```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="generator" content="pandoc" />

<meta name="author" content="Onur Akpolat" />


<title>Predict activity quality from activity monitors</title>

<script src="data:application/x-javascript,%2F%2A%21%20jQuery%20v1%2E11%2E0%20%7C%20%28c%29%202005%2C%202014%20jQuery%20Foundation%2C%20Inc%2E%20%7C
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link href="data:text/css,%2F%2A%21%0A%20%2A%20Bootstrap%20v2%2E3%2E2%0A%20%2A%0A%20%2A%20Copyright%202013%20Twitter%2C%20Inc%0A%20%2A%20Licensed%2
<link href="data:text/css,%2F%2A%21%0A%20%2A%20Bootstrap%20Responsive%20v2%2E3%2E2%0A%20%2A%0A%20%2A%20Copyright%202013%20Twitter%2C%20Inc%0A%20%2A%
<script src="data:application/x-javascript,%2F%2A%21%0A%2A%20Bootstrap%2Ejs%20by%20%40fat%20%26%20%40mdo%0A%2A%20Copyright%202013%20Twitter%2C%20Inc

<style type="text/css">code{white-space: pre;}</style>
<link href="data:text/css,pre%20%2Eoperator%2C%0Apre%20%2Eparen%20%7B%0A%20color%3A%20rgb%28104%2C%20118%2C%20135%29%0A%7D%0A%0Apre%20%2Eliteral%20%
<script src="data:application/x-javascript,%0Avar%20hljs%3Dnew%20function%28%29%7Bfunction%20m%28p%29%7Breturn%20p%2Ereplace%28%2F%26%2Fgm%2C%22%26a
<style type="text/css">
  pre:not([class]) {
    background-color: white;
  }
</style>
<script type="text/javascript">
if (window.hljs && document.readyState && document.readyState === "complete") {
   window.setTimeout(function() {
      hljs.initHighlighting();
   }, 0);
}
</script>



</head>

<body>

<style type="text/css">
.main-container {
  max-width: 940px;
  margin-left: auto;
  margin-right: auto;
}
</style>
<div class="container-fluid main-container">


<div id="header">
<h1 class="title">Predict activity quality from activity monitors</h1>
<h4 class="author"><em>Onur Akpolat</em></h4>
<h4 class="date"><em><ol start="24" style="list-style-type: decimal">
<li>January 2015</li>
</ol></em></h4>
</div>


<div id="synopsis" class="section level2">
<h2>Synopsis</h2>
<p>Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relative
<p>The goal of this project is to predict the manner in which they did the exercise. This is the <code>classe</code> variable in the training set.</
</div>
<div id="data-description" class="section level2">
<h2>Data description</h2>
<p>The outcome variable is <code>classe</code>, a factor variable with 5 levels. For this data set, participants were asked to perform one set of 10
<ul>
<li>exactly according to the specification (Class A)</li>
<li>throwing the elbows to the front (Class B)</li>
```

```
<li>lifting the dumbbell only halfway (Class C)</li>
<li>lowering the dumbbell only halfway (Class D)</li>
<li>throwing the hips to the front (Class E)</li>
</ul>
</div>
<div id="initial-configuration" class="section level2">
<h2>Initial configuration</h2>
<p>The initial configuration consists of loading some required packages and initializing some variables.</p>
<pre class="r"><code>#Data variables
training.file   &lt;- './data/pml-training.csv'
test.cases.file &lt;- './data/pml-testing.csv'
training.url    &lt;- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
test.cases.url  &lt;- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'

#Directories
