SoftwareMining

Executary Summary

We Employ Understand to get the metrics of our files, By mining the Bugzilla and Github Repo, We semi-mannually label the file with all metrics. Finally, We used a standard machine learning method to train our classifier.

Model

We can build a random forest model using the numerical variables provided.

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
options (warn=-1)
setwd('E:/Project/feature')
rawdata <- read.csv('ProcessedData.csv', header = TRUE, sep = ',')

data <- rawdata;

for(i in c(2:ncol(rawdata)-1)) {data[,i] = as.numeric(as.character(rawdata[,i]))}

featuresnames <- colnames(data)[-(1:1)]
features <- data[featuresnames]
```

```
set.seed(5188)
xdata <- createDataPartition(y=features$classe, p=3/4, list=FALSE )
training <- features[xdata,]
testing <- features[-xdata,]

rf_model  <- randomForest(classe ~ ., training, ntree=500, mtry=32)
```

Cross Validation

We are able to measure the accuracy using our training set and our cross validation set. With the training set we can detect if our model has bias due to ridgity of our mode. With the cross validation set, we are able to determine if we have variance due to overfitting.

In-sample accuracy

```
training_pred <- predict(rf_model, training)
print(confusionMatrix(training_pred, training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction false true
##      false  1895    0
##      true      0   58
##
##                Accuracy : 1
##                  95% CI : (0.9981, 1)
##     No Information Rate : 0.9703
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9703
##          Detection Rate : 0.9703
##    Detection Prevalence : 0.9703
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : false
##
```

Out-of-sample accuracy

```
testing_pred <- predict(rf_model, testing)
print(confusionMatrix(testing_pred, testing$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction false true
##      false   630   12
##      true      1    7
##
##                Accuracy : 0.98
##                  95% CI : (0.966, 0.9893)
##     No Information Rate : 0.9708
##     P-Value [Acc > NIR] : 0.095058
##
##                   Kappa : 0.51
##  Mcnemar's Test P-Value : 0.005546
##
##             Sensitivity : 0.9984
##             Specificity : 0.3684
##          Pos Pred Value : 0.9813
##          Neg Pred Value : 0.8750
##              Prevalence : 0.9708
##          Detection Rate : 0.9692
##    Detection Prevalence : 0.9877
##       Balanced Accuracy : 0.6834
##
##        'Positive' Class : false
##
```