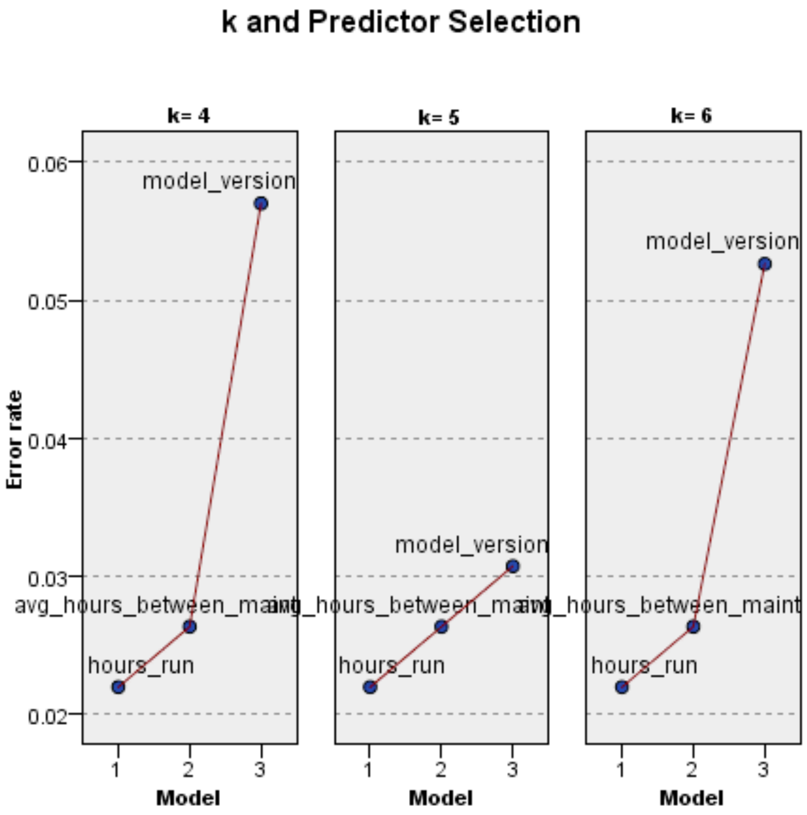
Austin

BIT-445-O500

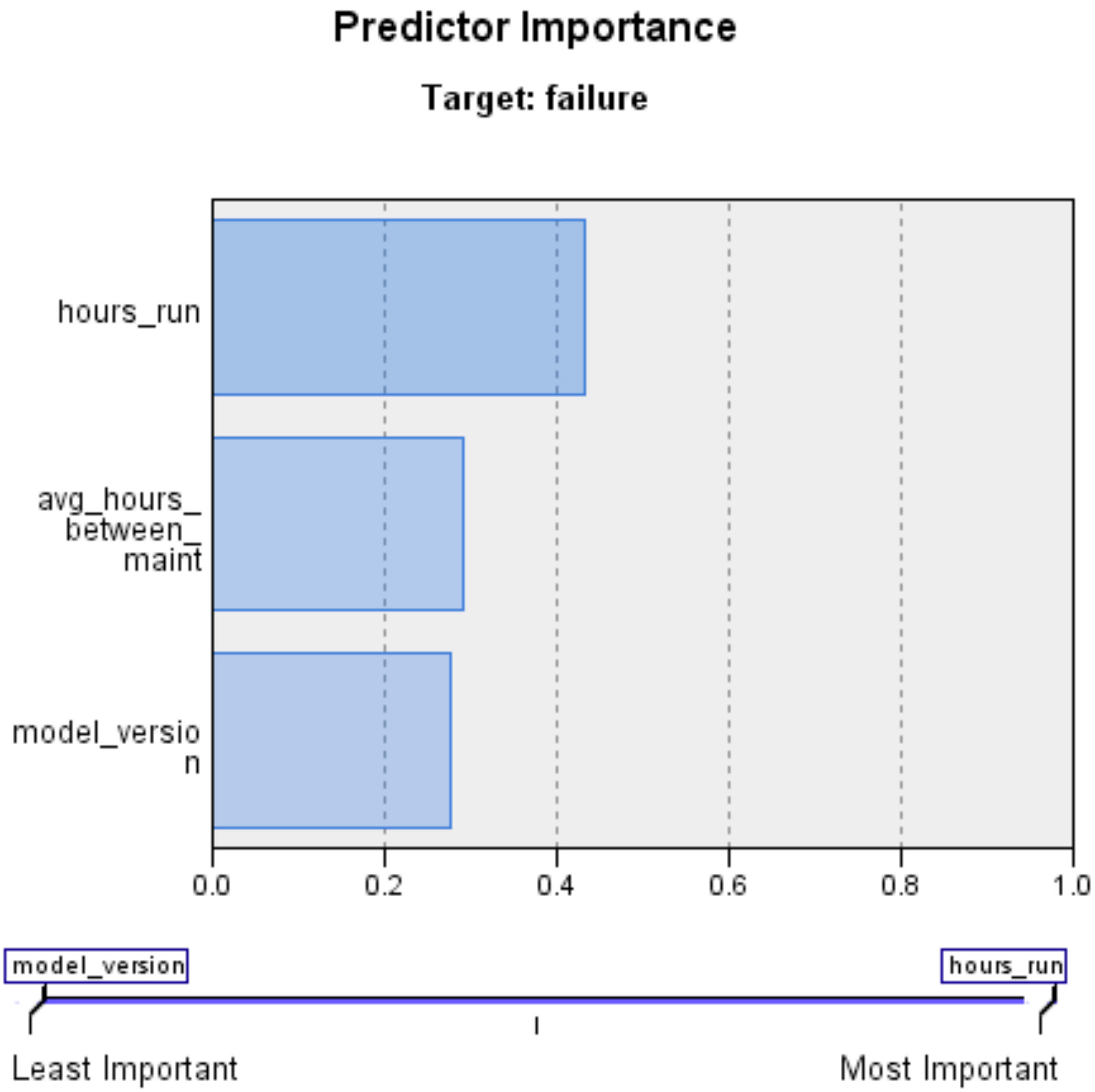
K-Nearest Neighbor Algorithm

1. The way I approached this problem was to follow the steps I would usually follow, then work on building the model and K-Nearest Neighbor analysis for the dataset. A K-nearest neighbor analysis is an algorithm that classifies data into different groups based on something they share in common. Using the K-nearest neighbor