# Distracted Driving Project Outline

*Joseph Blubaugh*

*12 December 2016*

## Contents

**Data Extraction, Preparation, and Project Management** **2**
- Data Overview . . . 2
- Data Extraction with Python . . . 2
- Data Preparation in R . . . 2
- Project Management: Reproducible Research and Version Control . . . 3

**Exploratory Analysis and Model Proposal** **3**
- Exploring the Texting Simulation . . . 3
- Model Proposal: Detecting Texting Events through Facial Expressions . . . 4

**Exploratory Modeling** **4**
- Establishing a Baseline . . . 4
- Subject Independent Macro Level Model . . . 4

**Understanding Basic Neural Networks** **6**
- Nodes, Weights and the Activation Function . . . 6
- Training a Neural Network . . . 6

**Model Selection** **7**
- Training/Testing Strategy and General Model Form . . . 7
- Model Fitting with the Caret Package . . . 7
- Model Performance . . . 7
- Time Omission . . . 8

**Best Model Analysis** **9**
- Interpreting Weights . . . 9
- Variable Importance . . . 9
- Texting Event Probability Conditional on Emotion . . . 10
- Demographic Effects . . . 11

## Data Extraction, Preparation, and Project Management

### Data Overview

The data in this project are of 8 driving simulations for 66 individuals ranging from 3,000 to 30,000 observations per simulation. There are over 6.7 million observations in the entire dataset. The data from each simulation includes likelihood scores for 8 facial expressions recorded at a fixed interval of .03 seconds. Stimuli data which records targetted events introduced into each simulation and basic demographic data on each subject are also available. The data was originally stored in 777 data files.

### Data Extraction with Python

- **Simulation Files**: In order to prepare the data for analysis each of the 509 simulation files (.xlsx) needed to be extracted and combined into a single file to make it more usable. I wrote a python program to extract data from each simulation file which for the most part had identical layouts. Two of the simulation files did not start on the same row and had to be corrected manually. Each simulation consisted of the Frame, Time and the 8 Emotional Likelihoods. The name of each file identified the subject and the trial which I extracted and combined with the simulation data so the subject and trial could be identified in the analysis.
- **Event Files**: There were 267 stimuli files (.stm) that recorded the events that occured in each simulation. They had identical layouts and were able to be extracted in the same manner as the simulation data. Python was also used to extract and combine the data.
- **Demographics**: A single file that recorded the age (young/old) and gender (male/female). No special processing was required.

### Data Preparation in R

- **Merging**: Merging the event data with the simulation data was not straight forward. The event data captured the starting and ending times of events only so in order to combine the two datasets I had to write a loop function to go through each record and compare time between the simulation and the starting/ending time of the event. For any matching subject and trial, if the simulation time fell within the starting/ending time interval in the event data, then all of the records in the simulation during that time interval were coded with that event, all observations outside of the event time interval were coded as No Events.
  - *additional note*: We dont know exactly what occurred during the time interval so I made the assumption that all of the time within the event interval should be coded as an event. For instance, we dont know the sequence of events that occurred during the interval which lasted 2.5 minutes on average. Was it one long texting action or was it a series sending, waiting, receiving actions?
- **Data Issues**: There were some data quality issues that needed to be addressed during the data preparation process. In additional to the two simulation files with differing

formats, there were mispellings and inconsistent labels in the event descriptions. Luckily there was a corresponding coded variable for each event description. I recoded all of the event descriptions based on the highest frequency description for each coded variable. This reduced the number of unique event descriptions from 20 to 6 which was consistent with the event coded variable. When creating a frequency table of Events to Trial I discovered a possible miss coding for a single subject in Trial 6. Trial 006 has 62 out of 63 trials where the only event involves emotional questions, however one trial shows as having mathematical question events. This appears to be a data error based on the experimental design where mathematical and anlaytical questions are only supposed to occur in trial 005. The subject who this possible misscoding is associated with is T018-006.