if (!file.exists(&quot;data&quot;)){
  dir.create(&quot;data&quot;)
}
if (!file.exists(&quot;data/submission&quot;)){
  dir.create(&quot;data/submission&quot;)
}

#R-Packages
IscaretInstalled &lt;- require(&quot;caret&quot;)</code></pre>
<pre><code>## Loading required package: caret
## Loading required package: lattice
## Loading required package: ggplot2</code></pre>
<pre class="r"><code>if(!IscaretInstalled){
    install.packages(&quot;caret&quot;)
    library(&quot;caret&quot;)
    }

IsrandomForestInstalled &lt;- require(&quot;randomForest&quot;)</code></pre>
<pre><code>## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.</code></pre>
<pre class="r"><code>if(!IsrandomForestInstalled){
    install.packages(&quot;randomForest&quot;)
    library(&quot;randomForest&quot;)
    }

IsRpartInstalled &lt;- require(&quot;rpart&quot;)</code></pre>
<pre><code>## Loading required package: rpart</code></pre>
<pre class="r"><code>if(!IsRpartInstalled){
    install.packages(&quot;rpart&quot;)
    library(&quot;rpart&quot;)
    }

IsRpartPlotInstalled &lt;- require(&quot;rpart.plot&quot;)</code></pre>
<pre><code>## Loading required package: rpart.plot</code></pre>
<pre class="r"><code>if(!IsRpartPlotInstalled){
    install.packages(&quot;rpart.plot&quot;)
    library(&quot;rpart.plot&quot;)
    }

# Set seed for reproducability
set.seed(9999)</code></pre>
</div>
<div id="data-processing" class="section level2">
<h2>Data processing</h2>
<p>In this section the data is downloaded and processed. Some basic transformations and cleanup will be performed, so that <code>NA</code> values ar
<p>The <code>pml-training.csv</code> data is used to devise training and testing sets. The <code>pml-test.csv</code> data is used to predict and ans
<pre class="r"><code># Download data
download.file(training.url, training.file)
download.file(test.cases.url,test.cases.file )

# Clean data
training   &lt;-read.csv(training.file, na.strings=c(&quot;NA&quot;,&quot;#DIV/0!&quot;, &quot;&quot;))
testing &lt;-read.csv(test.cases.file , na.strings=c(&quot;NA&quot;, &quot;#DIV/0!&quot;, &quot;&quot;))
training&lt;-training[,colSums(is.na(training)) == 0]
testing &lt;-testing[,colSums(is.na(testing)) == 0]

# Subset data
```

```
training   &lt;-training[,-c(1:7)]
testing &lt;-testing[,-c(1:7)]</code></pre>
</div>
<div id="cross-validation" class="section level2">
<h2>Cross-validation</h2>
<p>In this section cross-validation will be performed by splitting the training data in training (75%) and testing (25%) data.</p>
<pre class="r"><code>subSamples &lt;- createDataPartition(y=training$classe, p=0.75, list=FALSE)
subTraining &lt;- training[subSamples, ]
subTesting &lt;- training[-subSamples, ]</code></pre>
</div>
<div id="expected-out-of-sample-error" class="section level2">
<h2>Expected out-of-sample error</h2>
<p>The expected out-of-sample error will correspond to the quantity: 1-accuracy in the cross-validation data. Accuracy is the proportion of correct
</div>
<div id="exploratory-analysis" class="section level2">
<h2>Exploratory analysis</h2>
<p>The variable <code>classe</code> contains 5 levels. The plot of the outcome variable shows the frequency of each levels in the subTraining data.
