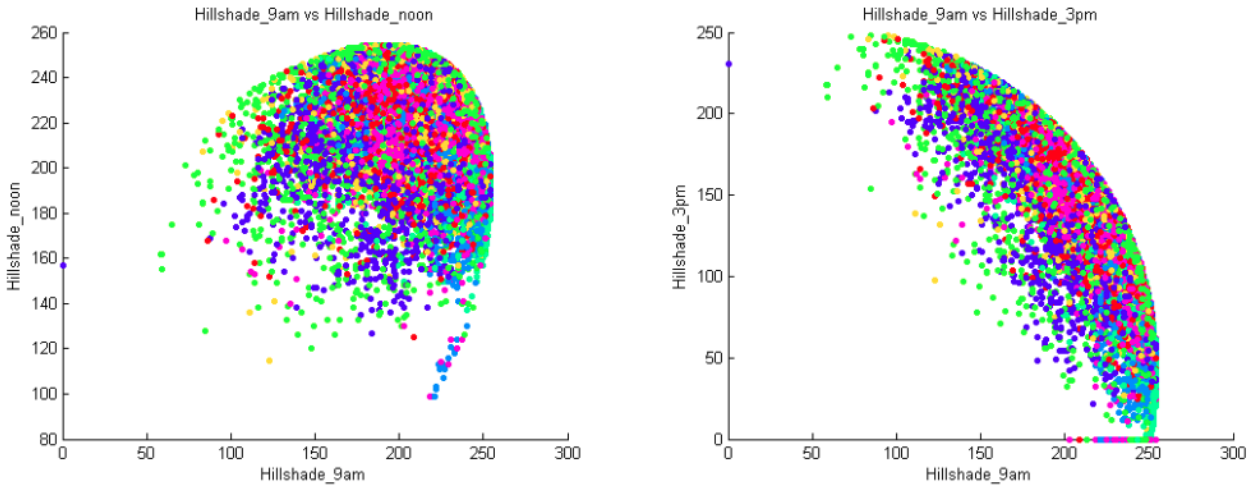


Figure 4: the plot for the relationship between hill shades.

The Figure 4 shows non-linear relationship of hill shades. From the right plot in Figure 4, we can find out that some value of hill shade at 3 pm is zero, this is abnormal. Because these points do not follow the non-linear distribution when comparing with other plots. We consider that the records in hill shade 3pm have some errors. In our data processing, we can use some random value to replace these zero records. These points are also discovered by another team [3] who are working on the forest cover prediction project.

From the result of plots and physical meaning of features, we can generate additional new features. For example, form the feature of horizontal distance to hydrology and vertical distance to hydrology, we can get the Euclidean distance to hydrology. We use the similar method as reference [3] and generate 10 new features. These new features are listed in the table.

Table 1: Description for 10 new features

Features	Description
The sign of vertical distance to hydrology	If vertical distance is positive, the sign is positive. If vertical distance is negative, the sign is negative.
Elevation and vertical distance to hydrology	Elevation minus vertical distance to hydrology

Elevation and horizontal distance to hydrology	Elevation minus horizontal distance to hydrology
Distance to hydrology	The Euclidean distance to hydrology
Horizontal distance to fire point and hydrology	The horizontal distance to fire point plus horizontal distance to hydrology
The offset of Horizontal distance to fire point and hydrology	The horizontal distance to fire point minus horizontal distance to hydrology
Horizontal distance to fire point and roadway	The horizontal distance to fire point plus horizontal distance to roadway
The offset of Horizontal distance to fire point and roadway	The horizontal distance to fire point minus horizontal distance to roadway
Horizontal distance to hydrology point and roadway	The horizontal distance to hydrology plus horizontal distance to roadway
The offset of Horizontal distance to hydrology and roadway	The horizontal distance to hydrology minus horizontal distance to roadway

Since in the table, these features are combination of two existing features. The new features can help classifier to classify tree types easier than before. In the experiment section, we will evaluate that by adding these new features, how much benefits we can get in performance evaluation.

2. Classification processing

I propose to use ANN to predict the forest cover type. There are several reasons to use ANN as classification model. Comparing with traditional statistical method, the ANN can build a highly non-linear model. The applications of ANN have gotten some descent result in these pattern reorganization studies [4][5]. In this section, I will introduce the ANN we applied in the project.

In the project, we use two different types models based on ANN as shown in Figure 5 and Figure 6. In Figure 5, we use one ANN classifier. The input layer has 54 nodes to accept 54 features from records. The hidden layer has 48 hidden nodes. The size of hidden nodes is difficult to be determined. We run experiment to get the best size of hidden nodes. When some new features are added, the input nodes will make changing correspondingly. The output has seven nodes, which can give a vector with seven elements. To get a result that specifies a tree type, we round the output result to 1 or 0. The output is restricted to have only one element is assigned to 1. The element that has value 1 is considered as the type of tree from classification.

To avoid over fitting problem, the early stopping policy is applied in the training process of ANN. To run the early stopping validation, the training set is divided into two parts. One part is used for training that is adjusting weight in the network to fit the output result. The second part is used for validation, which is called validation set. After each training iteration, the validation is applied. If the performance in validation set is not improved in continuous iteration in training process, the training is considered over

fitting. A over fitting model can get high evaluation result on training set, but it performs bad on testing set, that is not used in training. To avoid the situation, a parameter is settled as maximal allowed maximal failed validation. If the parameter is too small, the training process may terminate at a local optimal result which is not the best option. In a training process, if there are continuous validation failures, and the failure times exceeds the predefined maximal allowed failed threshold, the training process is terminated and retreated to the point that gotten the local optimal result. The parameter for maximal literation is also important in training ANN. This parameter shows when training is finished though it is not convergence. This is another policy that to avoid over fitting problem.

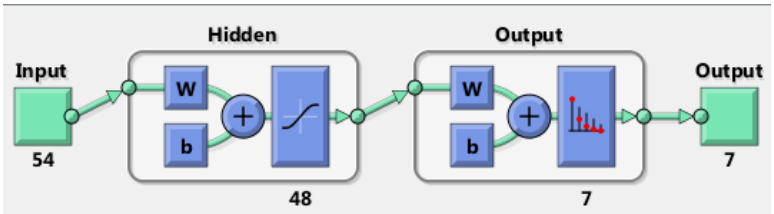


Figure 5: a typical neural network structure

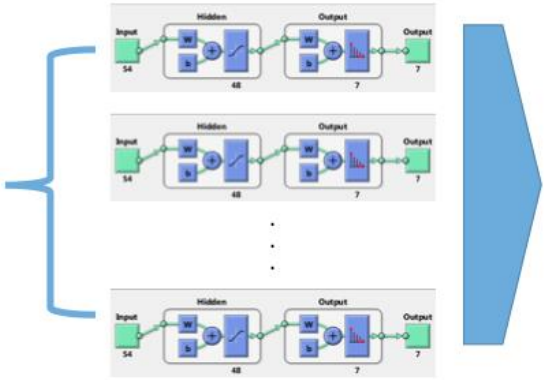


Figure 6: a group of ANN for classification

In the model that use a group of ANN for classification is another method that can improve the performance. From one aspect, we will train many ANN. Some ANN can get better result than others. An intuition here is that we can collect the outperform ANN together, and let them to make decision for the classification. In Figure 6, we put some trained ANN together, they accept the same input. But they have different number of hidden nodes and weights. The output based on majority vote. We do a summation for all the result, and put the summation in to a vector. The element in the vector has the highest value is considered as the type of tree that is predicted. Usually, we can use a group of ANN to increase accuracy. However, the number of ANN in the group should be carefully determined. A large number of ANN not means a better result. In the project, we will list the performance of trained ANN by descend order, then choose top 1, top 10, and top 100 ANNs. For the top1 ANN classifier, we use it directly to predict tree types in testing set. For top 10 and top 100 ANNS, we use them in the group classification model for prediction.