### Project Management: Reproducible Research and Version Control

- I believe it is responsible to have reproducible work. All of my code and documents are organized in a git repository hosted on github. The main page has an introductory document to the project and explains the steps someone would need to take to reproduce the data, models, and results. The data itself is too large to be hosted on github and will need to be retrieved elsewhere. (www.github.com/jestonblu/driving)

## Exploratory Analysis and Model Proposal

### Exploring the Texting Simulation

Initial exploration of the data beyond the data preparation section included creating plots by emotion and event to see if there was a relationship between texting events and facial expressions that could be represented visually. This was accomplished primarily through two plots.

- **Subject Specific Plot**: A subject specific plot that overlayed Trial 004 (No Event Simulation which was later used as a baseline measurement) with Trial 007 (the texting event trial). For each of the 8 emotions, each observations of both trials were plotted and colored based on an event indicator. Since Trial 004 had no events, all were colored gray. Trial 007 had 2 intervals of texting events which were colored in yellow, the observations outside of the texting event intervals were assigned the same gray color as Trial 004. For each emotion and trial, a LOESS (Local Polynomial Regression) curve was added to the plot which showed the moving average of the recent observations.
- **Plot of all Subjects**: A faceted plot of LOESS curves for each emotion by subject and trial. The first row contained 59 LOESS curves for Trial 004 (no event trial). The second row diplayed the same curves for Trial 007 (texting trial).
- **Takeaways**:

    1) Many subjects displayed visual differences between the two trials during texting events indicating potential to detect texting from facial expressions.
    2) Anger, Contempt, Disgust, and Neutral displayed a lot of variation between both trials and Trial 004 showed less variation than Trial 007 on average.

3) Fear, Joy, Sad, and Surprise displayed very little variation and for the most part appeared to remain constant during both trials.
4) Anger, Contempt, Disgust, Sad, and Neutral all showed more variation in the texting trial (007) than the no event trial (004).
5) The increased variation in emotions indicates that there may be an opportunity to aggregate the data and use summary statistics as predictor variables for whether a texting even is taking place.

### Model Proposal: Detecting Texting Events through Facial Expressions

I proposed that this analysis focus on building a model to detect whether the subject is texting based on demographics and changes in facial expressions. I initially wanted to leave in the time series component and experiment with complex models such as neural nets. I also proposed to measure the change in probability of texting events when facial expressions change. Lastly I wanted to understand if one model performs reasonably well on all or most subjects or would a subject factor variable would be required.

## Exploratory Modeling

### Establishing a Baseline

This analysis focussed on trying to detect texting events from subjects participating in the driving simulation trial 007. In the distracted driving experiment, trial 004 was used as a practice run where the subject drove approximately 11 kilometers on a crowded four lane road with a speed limit of 70 kph. There was a period in the drive where the driver had to take a short detour, but otherwise the drive was a straight line. No events occurred in this simulation so I elected to use this trial as a baseline to compare against trial 007 where the driver encountered texting events on the same road with the same conditions. In order to establish a baseline the overall mean was computed for each subject's emotional likelihood over the entire simulation. The mean was subtracted from each observation in trial 007 to calculate the relative position vs the baseline trial average. An observation value of zero indicates a value equal to the computed mean in the baseline trial.

### Subject Independent Macro Level Model

- **Data Aggregation**: In 2012 Ford sponsored a competition to develop a model that could successfully detect whether a driver was paying attention (https://www.kaggle.com/c/stayalert). The data was similar to this project and the winner of the Ford competition wrote that instead of approaching the analysis as a time series problem, he instead averaged all values over the entire trial so that he had a single observation per trial. I have elected to try a similar approarch first. I computed the min, max, mean, var, and iqr for every simulation, grouped by the event. The aggegated data consisted of 119 observations and 40 descriptive statistics. The data were split by subject, but the subject variable was not retained. The observations were split into a training and testing set with a 60/40 ratio. A neural net and random forest model were both

applied to the training set with default parameters. An overall quality measure of (Sensitivity + Specificity) / 2 was used to evaluate each model.