<pre class="r"><code>plot(subTraining$classe, col=&quot;orange&quot;, main=&quot;Levels of the variable classe&quot;, xlab=&quot;classe levels&quot;
<p><img src="
<p>The plot above shows that Level A is the most frequent classe. D appears to be the least frequent one.</p>
</div>
<div id="prediction-models" class="section level2">
<h2>Prediction models</h2>
<p>In this section a decision tree and random forest will be applied to the data.</p>
<div id="decision-tree" class="section level3">
<h3>Decision tree</h3>
<pre class="r"><code># Fit model
modFitDT &lt;- rpart(classe ~ ., data=subTraining, method=&quot;class&quot;)

# Perform prediction
predictDT &lt;- predict(modFitDT, subTesting, type = &quot;class&quot;)

# Plot result
rpart.plot(modFitDT, main=&quot;Classification Tree&quot;, extra=102, under=TRUE, faclen=0)</code></pre>
<p><img src="
<p>Following confusion matrix shows the errors of the prediction algorithm.</p>
<pre class="r"><code>confusionMatrix(predictDT, subTesting$classe)</code></pre>
<pre><code>## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1266  208   25   91   29
##          B   33  535   71   30   67
##          C   28   90  676  130   94
##          D   45   72   59  501   43
##          E   23   44   24   52  668
##
## Overall Statistics
##
##                Accuracy : 0.7435
##                  95% CI : (0.731, 0.7557)
##     No Information Rate : 0.2845
##     P-Value [Acc &gt; NIR] : &lt; 2.2e-16
##
##                   Kappa : 0.6738
##  Mcnemar's Test P-Value : &lt; 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9075   0.5638   0.7906   0.6231   0.7414
## Specificity            0.8994   0.9492   0.9155   0.9466   0.9643
## Pos Pred Value         0.7820   0.7269   0.6640   0.6958   0.8237
## Neg Pred Value         0.9607   0.9007   0.9539   0.9276   0.9431
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2582   0.1091   0.1378   0.1022   0.1362
## Detection Prevalence   0.3301   0.1501   0.2076   0.1468   0.1654
## Balanced Accuracy      0.9035   0.7565   0.8531   0.7849   0.8528</code></pre>
</div>
<div id="random-forest" class="section level3">
<h3>Random forest</h3>
<pre class="r"><code># Fit model
modFitRF &lt;- randomForest(classe ~ ., data=subTraining, method=&quot;class&quot;)
```

```
# Perform prediction
predictRF &lt;- predict(modFitRF, subTesting, type = &quot;class&quot;)</code></pre>
<p>Following confusion matrix shows the errors of the prediction algorithm.</p>
<pre class="r"><code>confusionMatrix(predictRF, subTesting$classe)</code></pre>
<pre><code>## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    2    0    0    0
##          B    1  946    8    0    0
##          C    0    1  846    6    0
##          D    0    0    1  796    1
##          E    0    0    0    2  900
##
## Overall Statistics
##
##                Accuracy : 0.9955
##                  95% CI : (0.9932, 0.9972)
##     No Information Rate : 0.2845
##     P-Value [Acc &gt; NIR] : &lt; 2.2e-16
##
##                   Kappa : 0.9943
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   0.9968   0.9895   0.9900   0.9989
## Specificity            0.9994   0.9977   0.9983   0.9995   0.9995
## Pos Pred Value         0.9986   0.9906   0.9918   0.9975   0.9978
## Neg Pred Value         0.9997   0.9992   0.9978   0.9981   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1929   0.1725   0.1623   0.1835
## Detection Prevalence   0.2847   0.1947   0.1739   0.1627   0.1839
## Balanced Accuracy      0.9994   0.9973   0.9939   0.9948   0.9992</code></pre>
</div>
</div>
<div id="conclusion" class="section level2">
<h2>Conclusion</h2>
<div id="result" class="section level3">
<h3>Result</h3>
<p>The confusion matrices show, that the Random Forest algorithm performens better than decision trees. The accuracy for the Random Forest model was
</div>
<div id="expected-out-of-sample-error-1" class="section level3">
<h3>Expected out-of-sample error</h3>
<p>The expected out-of-sample error is estimated at 0.005, or 0.5%. The expected out-of-sample error is calculated as 1 - accuracy for predictions m
</div>
</div>
<div id="submission" class="section level2">
<h2>Submission</h2>
<p>In this section the files for the project submission are generated using the random forest algorithm on the testing data.</p>
<pre class="r"><code># Perform prediction
predictSubmission &lt;- predict(modFitRF, testing, type=&quot;class&quot;)
predictSubmission</code></pre>
<pre><code>##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E</code></pre>
<pre class="r"><code># Write files for submission
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0(&quot;./data/submission/problem_id_&quot;,i,&quot;.txt&quot;)
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictSubmission)</code></pre>
</div>


</div>
```

```
<script>

// add bootstrap table styles to pandoc tables
$(document).ready(function () {
  $('tr.header').parent('thead').parent('table').addClass('table table-condensed');
});

</script>

<!-- dynamically load mathjax for compatibility with self-contained -->
<script>
  (function () {
    var script = document.createElement("script");
    script.type = "text/javascript";
    script.src  = "https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML";
    document.getElementsByTagName("head")[0].appendChild(script);
  })();
</script>

</body>
</html>
```