3. Experiment

The classification is implemented using Matlab R2014a. In matlab there is a Neural Network toolbox that we can use. It has already provided neural network for pattern classification. We use the toolbox, and just its parameters to an appropriate value. There are a few things we can tweak for the provided model. We can change its number of input nodes, the number of hidden nodes and the number of output nodes. We can also override its default normalization function which is applied at the input nodes. Some other parameters such as maximal iteration size, the maximal validation failed number can also be customized.

The different parameters are applied for training ANN. We use different size of hidden nodes for ANN classifier. The number of hidden nodes are {40, 60, 80, 100, 120, 140, … , 220}. Once the number of hidden nodes is determined, we consider to train it with different initial random weights. In this step we train 50 ANN that have the same number of hidden nodes with different initial weights. In all, we get 500 different ANN classifiers. Then we just the top 1, top 10 and top 100 for evaluation.

The performance of a classification is evaluated as precision, which we consider the ratio as the number of corrected classified samples to the total number of samples, as shown in the following formula.

$$Precision = \frac{TP}{TP + FP}$$

3.1 Experiment for normalization policy

The Matlab ANN toolbox has its default normalization policy which is not appropriate in the predication task. For numerical variable, its range is mapped to [-1 1] interval. For example, the elevation variable is mapped from [1859, 3859] to [-1 1], the horizontal distance to hydrology is mapped from [0, 1397] to [-1 1], and the vertical distance to hydrology [-173, 601] to [-1 1]. But the problem of the default mapping policy is in mapping of categorical variables. The default normalization maps a categorical variable, say, the first type of wilderness [1 0 0 0] to [1 -1 -1 -1], which does not make sense for this type of variable. Therefore, the proposed normalization policy in the task is that we do not make any categorical variable to [-1 1], and keep them as them are.

To make the mapping more convincing. We calculate the range of a feature based on its value in training set and testing set. If we just use training set to get its range, this range may not be exactly the same as the range on testing set. By calculation the range on the data set that is the merging of the two dataset, we can normalize the value from training or testing set to [-1 1].

The comparison of the default normalization policy and our proposed normalization method is shown in Figure 7. Form the result, we can point out that our normalization method get a better result on both training dataset and testing dataset. In the experiment all other parameters, we use default values.

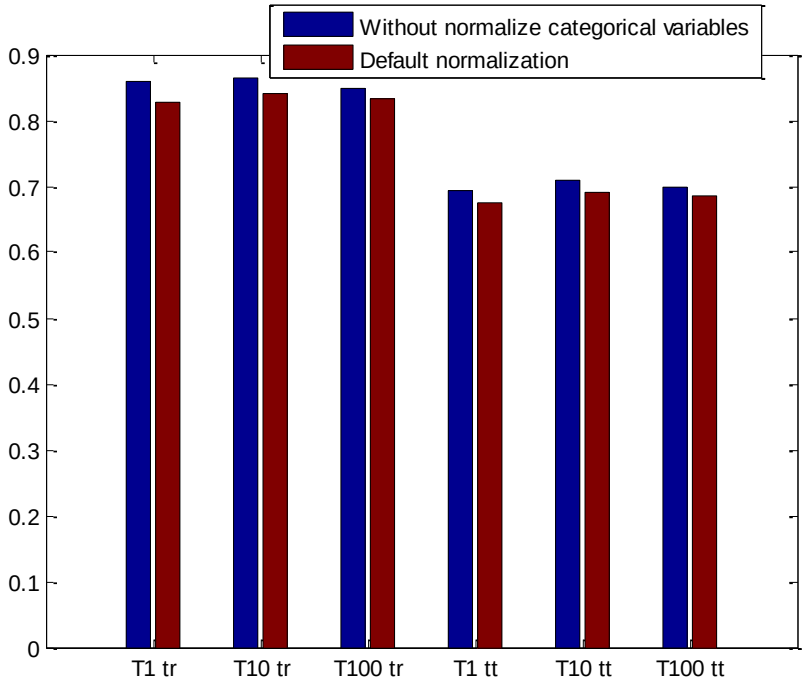


Figure 7: The “T1 tr” means that the top 1 ANN classifier is evaluated on training set. The “T2 tr” means the top 10 ANN group model classifier is evaluated on training set. The “T100 tr” means the top 100 group model classifier ANN is evaluated on training set. For the result of “T1 tt”, “T10 tt”, and “T100” mean the top1, top 10, and top 100 classifier model are evaluated on testing set. The y axis shows their precision.

3.2 Experiment for early stopping parameter setting

The maximal allowed validation failure is an early stopping parameter. The default value of the parameter is 6, which is considered too small in the training task. However, if the parameter is too large, the training process will be over fit. In the training task, we compare the cases that the maximal allowed failure as 6 and 128. We also consider a larger value, such as 256. Since the maximal iteration is 1000, the value that larger than 128 has the similar result as 128. The result is shown in Figure 8.

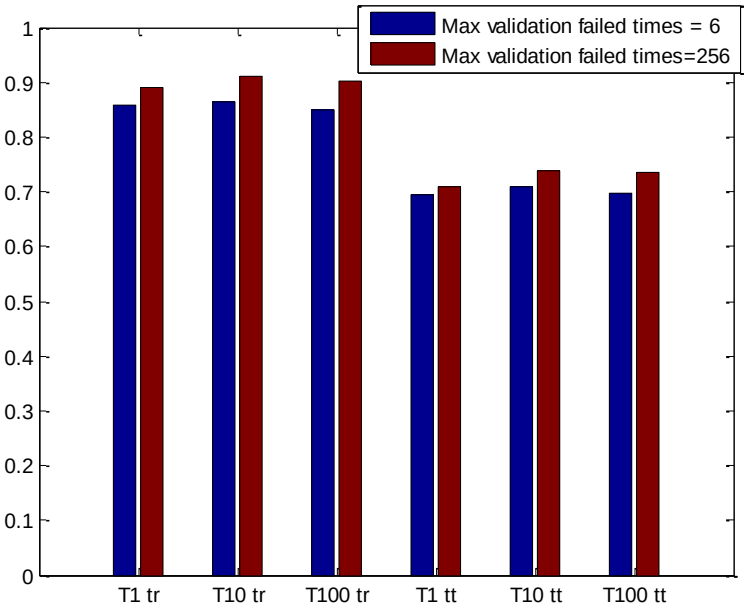


Figure 8: The experiment for ANN with different maximal validation failed times.