- **Random Forest**: Random Forests are well suited for data sets where the number of variables is comparable to the number of observations. The basic structure of a random forest is to create many regression trees where the predictor variables are chosen randomly. Each tree votes on the the outcome and the majority of votes determins the output of the model.
- **Neural Net**: Neural Networks are made up of a series of interconnected nodes and weights that iterate on training data to update weights and minimize training errors. There are several types of neural nets and the type that I have used so far is the feed forward neural net, meaning that data is fed into the model through the input nodes, through the hidden layers, to the output nodes.
- **Assessment of Fit**: A confusion matrix was produced for both models. Its a simple table which compares the predicted values to the actual values. In this case, whether a texting even had occurred. To assess the overall fit the overall number of correct predictions was compared to the incorrect predictions. Both models had perfect fits (100% correct) on the training data, but the testing performance was poor (random forest: 60%, neural net: 58%).
- **Conclusion**: The first pass analysis used summarized statistical measures to fit both a random forest and neural network model to the data. The results primarily showed that aggregating the data over an entire event prevented the two algorithms from detecting texting events. Also, because the data was aggregated, there were very few data points to model between testing and training sets. The random forest algorithm was a better choice than the neural network because of the large number of predictor variables relative to the number of observations. Both models however, were guilty of overfitting. For the next phase of the analysis I have chosen to go back to the original data and only do minor aggregation. I have chosen to focus on the neural network rather than the random forest mainly because of this data structure. Neural networks are well suited for large data sets of continuous variables.
- **Takeaways**:

    1) The summary statistics calculated at each event does not provide a good enough foundation to adequately detect event changes. The restrictive size of the aggregated dataset makes overfitting a challenge.
    2) There is evidence that a one size fits all model will not work for this problem given, Subject specific data should be used.
    3) Variable Importance Plot of the Random Forest model indicates that the Neutral Emotion may be the more important indicator of texting events. Of the top 5 most important variables, 3 of them were computed from the Neutral emotion (mean, variance, iqr).

## Understanding Basic Neural Networks

The feed-forward neural net consists of two basic components: nodes and weights. Nodes represent the entry, exit, and stopping points in the network flow. The weights represent the transformation that takes place enroute between nodes. In order for me to better understand the mechanics of the neural net model, I spent some time to diagram and calculate by hand a single iteration of a basic neural network model.

### Nodes, Weights and the Activation Function

- **Nodes**: There are three types of nodes: Input nodes (predictor variables), Output nodes (response variables), and hidden nodes. The number of hidden nodes represents the complexity of a neural net model. A model with no hidden nodes would be equivalent to a logistic regression model where the weights between the input and the output nodes represent the regression coefficients.
- **Weights**: Weights are initially chosen randomly so neural nets have an iterative process to minimize error by feeding the training error back through the model. The change for each iteration is calculated by multiplying the first derivative of the activation function by the most recent training error. In the case of a neural net with 2 inputs, 1 output, and 3 hidden nodes, the weight values exist between each input and hidden node and each hidden and output node.
- **Activation Function**: Neural networks require an activation function to transform the input into the appropriate output. In the case of this analysis I am trying to predict texting events so I only want predictions between 0 and 1. An appropriate activation function for this is sigmoid function: `1 / (1 + exp(-x))`.

