**Predictive Model on Respiratory Viral Dream Challenge  
(Summer Independent Study)**

By

Shreya Chakrabarti

April 2017

Submitted To

Dr. Yuni Xia

The Faculty of Purdue University

In Partial Fulfillment of the

Requirement for the Degree

Of

**Master of Science in Computer & Information Science**

**May 2017**

**Index**

Introduction…………………………………………………………………………………....3

Data Description……………………………………………………………………………….4

Materials and Methods………………………………………………………………………...7

Results…………………………………………………………………………………………11

Conclusions……………………………………………………………………………………12

**Introduction**

Respiratory viruses are highly infectious and cause acute illness in millions of people every year. However, there is wide variation in the physiologic response to exposure at the individual level. Some people that are exposed to virus can completely avoid infection. Others contract virus but can fight it off without exhibiting any symptoms of illness such as coughing, sneezing, sore throat or fever. It is not well understood what characteristics may protect individuals from respiratory viral infection.

In sub challenge 1, we are interested in the prediction of individuals that become infected with virus and contagious following nasal exposure. This is quantified by measuring shedding of virus from nasal passages on subsequent days’ post exposure.

In sub challenges 2, we are interested in the prediction of individuals that demonstrate symptomatic response following viral exposure. In sub challenge 2, symptoms are measured as a binary outcome defined by whether an individual becomes symptomatic following exposure.

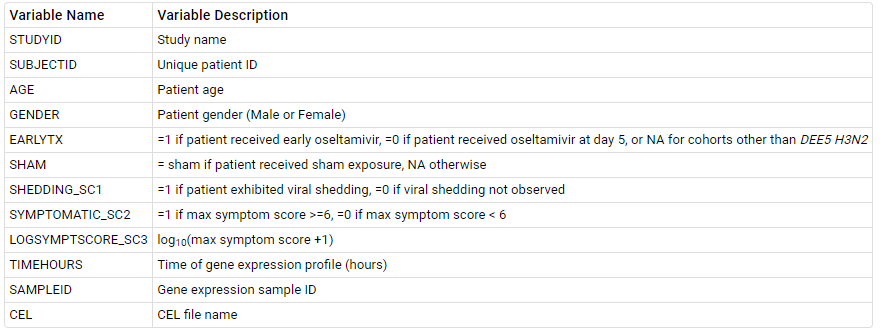
However, in this project I have spent consistent amount of time to determine accurate models for these sub challenges.

**Data Description**

Challenge data comes from 7 related viral challenge trials, representing 4 different respiratory viruses. The challenges are DEE1 RSV, DEE2 H3N2, DEE3 H1N1, DEE4X H1N1, DEE5 H3N2, Rhinovirus Duke, and Rhinovirus UVA. In each of these trials, healthy volunteers were followed for seven to nine days following controlled nasal exposure to one respiratory virus. Subjects enrolled into these viral challenge experiments had to meet several inclusion and exclusion criteria.

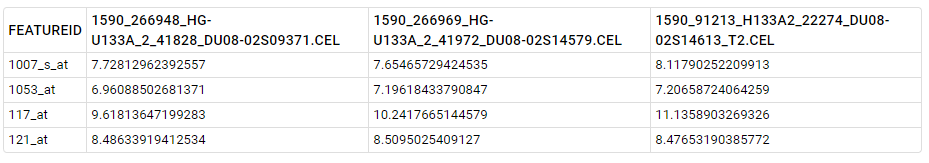
**Clinical Data**

A clinical data file is provided which includes clinical/demographic data, outcome variables, and information matching sample IDs to expression data at each timepoint. As such, each patient occurs multiple times in the file because multiple timepoints are represented per patient.



**Expression Data**

Gene expression profiling was performed on the Affy Human Genome U133A 2.0 array. Both a raw and normalized version of the gene expression data are available for use in this challenge. Both versions contain only profiles that pass QC metrics including those for RNA Degradation & scale factors



**Objectives**

* To develop a model to predict patients that exhibit viral shedding after viral exposure

Viral exposure in the data is measured in the form of shedding that occurs in a patient after exposure to virus. The parameter for measure in the data is Shedding\_SC1.

The graph below shows symptoms described by Severity in each of the viruses Patients were exposed.