3.3 Experiment for adding new features

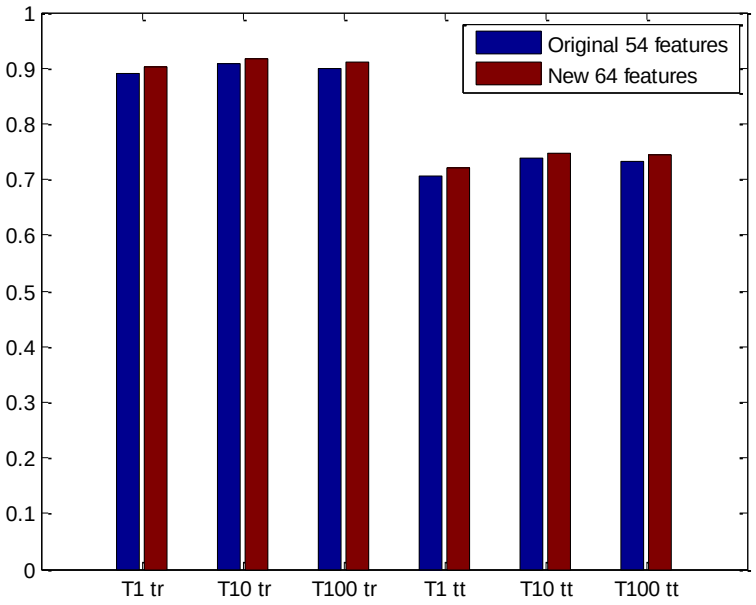


Figure 9: The experiment for 54 features vs. 64 features.

As I have discussed on section 1, some new features are generated as compound of old features, as shown in Table 1. We compare the performance of ANNs that has 54 features and 64 features. The result is shown in Figure 9. The result supports that the added new feature can improve the performance of classification.

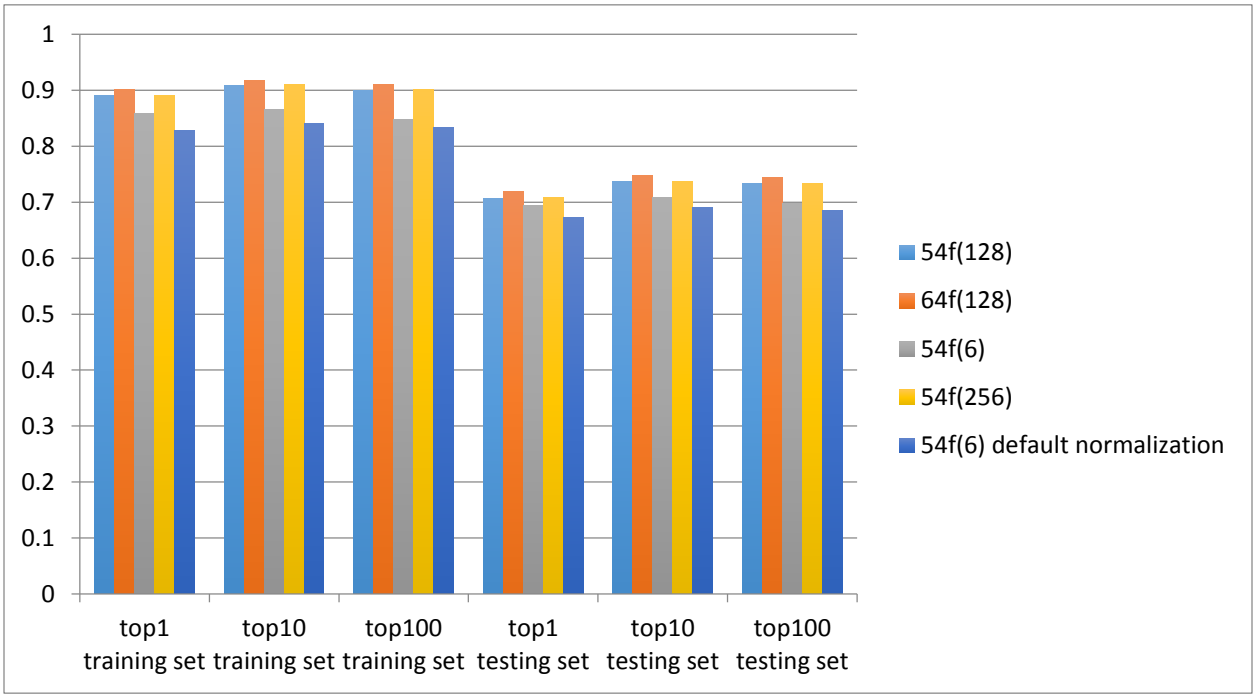


Figure 10: The all experiment results for different parameters.

From figure 10, we can compare precision on training dataset and testing dataset for different parameters. From the comparison, we can get that the best precision is gotten when we use 64 features and 10 neural networks as a group for classification. The precision is 74.73% in the set up. However, we can still improve the performance by considering different number of networks in a group. In the following experiment, I will use the optimal number of networks that we got from training set.

3.4 Experiment for choose the number of ANN classifiers in the group model

As shown in figure 6, we can use different number of ANN in a group. From experiment, we already know that top 10 is better than top 100. To determine the number of ANN in a group that can result a better performance, we should choose appropriate number of ANN. In Figure 11, the experiment for different top N ANNs are applied in evaluation of training dataset. From the result, we can get top 14 gets the best result in top 1 to top 100. Therefore, we also choose top 14 ANNs in a group and evaluate its performance in test dataset. The precision is 0.74819 which is better than the precision from top 10 ANNs model.

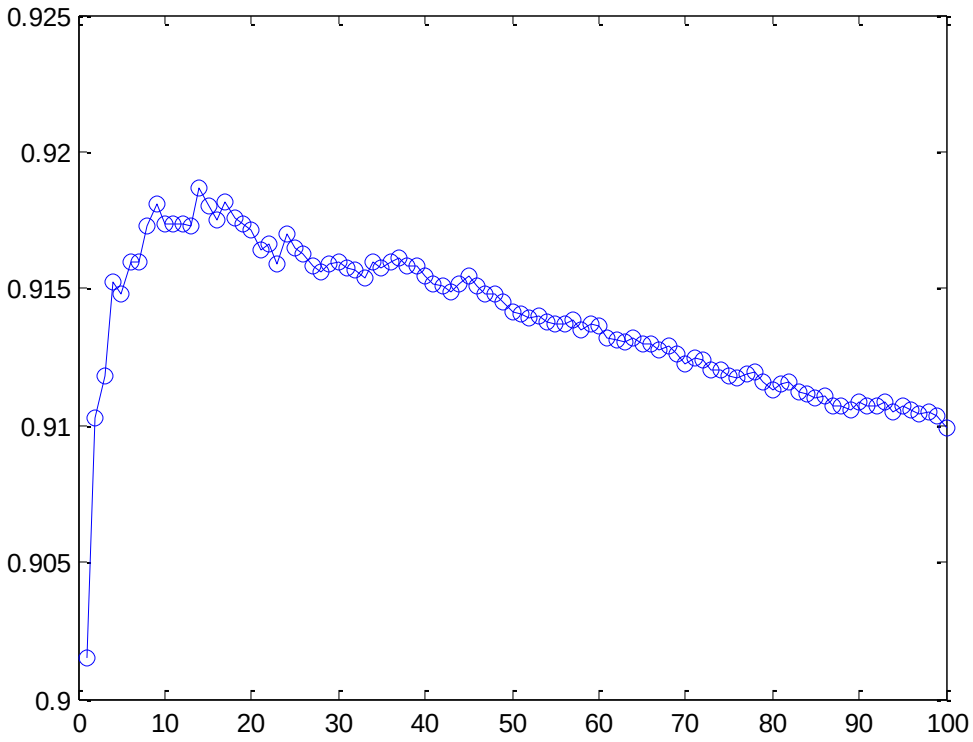


Figure 11: the precision for ANN group model with different number of groups. The ANNs has 64 features as their input. The maximal validation failed value is 128 in the experiment.

