Final Project: Comparing Classifiers

The goal of this project is to compare different classification algorithms to demonstrate how models can differ in their predictions accuracy. To do this, we will train 5 different models (Regularized logistic regression, Random Forrest, and 3 different neural networks each with different layer and neuron configurations).

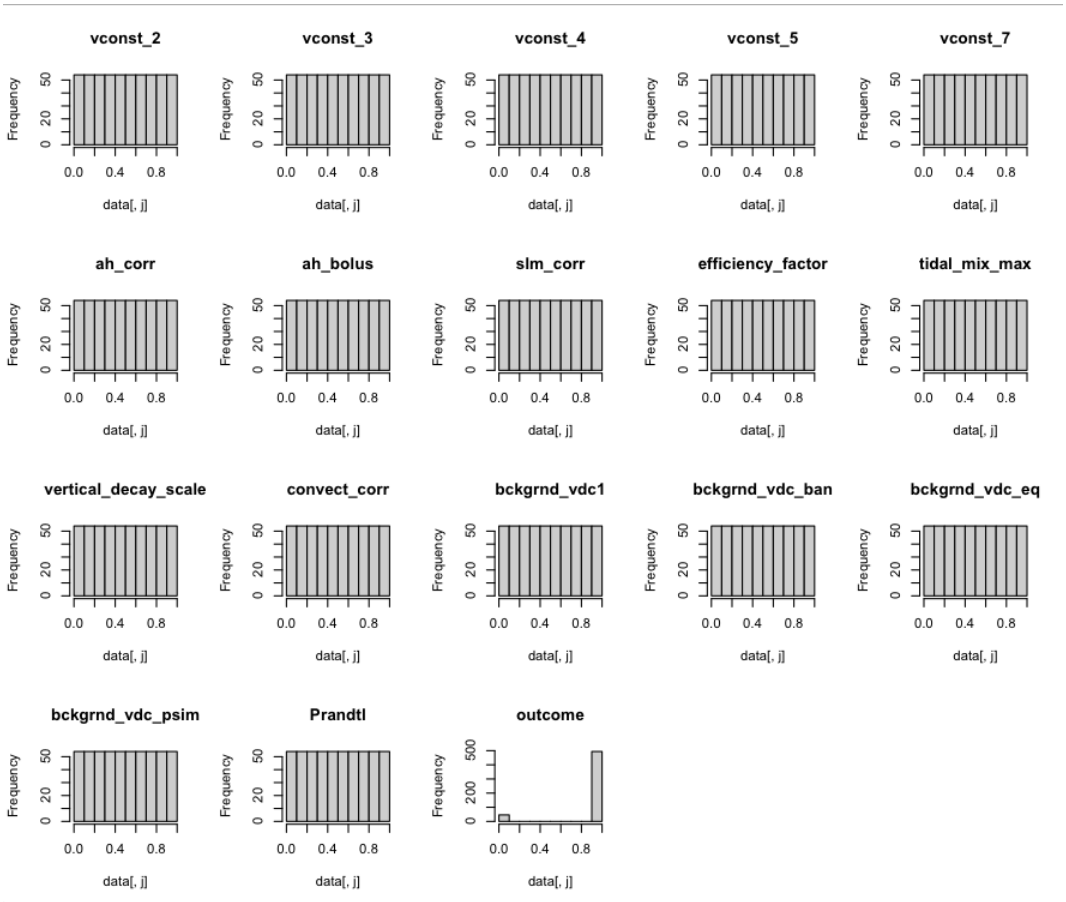


Figure 1 Visualization of Dataset

First, we will do some EDA on the dataset to get a better understanding of the problem. The first thing I notice is that all the columns have uniform distributions. In addition to this, I notice that the data has been standardized for a range of [0,1.0]. The third thing I notice is that this classification problem is imbalanced. This can be seen by looking at the histogram of the outcome variable. There are far more occurrences of 1 than there are of 0, which could make this take the form of an outlier detection problem. We also ran a correlation matrix on the data. There seems to be no multicollinearity present in this dataset. After some visualization, I found no NA values in the dataset, so no imputation was necessary.