algorithm I was able to classify things based on the groups of what they are most similar to. After having done this classification, I was able to then analyze the results and take note of my observations.
2. K and Predictor Selection



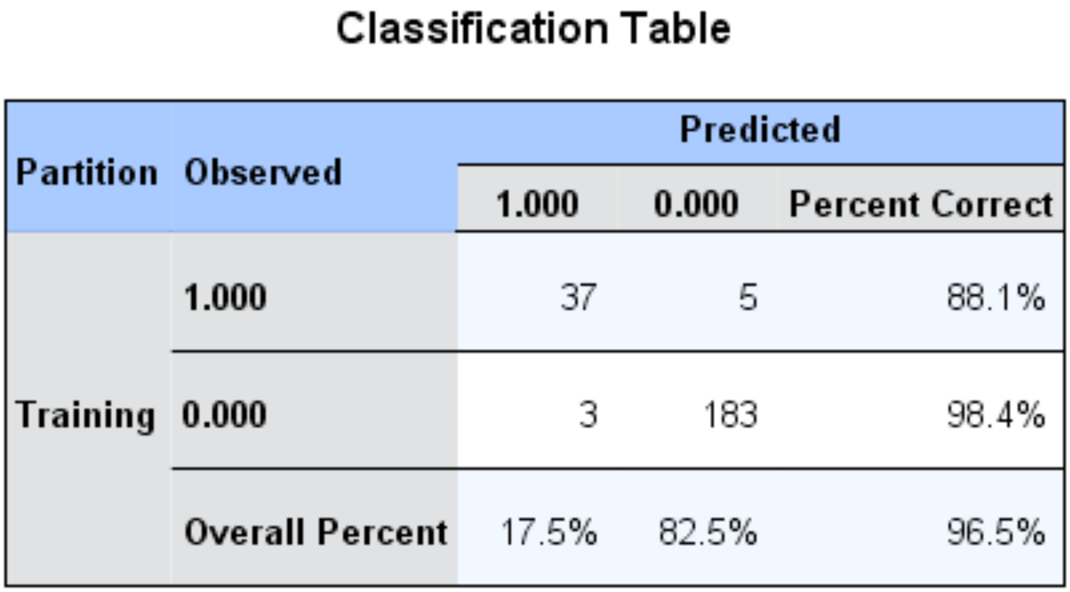
The optimal K-value will be 5 because the error rate for each variable is a lot lower than the K-values for 4 and 6, in which model_version's error rate is significantly higher.