4. Conclusion

The result on the leader board of kaggle.com is shown as follows. My proposed method got score as 0.74819, which is also the precision on the testing dataset.

385	↓18	**Wynter Han**	0.74819	6	Fri, 14 Nov 2014 00:43:16 (-7.9d)

I use the ANN model for classification. I also consider some important parameters that may affect the performance. The insight for different parameters in ANN are discussed. Then these insights are applied to improve the performance of classification. The result shows that the improvement is significant when comparing with the basic ANN with default parameter values.

References

[1] Kaggle.com forest cover type prediction, https://www.kaggle.com/c/forest-cover-type-prediction

[2] Blackard, Jock A. and Denis J. Dean. 2000. "Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables." Computers and Electronics in Agriculture 24(3):131-151.

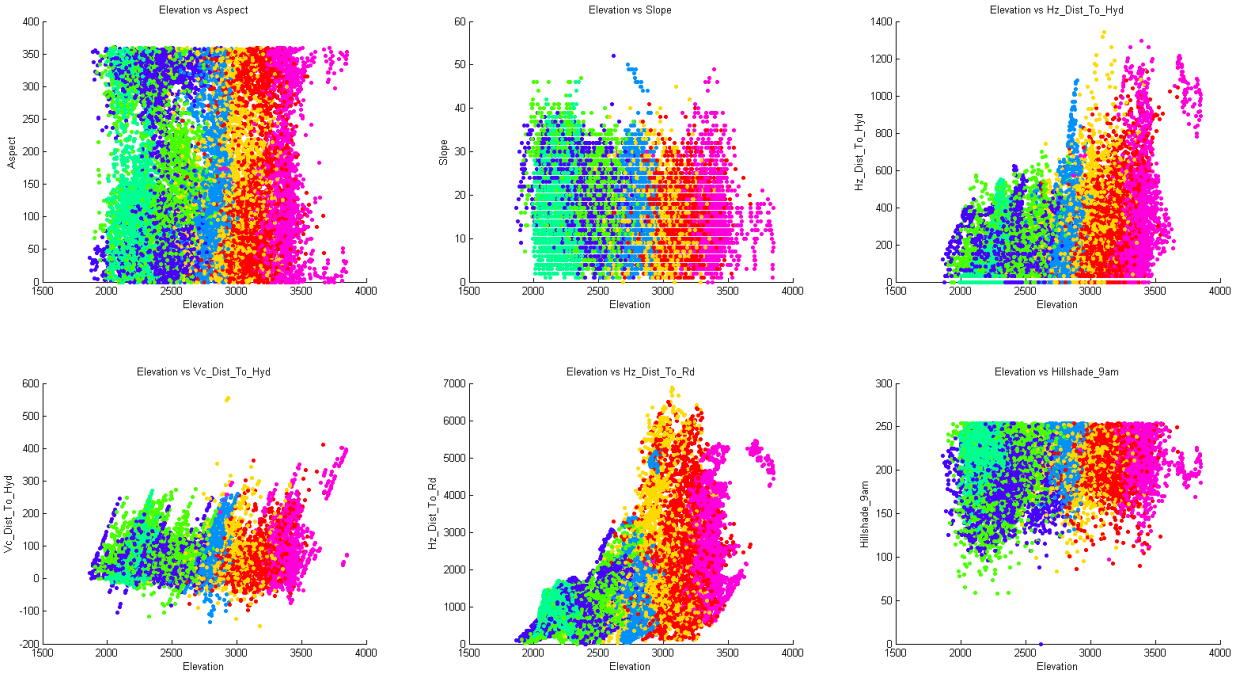
[3] Features engineering http://nbviewer.ipython.org/github/aguschin/kaggle/blob/master/forestCoverType_featuresEngineering.ipynb

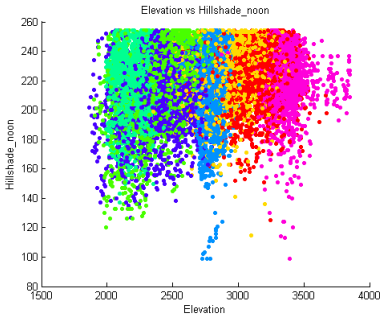
[4] Bishop, Christopher M. "Neural networks for pattern recognition." (1995): 5.

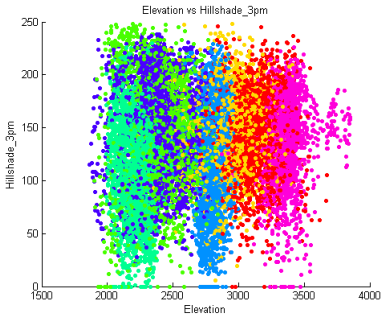
[5] Basu, Jayanta Kumar, Debnath Bhattacharyya, and Tai-hoon Kim. "Use of artificial neural network in pattern recognition." *International journal of software engineering and its applications* 4.2 (2010).

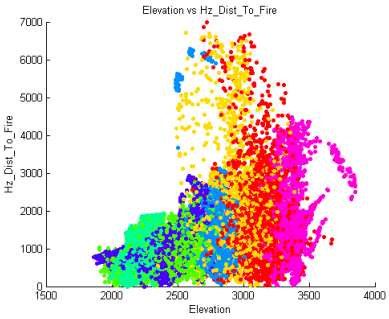
Appendix

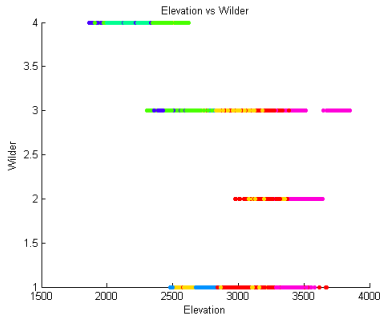
The plots for all pairs of features.