	vconst_corr	vconst_2	vconst_3	vconst_4	vconst_5	vconst_7	ah_corr	ah_bolus	slm_corr	e
vconst_corr	1.00000	0.004039	0.009331	-0.018294	0.01888	1.54e-03	0.00371	-0.012735	0.00234	
vconst_2	0.00404	1.000000	-0.000456	-0.000614	-0.00829	-2.44e-02	-0.00518	0.004179	-0.01386	
vconst_3	0.00933	-0.000456	1.000000	0.009899	0.00629	-1.59e-03	0.01994	0.004402	-0.00770	
vconst_4	-0.01829	-0.000614	0.009899	1.000000	0.02050	2.19e-02	0.00180	-0.002334	-0.00173	
vconst_5	0.01888	-0.008292	0.006289	0.020504	1.00000	5.89e-03	-0.00305	0.012453	0.00363	
vconst_7	0.00154	-0.024379	-0.001587	0.021931	0.00589	1.00e+00	-0.01677	-0.021644	0.00124	
ah_corr	0.00371	-0.005182	0.019941	0.001805	-0.00305	-1.68e-02	1.00000	-0.035498	-0.00512	
ah_bolus	-0.01274	0.004179	0.004402	-0.002334	0.01245	-2.16e-02	-0.03550	1.000000	-0.00940	
slm_corr	0.00234	-0.013860	-0.007695	-0.001731	0.00363	1.24e-03	-0.00512	-0.009403	1.00000	
efficiency_factor	0.01062	-0.011072	0.007100	-0.004753	0.00108	1.51e-02	0.00960	0.012260	0.00876	
tidal_mix_max	-0.01421	0.019706	-0.009428	0.018320	0.02135	7.45e-05	-0.00683	0.012005	0.00257	
vertical_decay_scale	-0.00899	0.001623	-0.024702	-0.010004	-0.01631	1.53e-02	0.01650	-0.003947	0.00227	
convect_corr	-0.00298	0.002608	-0.020637	-0.006762	0.02138	7.04e-03	0.00292	-0.019307	0.00263	
bckgrnd_vdc1	-0.00213	-0.014716	-0.004264	0.020442	0.00989	-3.64e-03	0.01245	-0.010642	-0.00304	
bckgrnd_vdc_ban	-0.00210	0.004386	-0.005210	-0.001080	-0.01918	-7.90e-03	-0.00337	0.004866	0.00602	
bckgrnd_vdc_eq	0.01597	0.005999	-0.000559	-0.009262	-0.02075	-6.58e-03	0.00705	0.032398	-0.00845	
bckgrnd_vdc_psim	-0.01663	0.004202	0.004771	-0.017147	-0.00932	1.32e-02	0.00244	0.000259	-0.00230	
Prandtl	-0.00147	0.009141	-0.001334	0.005053	0.01227	8.41e-03	-0.00238	0.007055	0.01428	
outcome	-0.30479	-0.302388	0.000227	0.072297	0.05439	4.86e-02	0.01705	0.003895	0.04886	

Figure 2 Snippet of Correlation Matrix

After we have looked have done the EDA, it is now time to train the models. First, we split the data into 2 parts training and 1 part testing. We train the models on the training dataset and collect prediction metrics based on the test set. The 5 training models in include: Lasso Logistic regression, Random Forest with standard settings, Neural Network 1(NNet 1) with 1 layer and 3 neurons, NNet 2 with 1 layer and 10 neurons, and NNet 3 with 3 layers with 3 neurons each.

As a quick aside, we plot the variable importance from the Random Forest. We can see from this plot that vconst_2 and vconst_corr are the most important variables in the dataset.