### Training a Neural Network

Training a neural network in practice does not require an advanced knowledge of the underlying mechanics. While the underlying ideas of neural nets are rooted in regression, the nature of the many interconnected relationships in the network make extracting and doing inference on the components very difficult. As a result, most model diagnostics center around comparing the training and testing set performances and searching for the optimal combination of model parameters. When training a feed-forward neural network in R (nnet package) you basically have the following parameters to change:

- **Size**: The number of hidden nodes. Directly effects number of weights and training time.
- **Decay**: A penalty that is applied to weights that do not update after an iteration, similar to ridge regression.
- **Max Iterations**: The number of training cycles. Used to prevent overfitting.
- **Max Weights**: Total number of weights allowed. Used to reduce training time. (Input Nodes * Hidden Nodes) + (1 Bias Correction * Hidden Nodes) + (Hidden Nodes * Output Node) + (1 Output Bias Correction * Output Node) = Total Weights.

## Model Selection

### Training/Testing Strategy and General Model Form

- **General Model Form**: nnet(Texting ~ Subject + Age + Gender + Anger + Contempt + Disgust + Fear + Joy + Sad + Surprise + Neutral)

The general strategy for selecting a model is to use the same model form on various cuts of the data. I have selected 6 different data processing methods and 2 data split methods to train. In total there are 12 models that are fit. The best model is selected based on total accuracy of the testing set.

- **Data Processing**:
  - Original: Data in its original form. Variables are the 8 emotions measured at .03 second intervals.
  - Differencing: First order differening of each original observation.
  - Moving Avg: Running averages n=30 for all of the emotions.
  - 1/2 Sec Cut: Time is cut into half second intervals with the average value of each emotion recorded over the interval.
  - 1/2 Sec Diff: First order differncing of the 1/2 second cut data.
  - 1/2 Sec Cut Stat: In addition to mean, sd, min, max, iqr, and median are calculated.
- **Data Split**:
  - 365 Split: The data are split at the 365 second of the simulation. This is approximately half way through the entire simulation and is in between texting events for all subjects.
  - Entire Sim: The training set is randomly selected from the entire simulat

### Model Fitting with the Caret Package

In order to standardize the results of my analysis, I have elected to use the R package caret to train all of my models (http://topepo.github.io/caret/index.html). Caret is a modeling framework that allows you to run many different types of models with different combinations of parameter sets at the same time. It can easily use models from other popular packages and offers a rich set of validation tests and diagnostic plots. Another benefit to using this package is enabling parallel computing of cross validation tasks. For the following models I am using the nnet package which ships with base R. I am using k=10 cross validation for all models and Im training on different combinations of size and decay parameters. Training and testing sets are all approximately a 50/50 split.

### Model Performance

| Model | Data Processing | Data Split | MaxItr | Converged | Size | Decay | Training | Testing |
|---|---|---|---|---|---|---|---|---|
| Model 1: | Original | 365 Split | 500 | No | 15 | .20 | .783 | .690 |
| Model 2: | Original | Entire Sim | 500 | No | 100 | .10 | .798 | .796 |
| Model 3: | Differencing | 365 Split | 500 | Yes | 15 | .05 | .594 | .571 |
| Model 4: | Differencing | Entire Sim | 500 | Yes | 15 | .05 | .592 | .591 |

| Model | Data Processing | Data Split | MaxItr | Converged | Size | Decay | Training | Testing |
|---|---|---|---|---|---|---|---|---|
| Model 5: | Moving Avg | 365 Split | 500 | Yes | 30 | .05 | .525 | .527 |
| Model 6: | Moving Avg | Entire Sim | 500 | Yes | 10 | .00 | .528 | .528 |
| Model 7: | 1/2 Sec Cut | 365 Split | 500 | Yes | 100 | .00 | .884 | .722 |
| Model 8: | 1/2 Sec Cut | Entire Sim | 500 | Yes | 100 | .10 | .839 | .816 |
| Model 9: | 1/2 Sec Diff | 365 Split | 500 | Yes | 100 | .10 | .716 | .647 |
| Model 10: | 1/2 Sec Diff | Entire Sim | 100 | Yes | 100 | .10 | .687 | .625 |
| Model 11: | 1/2 Sec Cut Stat | 365 Split | 100 | Yes | 100 | .00 | .849 | .717 |
| Model 12: | 1/2 Sec Cut Stat | Entire Sim | 100 | Yes | 75 | .10 | .826 | .809 |

