Predictive Maintenance

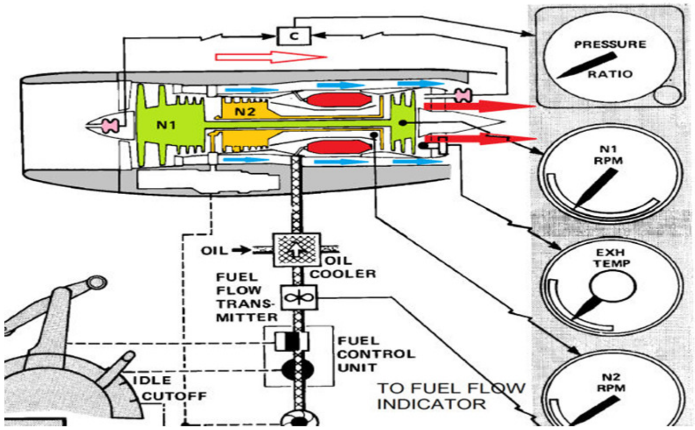
Exploratory Data Analysis
Regression Modeling
Binary Classification
Multiclass Classification

Sami Mustafa

The Problem

Airlines are interested in predicting engines failures in advance to enhance operations and reduce flight delays. Observing engine's health and condition through sensors and telemetry data is assumed to facilitate this type of maintenance by predicting Time-To-Failure (TTF) of in-service equipment. Using aircraft engine's sensors measurements, can we predict engine's TTF?

Solution Approach



By exploring aircraft engine's sensor values over time, machine learning algorithm can learn the relationship between sensor values and changes in sensor values to the historical failures in order to predict failures in the future.

Data

Text files contain simulated aircraft engine run-to-failure events, operational settings, and 21 sensors measurements are provided by Microsoft. It is assumed that the engine progressing degradation pattern is reflected in its sensor measurements.

Training Data

id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21
1	1	-0.0007	-0.0004	100	518.67	641.82	1589.7	1400.6	14.62	21.61	554.36	2388.06	9046.19	1.3	47.47	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100	39.06	23.419
1	2	0.0019	-0.0003	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04	9044.07	1.3	47.49	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100	39	23.4236
1	3	-0.0043	0.0003	100	518.67	642.35	1587.99	1404.2	14.62	21.61	554.26	2388.08	9052.94	1.3	47.27	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100	38.95	23.3442
100	198	0.0004	0	100	518.67	643.42	1602.46	1428.18	14.62	21.61	550.94	2388.24	9065.9	1.3	48.09	520.01	2388.24	8141.05	8.5646	0.03	398	2388	100	38.44	22.9333
100	199	-0.0011	0.0003	100	518.67	643.23	1605.26	1426.53	14.62	21.61	550.68	2388.25	9073.72	1.3	48.39	519.67	2388.23	8139.29	8.5389	0.03	395	2388	100	38.29	23.064
100	200	-0.0032	-0.0005	100	518.67	643.85	1600.38	1432.14	14.62	21.61	550.79	2388.26	9061.48	1.3	48.2	519.3	2388.26	8137.33	8.5036	0.03	396	2388	100	38.37	23.0522

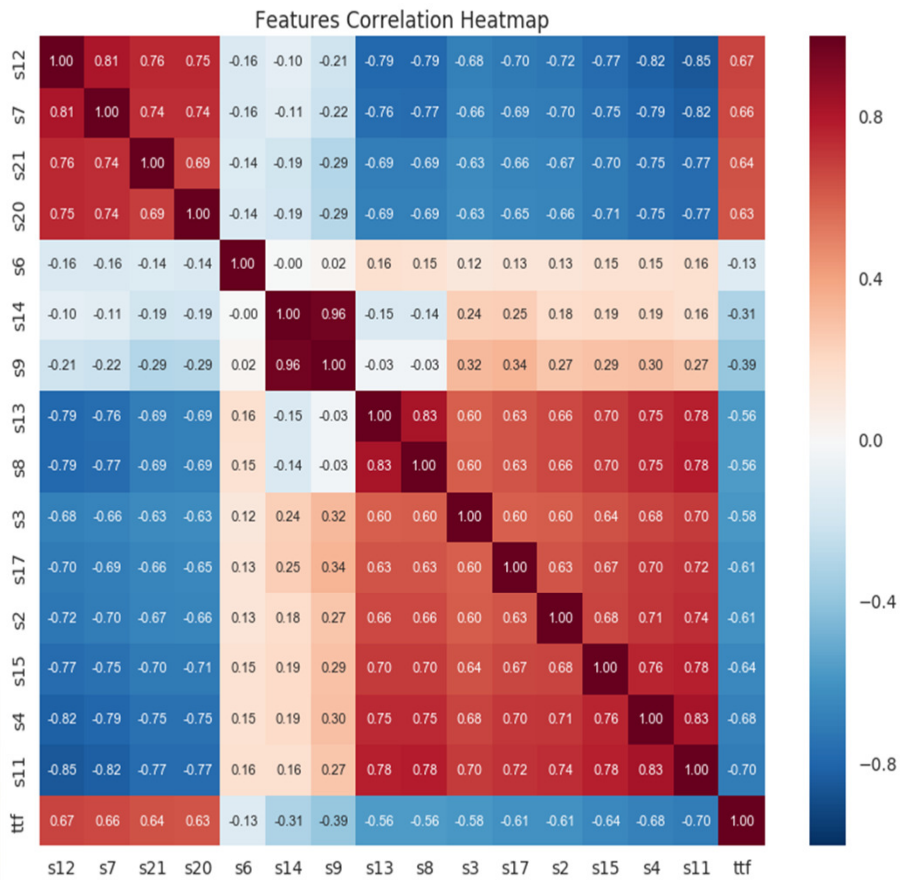
- ✓ **Id**: Engine Id. There 100 engines with average 109 cycles/engine in the training data
- ✓ **Cycle**: sequence per engine, starts from 1 to the cycle number where failure has happened
- ✓ **Setting1** to **setting3**: engine operational settings
- ✓ **S1** to **S21**: sensors measurements in each cycle