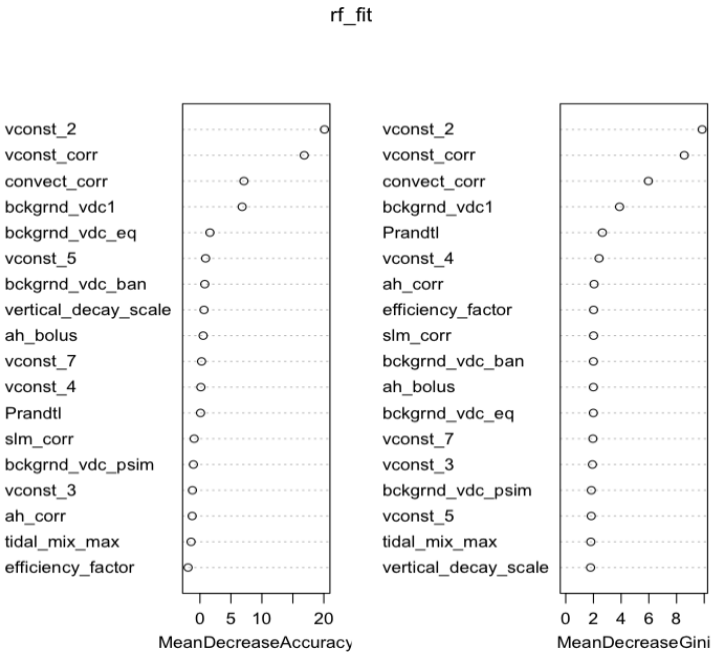
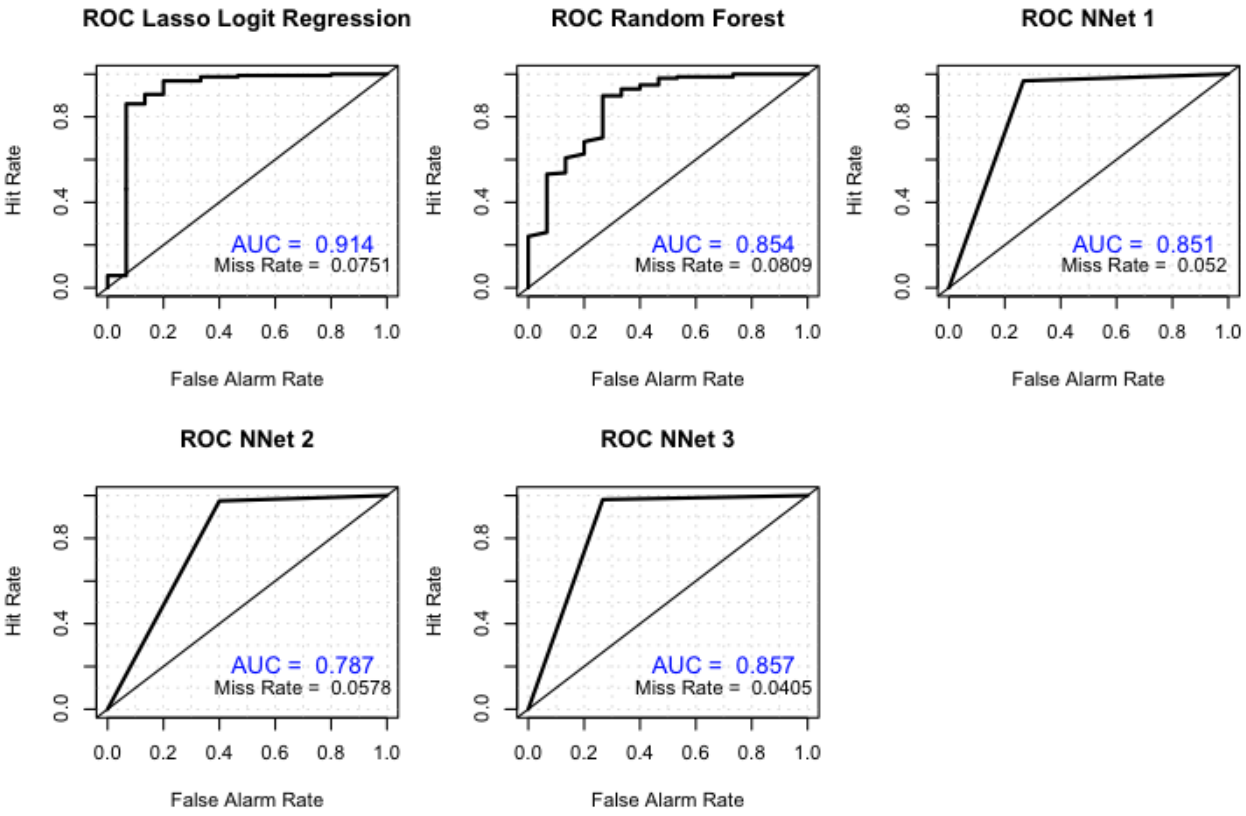


Figure 3 Variable Importance for Random Forest

Figure 3 shows the resulting ROC curve of each model, with missrate and AUC printed on the plot. From these plots, we can see that the Logistic Regression model has the highest AUC but Nnet 3 has the smallest missrate. This is likely because NNets are more prone to becoming overfit if there isn't a lot of data. This dataset only included 540 observations, which is not very much as far as model training goes. Its interesting to see that the AUC dropped from NNet 1 to NNet 2. It seems that increasing the amount of neurons within a layer has a negative effect on the classifier. NNet 3 performed better than both NNet1 and NNet 2. This suggests that increasing the layers in a Neural Network has a positive effect on the classification rate (at least in this example). The Random forest model seems to be middle of the road when compared to all of the others with and AUC = 0.854 and a Missrate = 8.09%. If I had to choose a model to proceed with predictions, I would choose the Lasso Logistic regression model. I would consider using NNet 3 if we could get a lot more data to train the model on.



Overall, this was a great exercise in differentiating the effectiveness of various classification model building algorithms. It seems that logistic regression does well on small data sets, but Neural Networks with multiple layers could do better if the dataset was much larger. The random forest model seems to be a decent model, though some tuning would need to take place to see some better results.