3. Predictor importance



The most important variable is hours_run, and the least important variable is model_version. Even though hours_run is not that much further ahead than its neighbors, it still is the most important predicting variable.

4. Classification Table

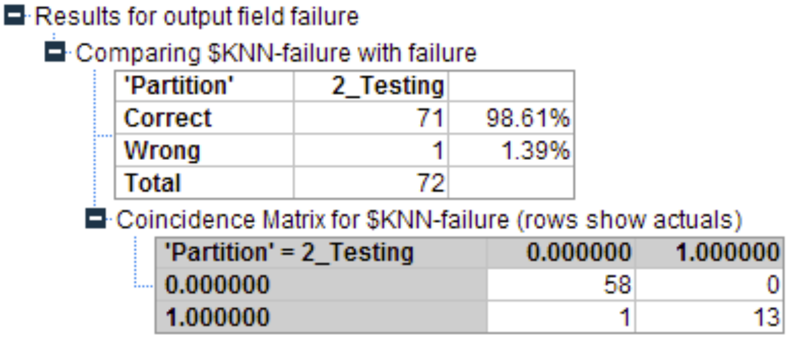


Classification Table

Partition	Observed	Predicted		
		1.000	0.000	Percent Correct
Training	1.000	37	5	88.1%
	0.000	3	183	98.4%
	Overall Percent	17.5%	82.5%	96.5%

The training model has a high level of accuracy with there being 82.5% in success, and 17.5% in failure.

5. Analysis node



Results for output field failure

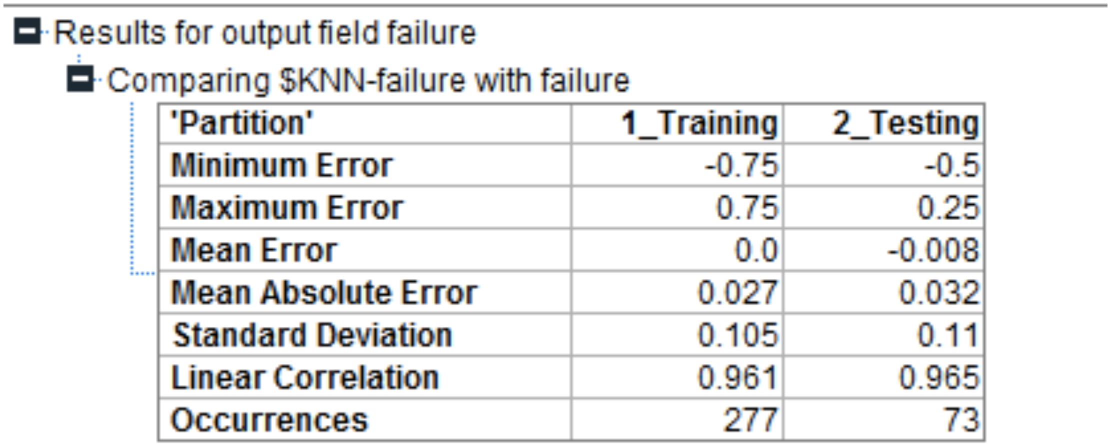
Comparing $KNN-failure with failure

'Partition'	2_Testing	
Correct	71	98.61%
Wrong	1	1.39%
Total	72	

Coincidence Matrix for $KNN-failure (rows show actuals)

'Partition' = 2_Testing	0.000000	1.000000
0.000000	58	0
1.000000	1	13

The test model has a higher level of success than the training model (test results include 1.39% failure and 98.61% success, and training results include 17.5% failure and 82.5% success.