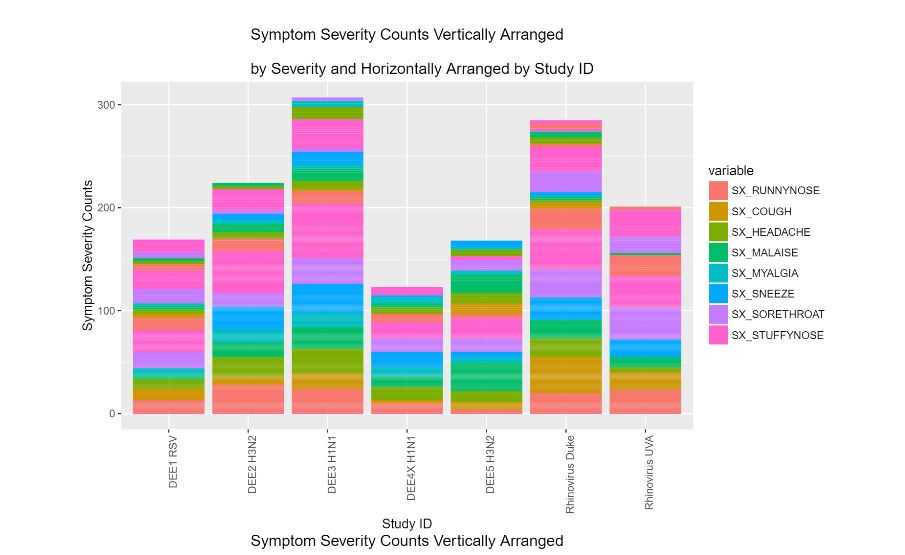


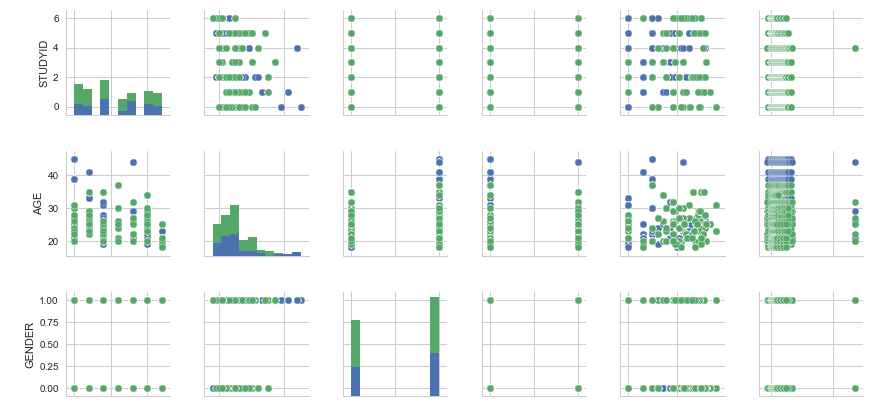
Fig.1

* To develop a model for the identification and prediction of patients that become symptomatic

Continuous exposure to virus can lead to a patient becoming symptomatic to the virus and develop illness There were cases where the patient has dropped out of the study because of symptoms prolonged exposure. The goal of this model was to measure which patient would be more prone to an infection than other.

**Materials and Methods**

**Data preprocessing**



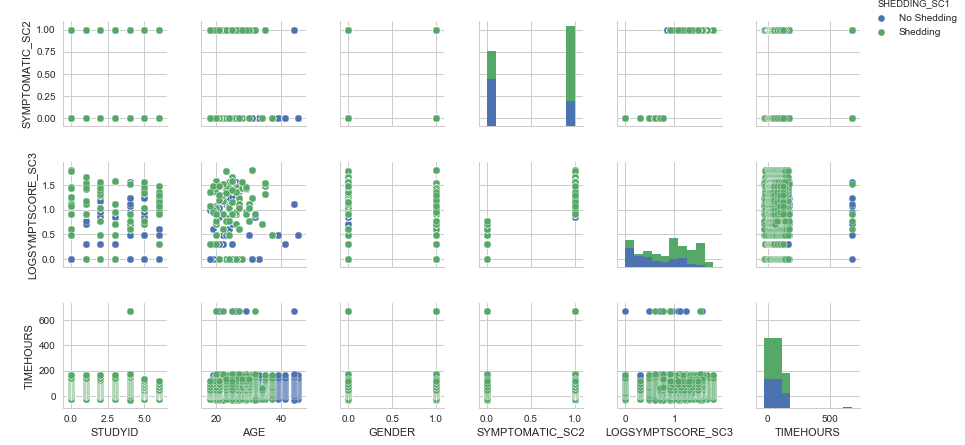


Fig.2

Figure shows a pair plot of all the features available for prediction in clinical data file paired with if shedding was observed as compared to the feature.  
  