- **Takeaways**:
  1) Model 8 currently has the best overall performance.
  2) Overall the models trained over the entire simulation fair better than models trained on the time split data sets. This suggests that there are could be distinct differences between two texting segments.
  3) Model 2 took 3 days to train on 6 cpus. Its interesting that it still did not converge during this time and would have continued improving if given more iterations. The testing and training scores are virtually even so Model 2 could probably be given more iterations.
  4) Overall the models built on data that was differenced perform the worst. The differnced models using the 30 second intervals is much improved over the the original differencing models.
  5) Models 2 and 8 are using the same dataset except that model 8 has cut the data into 30 second intervals. Performance actually improves when you look at the data at a slightly higher level indicating that looking at the raw data measured every .03 seconds might actually make inference more difficult
  6) Model 8 and Model 12 are different only in that Model 12 has many more descriptive statistic variables. It is interesting that Model 12 had slightly worse performance then Model 8 given there it had more information.
  7) The number of hidden nodes has a diminishing return on model accuracy. For model 8 there difference in accuracy between 25 nodes and 100 is negligable, but the difference in training times was significantly larger for the bigger model.

### Time Omission

Time has been intentionally omitted from all of the models. The training/testing strategies in this analysis include either splitting the data roughly in the middle of the simulation or sampling from the entire simulation. Since texting events happen during specific fixed intervals of each simulation, a neural net will learn that time is an important factor in determining texting and the results will be biased. I discovered this when I initially included the time variable in the split data set. The training set performance measured very high with an overall accuracy of .95, but the testing accuracy was 0 because the model only saw time observations between 0-365 seconds and so it predicted no texting for all observations >365.

## Best Model Analysis

The best model is Model 08 which used data segmented into 1/2 second intervals and computed the mean value for each facial expression over the entire interval (15 observations). One of the easiest ways to assess the model fit was to produce the same emotion plot from earlier in the analysis and change the color of the background panel to identify the actual texting intervals. Then I plotted the model predictions rather than the actual data to examine the proportion of predictions that fell within the texting interval.

- **Takeaways**:
  1) The highest accuracy by subject was .971 (Subject 022), the lowest accuracy by subject was .513 (Subject 038).
  2) 7 subjects had an overall accuracy < .7, 15 subjects had an overall accuracy > .9.
  3) The model performed the worst on subjects with less variation in their emotional likelihood scores.

### Interpreting Weights

In neural net models, weights are the interconnection between Input_Node -> Hidden_Layer, and Hidden_Layer -> Output_Node. They are analogous to coefficients in regression models. The sum product of the weights and input nodes generates the model output after going through a transformation (activation function). In the case of my current logistic regression setup, the activation function is the sigmoid function (shown in a previous report). In the current best model there are (68 Input Nodes * 100 Hidden Nodes) + (1 Bias Correction * 100 Hidden Nodes) + (100 Hidden Nodes * 1 Output Node) + (1 Output Bias Correction) = 7001 Weights.

Neural nets do not produce standard errors, tvalues, or pvalues like traditional regression models do for coefficients. In fact neural networks are not really intended to be looked at under the hood. Most of the diagnostics for neural nets rely on examining the differences between training and testing set performance differences. That being said the hidden nodes within a neural net can be thought of as individual logistic regression models. You can extract the weights between the input nodes and the hidden layers and examine the distribution of weights for common indicator variables like age, gender, and subject.

- **Takeaways**:
  1) Extracting and plotting the weights for any one variable returns a distribution centered at 0.
  2) Doing inference on extracted weights is very difficult and not typically done.