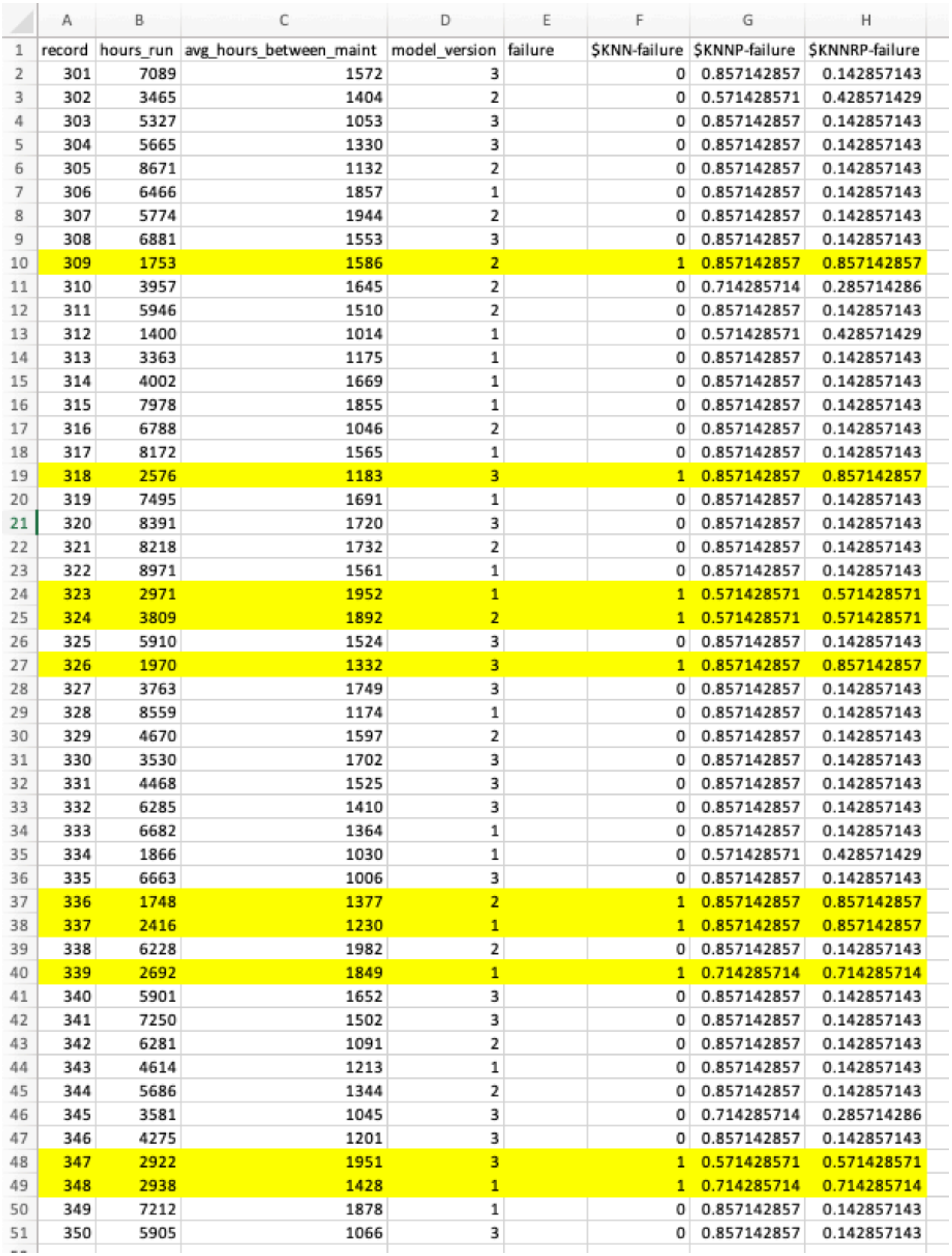


Results for output field failure

Comparing $KNN-failure with failure

'Partition'	1_Training	2_Testing
Minimum Error	-0.75	-0.5
Maximum Error	0.75	0.25
Mean Error	0.0	-0.008
Mean Absolute Error	0.027	0.032
Standard Deviation	0.105	0.11
Linear Correlation	0.961	0.965
Occurrences	277	73