Apart from all the microarray probe ids, I also took virus type, age, gender and time hours as input features for the training data dividing the training dataset into various ratios of training and testing split.

**Data Preprocessing**

**Step 1: Scaling the data**Feature scaling is the method to limit the range of variables so that they can be compared on common grounds. It is performed on continuous variables.

Feature scaling is done to prevent the largest range from dominating the outcome results and obtain less accurate predictions.

**Step 2: Dropping unnecessary columns**Unnecessary feature columns like‘SubjectID’, ‘EarlyTX’, ‘Sham’, ‘Cel’ & ‘SampleID’ are dropped.  
Also Gender and Study ID are converted into numerical values for ease of processing

**Step 3: Feature Dimensionality Reduction**Dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration. Since there are 22000 features mentioned in the project, using dimensionality reduction these features were reduced to 6000 valid features.

**Advantages of dimensionality reduction**

1. It reduces the time and storage space required.
2. Removal of multi-collinearity improves the performance of the machine learning model.

**Feature Selection**

**Random Forest Classifier to improve accuracy**

Random Forest classifier was used to decrease impurity in the data. Random forest consists of several decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. For classification, it is typically either [Gini impurity](http://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity) or [information gain/entropy](http://en.wikipedia.org/wiki/Information_gain_in_decision_trees) and for regression trees it is [variance](http://en.wikipedia.org/wiki/Variance). Thus, when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

For Phase 1: Top 100 Features were chosen  
  
For Phase 2: Top 100 Features were chosen

**Model Building and Evaluation**

Three models were built on the training data and were supplied with new data available for each phase to determine if a patient will experience viral shedding or will develop symptoms to the virus he has been exposed to. The Models were compared with respect to accuracy they offered based on which it was determined Support Vector Machines is a good fit for the available data.

As both the sub challenges expected binary output, Logistic Regression was thought to be a good fit for the available data.

**Logistic Regression**

Logistic Regression  is a [regression](https://en.wikipedia.org/wiki/Regression_analysis) model where the [dependent variable (DV)](https://en.wikipedia.org/wiki/Dependent_and_independent_variables) is [categorical](https://en.wikipedia.org/wiki/Categorical_variable).

The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the probability of a given outcome by a specific percentage.

Logistic regression can be binomial, ordinal or multinomial. Binomial or binary logistic regression deals with situations in which the observed outcome for a [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) can have only two possible types, "0" and "1" (which may represent, for example, "dead" vs. "alive" or "win" vs. "loss").

In Sub-Challenge 1 as well as 2 Binomial Logistic Regression was used to determine if the individual will develop viral shedding or symptoms to the virus he has been exposed to.

**Support Vector Machines**

A simple way to classify observations into two classes is to draw a linear boundary between them.However even if the data is perfectly seperable in this way there are many possible linear boundaries that can be used which is the best? One approach is to use some kind of statistical distribution fit.However this means that even points far from the boundary have an influence on where the boundary is located.Intuitively, it seems like the better approach to the problem will be to put the boundary as far as possible from any of the observations,this is the basic idea behind support vector machine classification.If the data is actually linearly separable this results in a simple optimization problem.Choose the linear coefficients of the boundary to maximize the margin i.e the distance between the boundary and the nearest observations subject to the constraint that all observations must be on the correct side of the boundary. The optimal solution is determined only by the observations nearest to the boundary.These observations are referred to as the support vectors.Real noisy data may not be linearly seperable i.e there is no linear boundary that can correctly classify every observation.In this case we can modify the optimization problem to maximize the margin but with a penalty term for misclassified observations. Observations are correctly classified only if they lie on the correct side of the margin so the penalty term prevents a solution that cheats by having a huge margin.The SVM solution then is the one that gives the best possible separation between classes i.e the widest margin without unnecessary misclassifications.Linear boundaries between classes is not appropraite for all problems.However SVM’s can still be used on non-linear classification problems by performing a transformation of variables into the space where the classes are linearly seperable.The linear boundary in that space is equivalent to non-linear boundary back in the orignal variable space. SVM classification works only in the Binary Classification problems.Multiclass problems are solved by combining multiple binary classifiers.