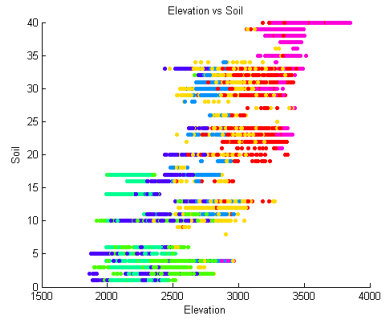


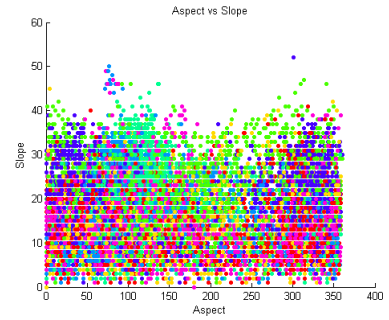

Elevation vs Hillshade_noon
Hillshade_noon
Elevation

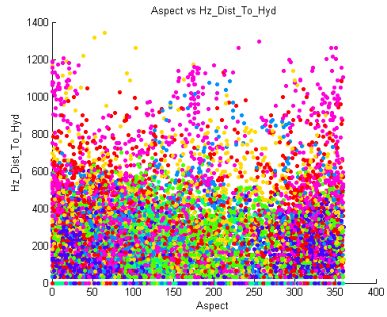

Elevation vs Hillshade_3pm
Hillshade_3pm
Elevation

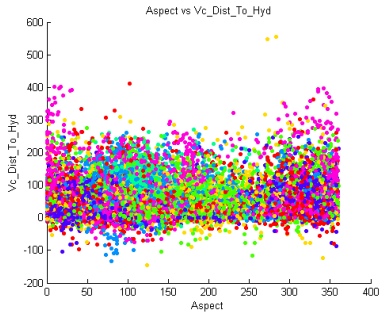

Elevation vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Elevation

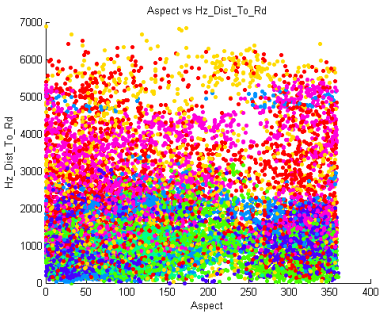

Elevation vs Wilder
Wilder
Elevation

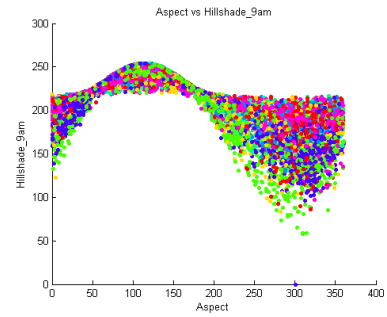

Elevation vs Soil
Soil
Elevation

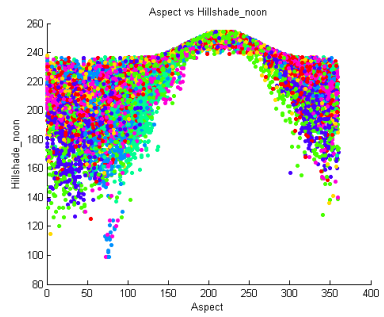

Aspect vs Slope
Slope
Aspect

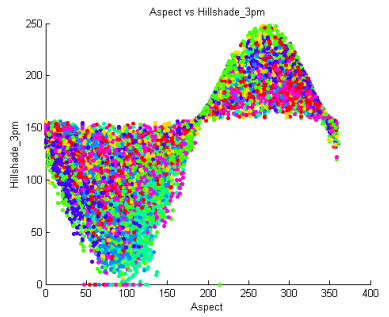

Aspect vs Hz_Dist_To_Hyd
Hz_Dist_To_Hyd
Aspect

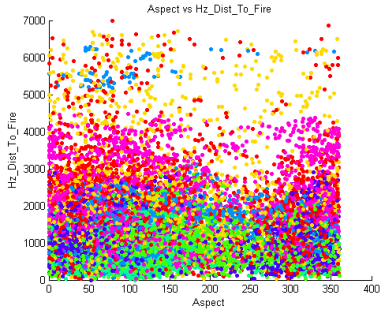

Aspect vs Vc_Dist_To_Hyd
Vc_Dist_To_Hyd
Aspect

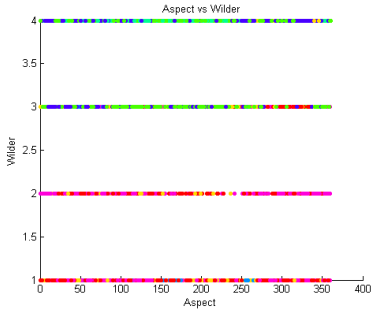

Aspect vs Hz_Dist_To_Rd
Hz_Dist_To_Rd
Aspect

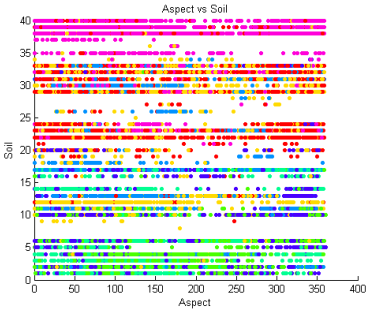

Aspect vs Hillshade_9am
Hillshade_9am
Aspect

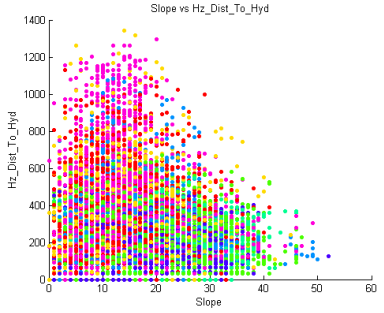

Aspect vs Hillshade_noon
Hillshade_noon
Aspect

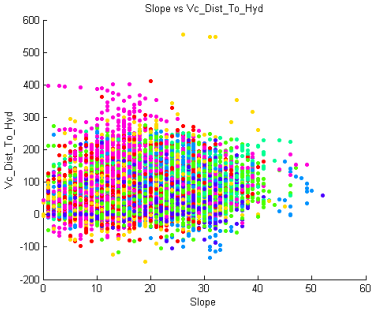

Aspect vs Hillshade_3pm
Hillshade_3pm
Aspect

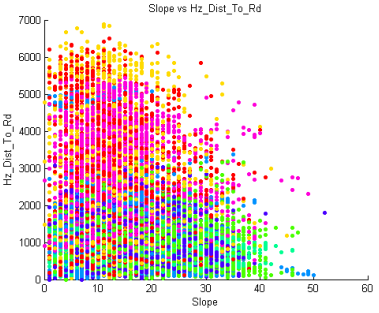

Aspect vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Aspect

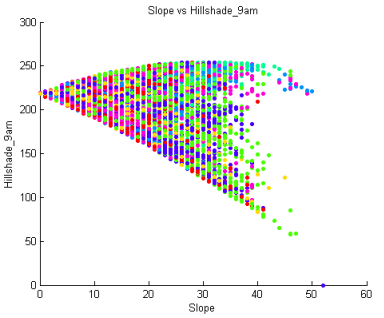

Aspect vs Wilder
Wilder
Aspect

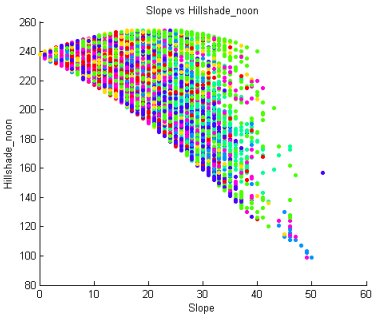

Aspect vs Soil
Soil
Aspect