6. Based on the results of the analysis, I can conclude that the K-nearest neighbor classification algorithm was a success in this scenario. The reason why was because the training model was outperformed by the test model, which means in the actual application it performed well. It was able to identify what the most important variable was (hours_run) and were able to find out a value for k that worked best, and identify a good model. The machines in records 301-350 that are predicted to fail are #'s 309, 318, 323, 324, 326, 336, 337, 339, 347, 348 (highlighted in yellow below).



record	hours_run	avg_hours_between_maint	model_version	failure	$KNN-failure	$KNNP-failure	$KNNRP-failure
301	7089	1572	3		0	0.857142857	0.142857143
302	3465	1404	2		0	0.571428571	0.428571429
303	5327	1053	3		0	0.857142857	0.142857143
304	5665	1330	3		0	0.857142857	0.142857143
305	8671	1132	2		0	0.857142857	0.142857143
306	6466	1857	1		0	0.857142857	0.142857143
307	5774	1944	2		0	0.857142857	0.142857143
308	6881	1553	3		0	0.857142857	0.142857143
309	1753	1586	2		1	0.857142857	0.857142857
310	3957	1645	2		0	0.714285714	0.285714286
311	5946	1510	2		0	0.857142857	0.142857143
312	1400	1014	1		0	0.571428571	0.428571429
313	3363	1175	1		0	0.857142857	0.142857143
314	4002	1669	1		0	0.857142857	0.142857143
315	7978	1855	1		0	0.857142857	0.142857143
316	6788	1046	2		0	0.857142857	0.142857143
317	8172	1565	1		0	0.857142857	0.142857143
318	2576	1183	3		1	0.857142857	0.857142857
319	7495	1691	1		0	0.857142857	0.142857143
320	8391	1720	3		0	0.857142857	0.142857143
321	8218	1732	2		0	0.857142857	0.142857143
322	8971	1561	1		0	0.857142857	0.142857143
323	2971	1952	1		1	0.571428571	0.571428571
324	3809	1892	2		1	0.571428571	0.571428571
325	5910	1524	3		0	0.857142857	0.142857143
326	1970	1332	3		1	0.857142857	0.857142857
327	3763	1749	3		0	0.857142857	0.142857143
328	8559	1174	1		0	0.857142857	0.142857143
329	4670	1597	2		0	0.857142857	0.142857143
330	3530	1702	3		0	0.857142857	0.142857143
331	4468	1525	3		0	0.857142857	0.142857143
332	6285	1410	3		0	0.857142857	0.142857143
333	6682	1364	1		0	0.857142857	0.142857143
334	1866	1030	1		0	0.571428571	0.428571429
335	6663	1006	3		0	0.857142857	0.142857143
336	1748	1377	2		1	0.857142857	0.857142857
337	2416	1230	1		1	0.857142857	0.857142857
338	6228	1982	2		0	0.857142857	0.142857143
339	2692	1849	1		1	0.714285714	0.714285714
340	5901	1652	3		0	0.857142857	0.142857143
341	7250	1502	3		0	0.857142857	0.142857143
342	6281	1091	2		0	0.857142857	0.142857143
343	4614	1213	1		0	0.857142857	0.142857143
344	5686	1344	2		0	0.857142857	0.142857143
345	3581	1045	3		0	0.714285714	0.285714286
346	4275	1201	3		0	0.857142857	0.142857143
347	2922	1951	3		1	0.571428571	0.571428571
348	2938	1428	1		1	0.714285714	0.714285714
349	7212	1878	1		0	0.857142857	0.142857143
350	5905	1066	3		0	0.857142857	0.142857143