As new data was to be supplied in each of the phases support vector machines model was chosen as it has the functionality to perform linear as well as non-linear classification of data.

**Naïve Bayes Classifier**

Naïve Bayes Classification works by assuming that the observations in each response class are samples from probability distributions. A separate distribution for each class. If we knew those probability distributions, We can then calculate probability for each class & then classify the observation according to which class is the most likely.

The Naïve assumption in Naïve Bayes is that each variable is independent in each response class that’s almost certainly not true. But this often works anyway and it simplifies the calculations. Fitting a normal distribution in each variable independently involves simply calculating the mean and standard deviation of each combination of class and variable. Performing predictions involves only determining the probability of the observation and applying the formula of Bayes’ theorem. Because we are assuming that the distributions are independent in each predictor, the probability is simply the product of the probabilities in each variable. One benefit of this probabilistic approach is that probabilities give some indication of how clear the classification is. Also, as the predictions are based on the statistical observations rather than individual observations they are somewhat robust noise in the training data.

I used naïve Bayes to check if the features individually could give a better accuracy. It was observed that training dataset performed very well on Naïve Bayes Classifier.

**Tools**

Scikit Learn library in python was used to develop models.

Numpy & Pandas libraries in python were used for creating data frames and to perform computations on the data frames.

I was completely unfamiliar with machine learning and tried various tools like Excel, Tableau, MATLAB.

I choose Python as I could understand the code for machine learning best.

**Results**

**Sub Challenge- 1**

The Following observations are recorded with respect 10 splits of training and test.

Range of Accuracies observed with respect to data is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Support Vector Machines | Logistic Regression | Naïve Bayes | Decision Tree |
| Test Data | Run 1 | 0.789325842697 | 0.737359550562 | 0.648876404494 | 0.643258426966 |
| Run 2 | 0.830056179775 | 0.768258426966 | 0.672752808989 | 0.648876404494 |
| Run 3 | 0.794943820225 | 0.712078651685 | 0.671348314607 | 0.640449438202 |
| Run 4 | 0.807584269663 | 0.779494382022 | 0.667134831461 | 0.634831460674 |
| Run 5 | 0.821629213483 | 0.77808988764 | 0.667134831461 | 0.660112359551 |
| Run 6 | 0.807584269663 | 0.787921348315 | 0.643258426966 | 0.681179775281 |
| Run 7 | 0.825842696629 | 0.761235955056 | 0.641853932584 | 0.613764044944 |
| Run 8 | 0.823033707865 | 0.77106741573 | 0.683988764045 | 0.619382022472 |
| Run 9 | 0.814606741573 | 0.747191011236 | 0.637640449438 | 0.619382022472 |
| Run 10 | 0.814606741573 | 0.73595505618 | 0.65308988764 | 0.625 |
| **Average Accuracy** | **0.812921** | **0.757865** | **0.658708** | **0.638624** |