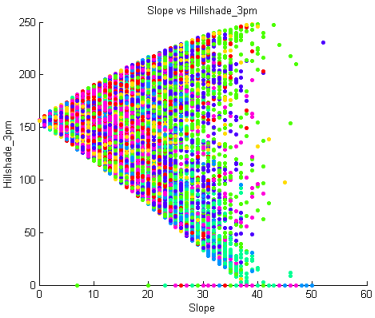

Slope vs Hz_Dist_To_Hyd
Hz_Dist_To_Hyd
Slope

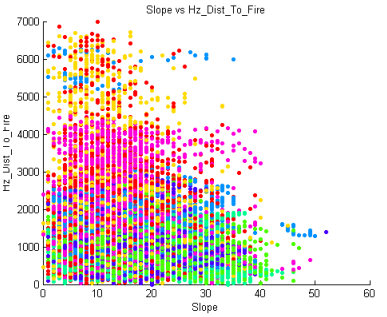

Slope vs Vc_Dist_To_Hyd
Vc_Dist_To_Hyd
Slope

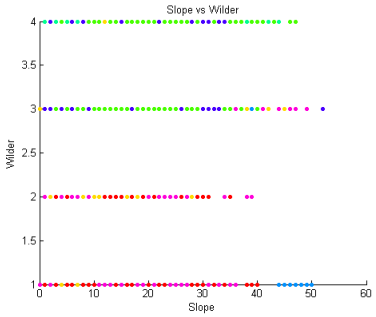

Slope vs Hz_Dist_To_Rd
Hz_Dist_To_Rd
Slope

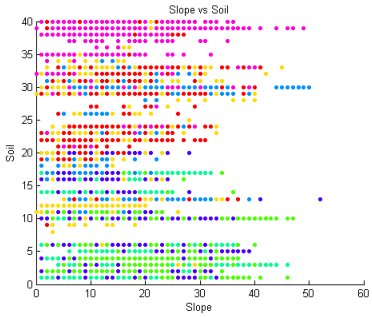

Slope vs Hillshade_9am
Hillshade_9am
Slope

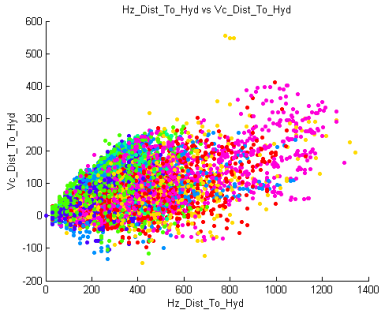

Slope vs Hillshade_noon
Hillshade_noon
Slope

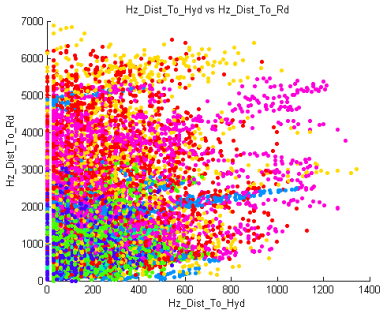

Slope vs Hillshade_3pm
Hillshade_3pm
Slope

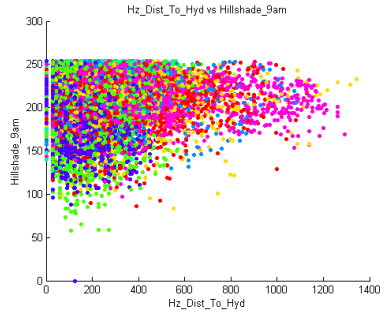

Slope vs Hz_Dist_To_Fire
Slope

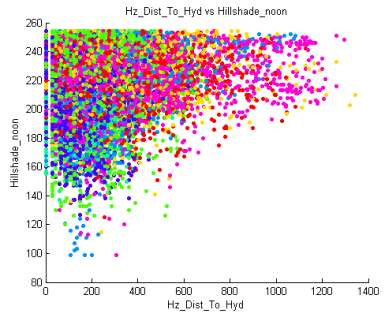

Slope vs Wilder
Wilder
Slope

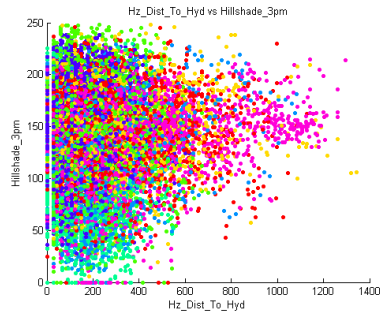

Slope vs Soil
Soil
Slope

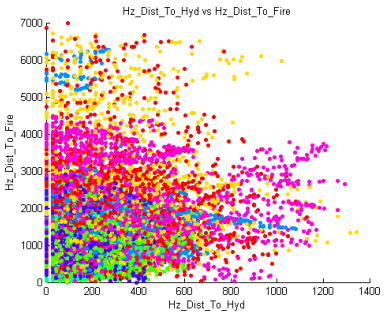

Hz_Dist_To_Hyd vs Vc_Dist_To_Hyd
Vc_Dist_To_Hyd
Hz_Dist_To_Hyd

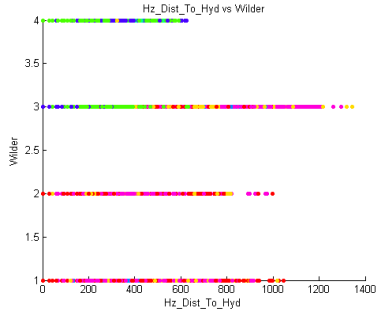

Hz_Dist_To_Hyd vs Hz_Dist_To_Rd
Hz_Dist_To_Rd
Hz_Dist_To_Hyd

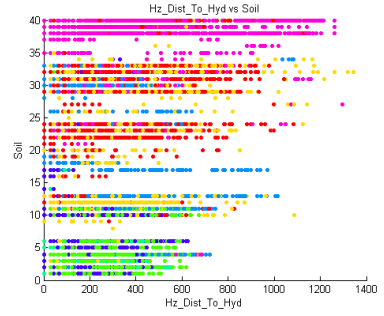

Hz_Dist_To_Hyd vs Hillshade_9am
Hillshade_9am
Hz_Dist_To_Hyd

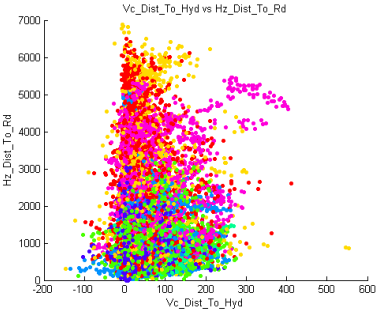

Hz_Dist_To_Hyd vs Hillshade_noon
Hillshade_noon
Hz_Dist_To_Hyd

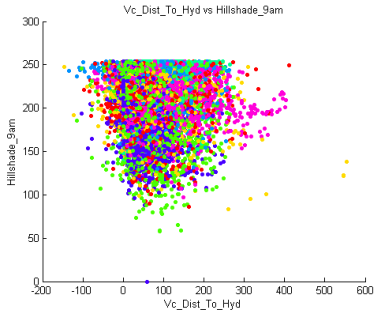

Hz_Dist_To_Hyd vs Hillshade_3pm
Hillshade_3pm
Hz_Dist_To_Hyd

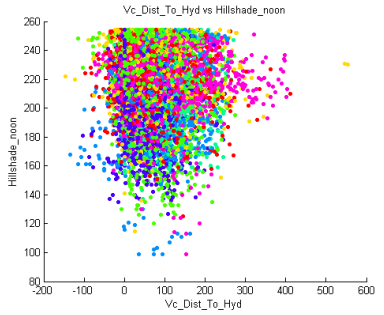

Hz_Dist_To_Hyd vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Hz_Dist_To_Hyd