### Variable Importance

One way of assessing variable importance is to look at relative contribution of each input variable to the hidden nodes and output nodes. I used a functiont to extract, compute, and plot the relative importance for each predictor variable excluding subject. I also plotted the relative importance score for each Subject against the overall testing accuracy of that subject. The method for assessing variable importance was originally outlined in this article: http://www.sciencedirect.com/science/article/pii/095418109400011S

- **Takeaways**:
    1) The Neutral emotion has the highest importance, followed by Surprise, Anger, and Gender.
    2) Contempt has the lowest importance of all predictors including Subject.
    3) The Surprise emotion is the second most important variable but has very little variation when plotted next to the other emotions. The variable importance plot of the earlier Random Forest model also listed Surprise as relatively high importance.
    4) There is a slight negative correlation (-.25) for subject relative importance compared to subject model accuracy. Removing the few outliers decreases the correlation (-.16).

### Texting Event Probability Conditional on Emotion

In order to assess how the probability of a texting event changes conditional on emotion, I created a series of plots where the output is the probability prediction for each subject based on the value of each emotion relative to their individual baseline value. I ran the prediction for each emotion in intervals of .1 between [-1, 1]. All other emotions were fixed at 0 (original baseline value). The idea behind this was to simulate each subjects normal state as they were driving in the non-event simulation (Trial 004) and to see how the probability of a texting event changes when you change a particular emotion.

- **Takeaways**:
    1) Contempt and Joy have to lowest variable importance scores. They also have similar plots in that the median prediction quickly moves to 1 or 0 when the emotional likelihood moves slightly away from the baseline. This lack of variation could support why the variable importance is so low.
    2) Sad has the next lowest variable importance score, but unlike the other emotions as the value of Sad increases, the likelihood of texting events increase, but then sharply decreases near the end of the interval.
    3) Fear, Anger, and Disgust, while not the most important variables, exibit a smoother relationship to the probability of texting events as their values change. The median prediction of digust tends to increase away from the baseline. As fear increases, the probabiliy of texting events diminishes to 0. As anger moves away from the baseline, the variation in predictions increase although for the most part the median prediction remains low except for the low point of the interval. All three emotions show a lot of variability in the predictions which may indicate why the variable importance scores are not very big.
    4) Surprise and Neutral are the two most important variables. As their values increase the probability of texting events increases. The variation in predictions is less than the other emotions as well. Surprise has more outlier predictions than Neutral which may indicate why its variable importance score is lower than Neutral.

**Demographic Effects**

To assess the significance of the demographics data I used the output of the best model to run an analysis of variance in order see if gender and age had an effect on the probability of texting events when emotions changed. All emotions except one were held constant during the predictions so I expected all of the emotions to test as being significant individually therefore emotion is redundant and not in the table below as a variable. Each column represents the significance of each variable when the emotion in question changes with all other emotions held constant. The following interpretations are relevant to prediction effects only. Emotions are listed in the order of their relative importance.

| Variable | Neutral | Surprise | Anger | Disgust | Fear | Sad | Joy | Contempt |
|---|---|---|---|---|---|---|---|---|
| Age | . | ** | *** | *** | *** | | *** | * |
| Gender | *** | *** | | | ** | | *** | . |
| Age*Gender | *** | | *** | * | | | . | *** |
| Emotion*Age | | ** | . | *** | *** | *** | *** | ** |
| Emotion*Gender | *** | | | | *** | *** | * | |
| Emotion*Age*Gender | * | | * | *** | *** | *** | | ** |

**Significance**: [. p-value < .1] [* p-value <.05] [** p-value < .01] [*** p-value < .001]

- **Takeaways**:
  1) The main effect of age is significant for 5 of the 8 emotions
  2) The main effect of gender is significant for 4 of the 8 emotions
  3) There is a significant 2-way interaction between age and gender for 4 of the 8 emotions
  4) Between Neutral and Surprise, all of the variables and interactions are significant
  5) There tends to be more significant variables for the emotions that are less important. Fear, Sad, Joy, and Contempt have a lot of overlapping variables that are significant. This might imply a reason that the neural net sees these variables as less important. They could have a similar effects.