Labels:

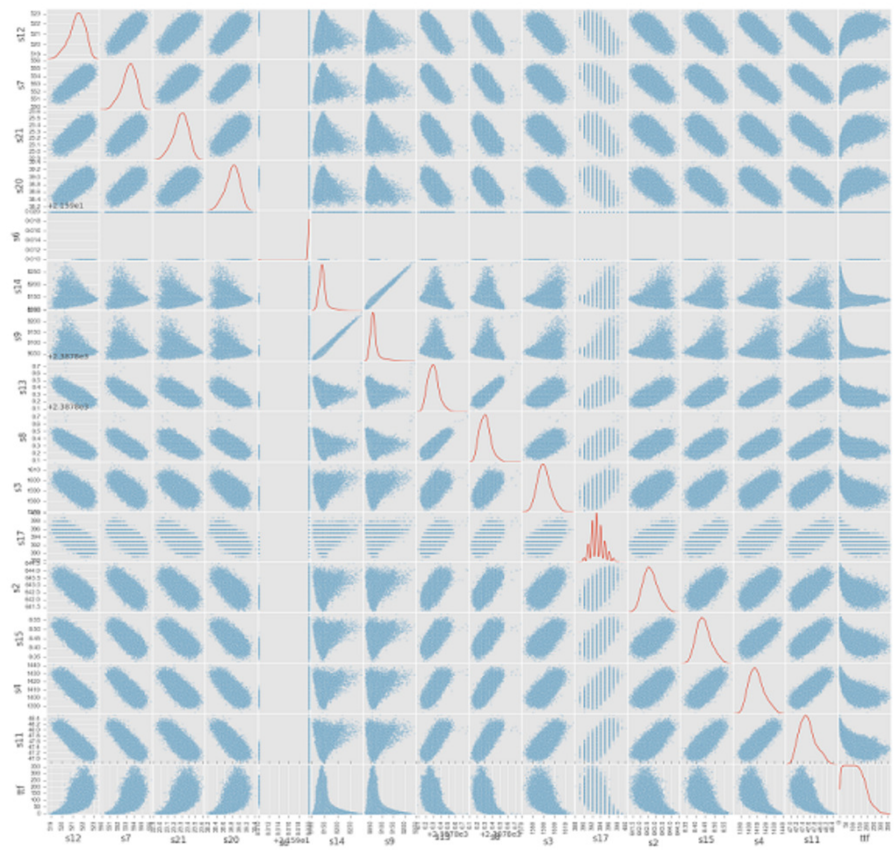
- ✓ **Regression**: Time-to-Failure (TTF), for each cycle/engine, is the number cycles between that cycle and last cycle of the engine.
- ✓ **Binary Classification**: if the remaining cycles (TTF) is less than specific number of cycles (e.g. 30) then the engine will fail in the next period, otherwise the engine is fine.
- ✓ **Multiclass Classification**: dividing TTF into bands (e.g. 0-15, 16-30, 30+), in which band will the engine fail?

Test Data: Similar to the training data, 100 engines, with one random cycle per engine, with addition of truth data (TTF).