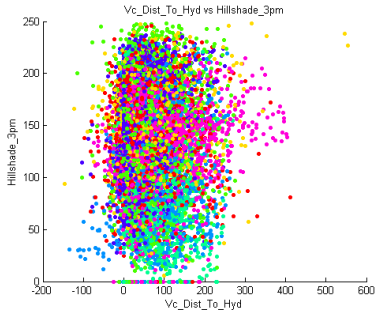

Hz_Dist_To_Hyd vs Wilder
Wilder
Hz_Dist_To_Hyd

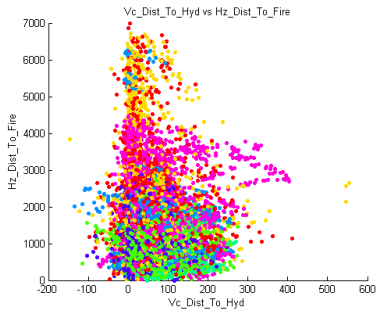

Hz_Dist_To_Hyd vs Soil
Soil
Hz_Dist_To_Hyd

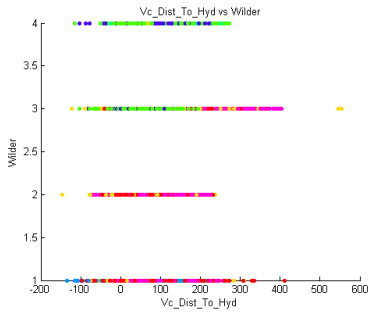

Vc_Dist_To_Hyd vs Hz_Dist_To_Rd
Hz_Dist_To_Rd
Vc_Dist_To_Hyd

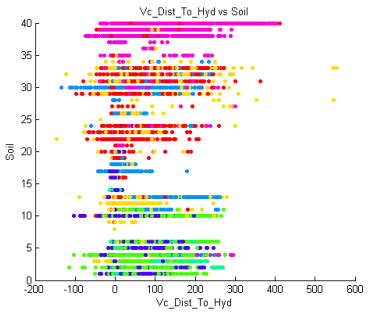

Vc_Dist_To_Hyd vs Hillshade_9am
Hillshade_9am
Vc_Dist_To_Hyd

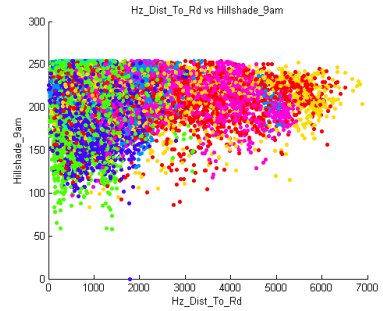

Vc_Dist_To_Hyd vs Hillshade_noon
Hillshade_noon
Vc_Dist_To_Hyd

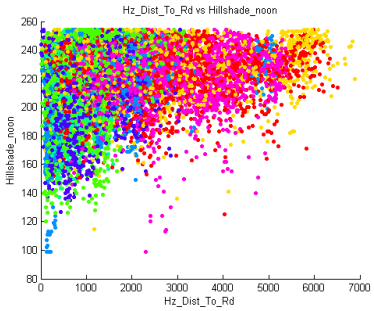

Vc_Dist_To_Hyd vs Hillshade_3pm
Hillshade_3pm
Vc_Dist_To_Hyd

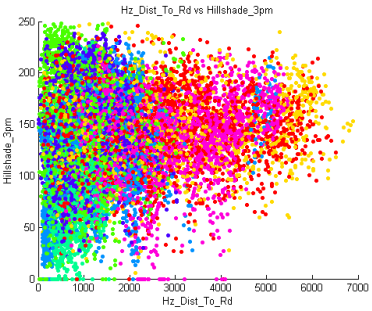

Vc_Dist_To_Hyd vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Vc_Dist_To_Hyd

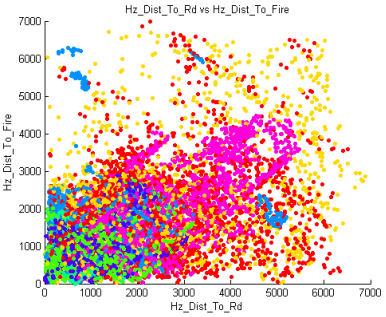

Vc_Dist_To_Hyd vs Wilder
Wilder
Vc_Dist_To_Hyd

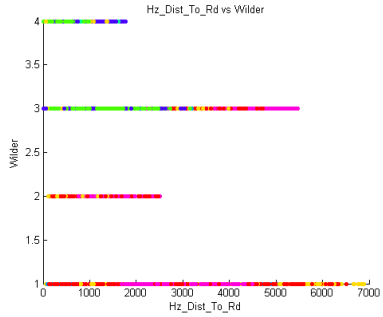

Vc_Dist_To_Hyd vs Soil
Soil
Vc_Dist_To_Hyd

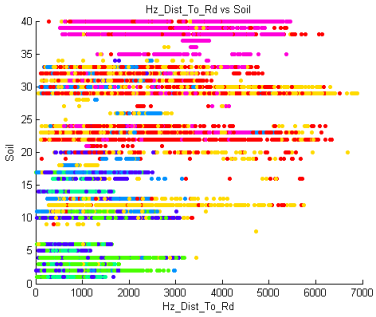

Hz_Dist_To_Rd vs Hillshade_9am
Hillshade_9am
Hz_Dist_To_Rd

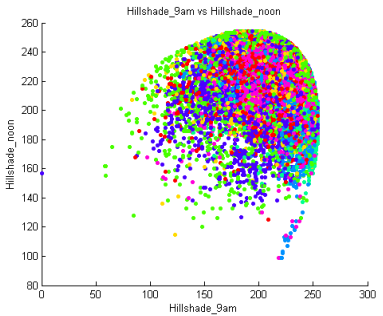

Hz_Dist_To_Rd vs Hillshade_noon
Hillshade_noon
Hz_Dist_To_Rd

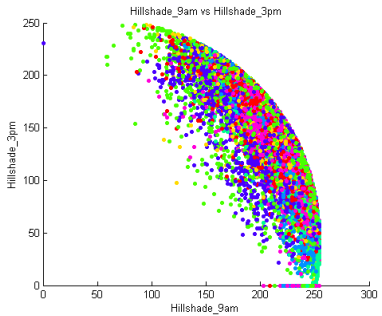

Hz_Dist_To_Rd vs Hillshade_3pm
Hillshade_3pm
Hz_Dist_To_Rd

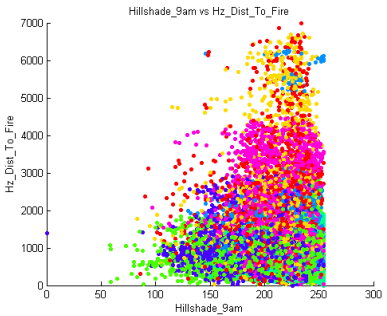

Hz_Dist_To_Rd vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Hz_Dist_To_Rd