**Sub Challenge 2**

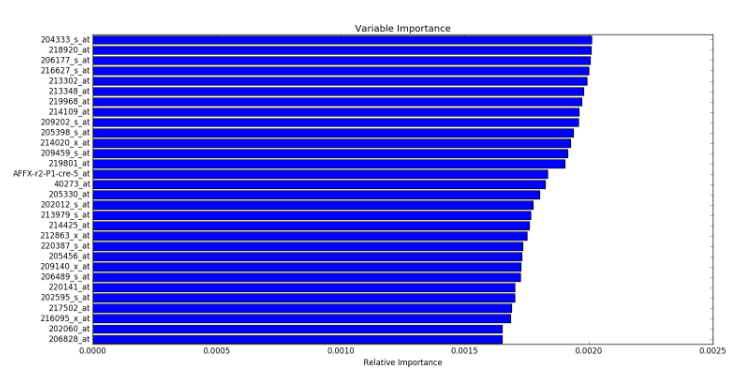
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Support Vector Machines | Logistic Regression | Naïve Bayes | Decision Tree |
| Test Data | Run 1 | 0.794943820225 | 0.731741573034 | 0.61797752809 | 0.595505617978 |
| Run 2 | 0.780898876404 | 0.731741573034 | 0.623595505618 | 0.61095505618 |
| Run 3 | 0.786516853933 | 0.695224719101 | 0.606741573034 | 0.575842696629 |
| Run 4 | 0.803370786517 | 0.695224719101 | 0.595505617978 | 0.598314606742 |
| Run 5 | 0.785112359551 | 0.712078651685 | 0.61797752809 | 0.601123595506 |
| Run 6 | 0.773876404494 | 0.669943820225 | 0.592696629213 | 0.578651685393 |
| Run 7 | 0.818820224719 | 0.674157303371 | 0.640449438202 | 0.584269662921 |
| Run 8 | 0.786516853933 | 0.720505617978 | 0.613764044944 | 0.598314606742 |
| Run 9 | 0.787921348315 | 0.679775280899 | 0.595505617978 | 0.63904494382 |
| Run 10 | 0.804775280899 | 0.705056179775 | 0.609550561798 | 0.570224719101 |
| **Average Accuracy** | **0.79227528** | **0.70506** | **0.610643** | **0.59225** |

Based on the accuracies observed while using this dataset it is observed that Support Vector Machines is the best suitable model for this dataset.

**Conclusions**

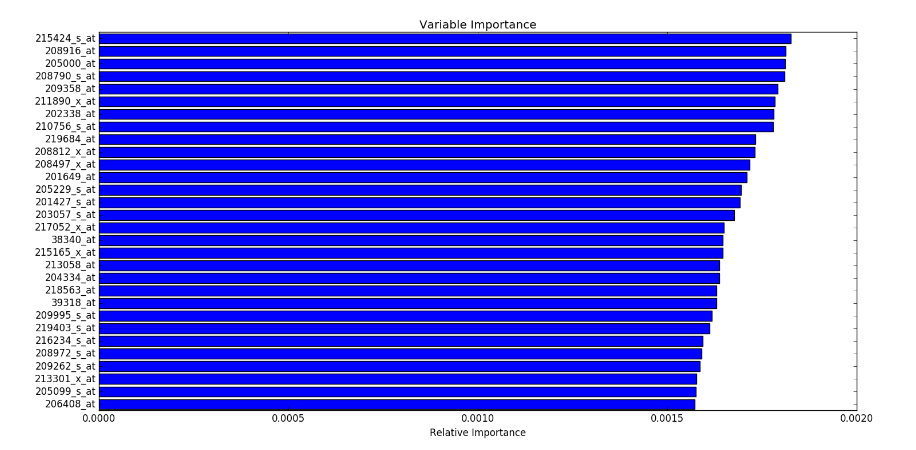
* Logistic Regression can be employed in Phase 1 for predictions worth 90% accuracy
* Support Vector Machines give the most efficient accuracy in Phase 2 of over 79% accuracy
* Based on the predictions of viral shedding and symptoms developed by the patient using this dataset it can be determined how long it will take for a patient to develop an infection to any of the 7 viruses
* It can be further enhanced to develop models to determine an individual’s immunity

**Top 20 Features for Phase 1**



Some of the top features according to importance for phase 1 are visible in the variable importance bar chart above.

**Top for Phase 2**

Some of the top features according to importance for phase 2 are visible in the variable importance bar chart above.

**Reference:**

<https://en.wikipedia.org/wiki/Support_vector_machine>

<https://en.wikipedia.org/wiki/Logistic_regression>

<https://www.synapse.org/#!Synapse:syn5647810/wiki/399103>

<http://scikit-learn.org/stable/>

<https://www.udemy.com/learning-python-for-data-analysis-and-visualization/learn/v4/overview>

[www.youtube.com](http://www.youtube.com)

<https://www.mathworks.com/solutions/machine-learning.html>

<http://blog.datadive.net/selecting-good-features-part-iii-random-forests/>