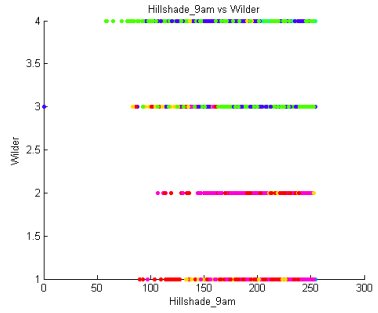

Hz_Dist_To_Rd vs Wilder
Wilder
Hz_Dist_To_Rd

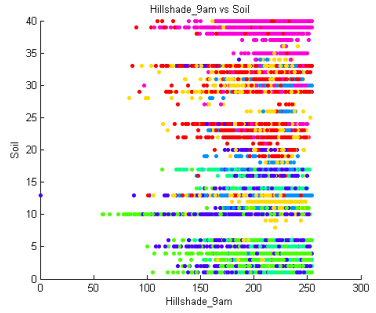

Hz_Dist_To_Rd vs Soil
Soil
Hz_Dist_To_Rd

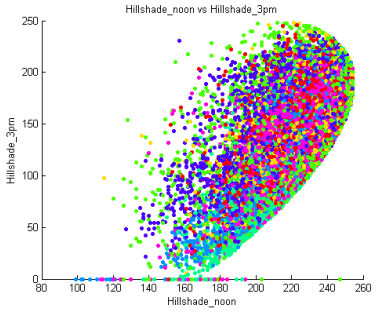

Hillshade_9am vs Hillshade_noon
Hillshade_noon
Hillshade_9am

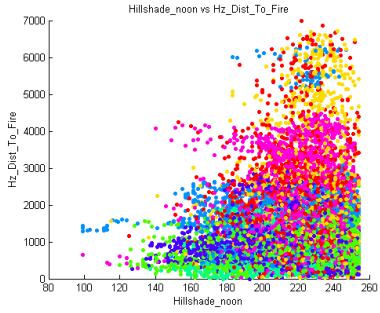

Hillshade_9am vs Hillshade_3pm
Hillshade_3pm
Hillshade_9am

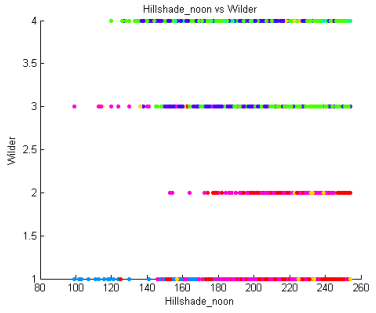

Hillshade_9am vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Hillshade_9am

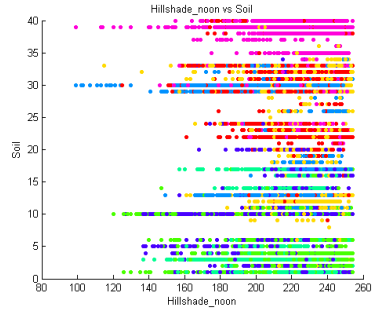

Hillshade_9am vs Wilder
Wilder
Hillshade_9am

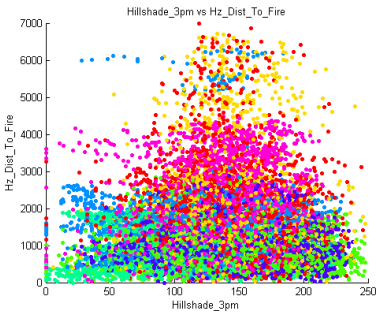

Hillshade_9am vs Soil
Soil
Hillshade_9am

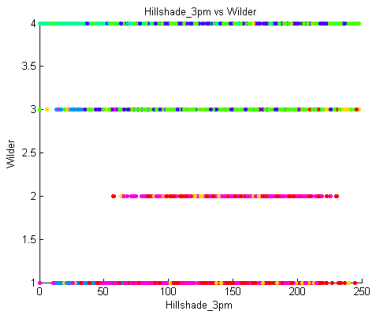

Hillshade_noon vs Hillshade_3pm
Hillshade_3pm
Hillshade_noon

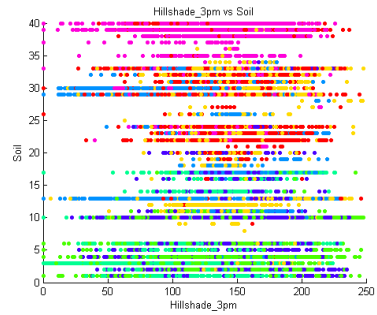

Hillshade_noon vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Hillshade_noon

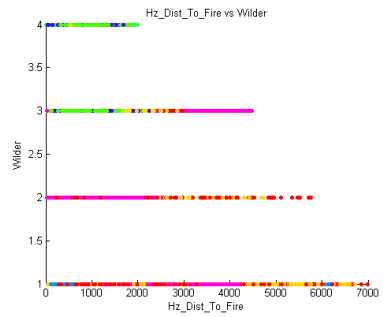

Hillshade_noon vs Wilder
Wilder
Hillshade_noon

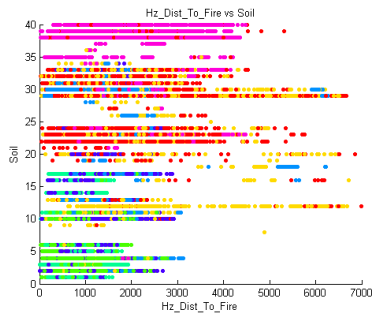

Hillshade_noon vs Soil
Soil
Hillshade_noon

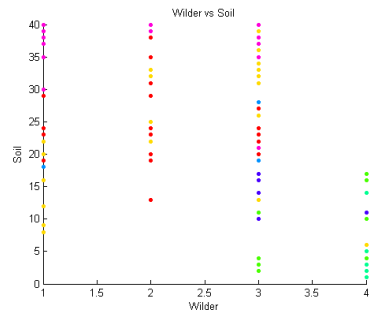

Hillshade_3pm vs Hz_Dist_To_Fire
Hz_Dist_To_Fire
Hillshade_3pm


Hillshade_3pm vs Wilder
Wilder
Hillshade_3pm


Hillshade_3pm vs Soil
Soil
Hillshade_3pm


Hz_Dist_To_Fire vs Wilder
Wilder
Hz_Dist_To_Fire


Hz_Dist_To_Fire vs Soil
Soil
Hz_Dist_To_Fire


Wilder vs Soil
Soil
Wilder