Exploratory Data Analysis

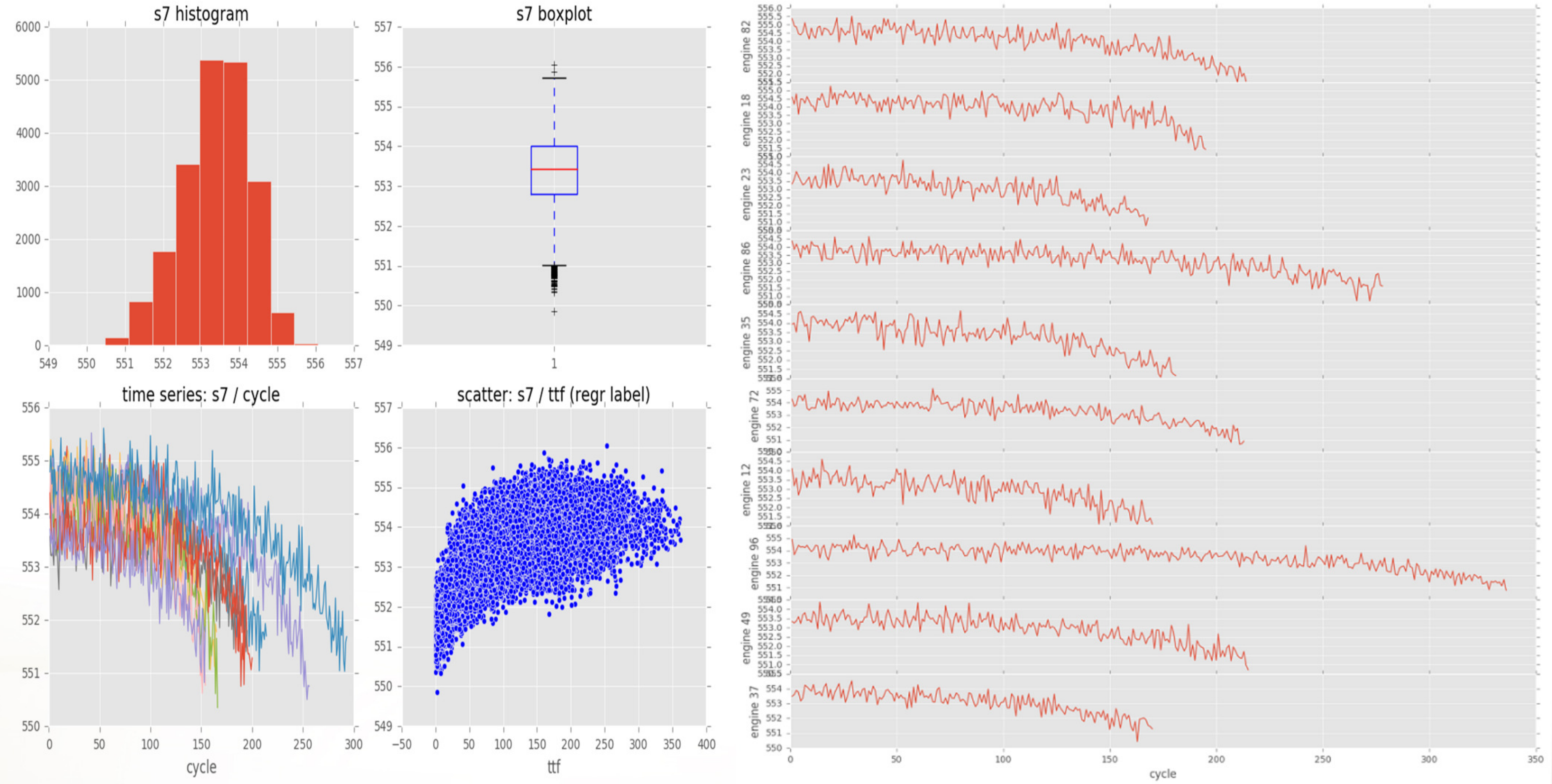


High positive/negative correlation between features/label (TTF), help in feature selection.



Most features have normal distribution and non-linear relationship with the label (TTF).

Exploratory Data Analysis (S7)



A set of typical EDA graphs applied to each feature individually

Regression Modeling

Used to predict the number remaining cycles before engine failure

Machine learning algorithms used:

- ✓ Linear Regression
- ✓ LASSO Regression
- ✓ Ridge Regression
- ✓ Decision Tree Regression
- ✓ Polynomial Regression
- ✓ Random Forest Regression

Regression metrics calculated:

- ✓ R-squared (R^2)
- ✓ Root Mean Squared Error (RMSE)
- ✓ Mean Absolute Error
- ✓ Explained Variance

Regression Modeling Results

	Linear	LASSO	Ridge	Decision Tree	Polynomia	Random Forest
Root Mean Squared Error	32.04	31.97	31.97	32.10	29.68	28.63
Mean Absolute Error	25.59	25.55	25.54	24.32	22.38	23.17
R-Squared (R^2)	0.41	0.41	0.41	0.40	0.49	0.53
Explained Variance	0.67	0.67	0.67	0.63	0.65	0.77

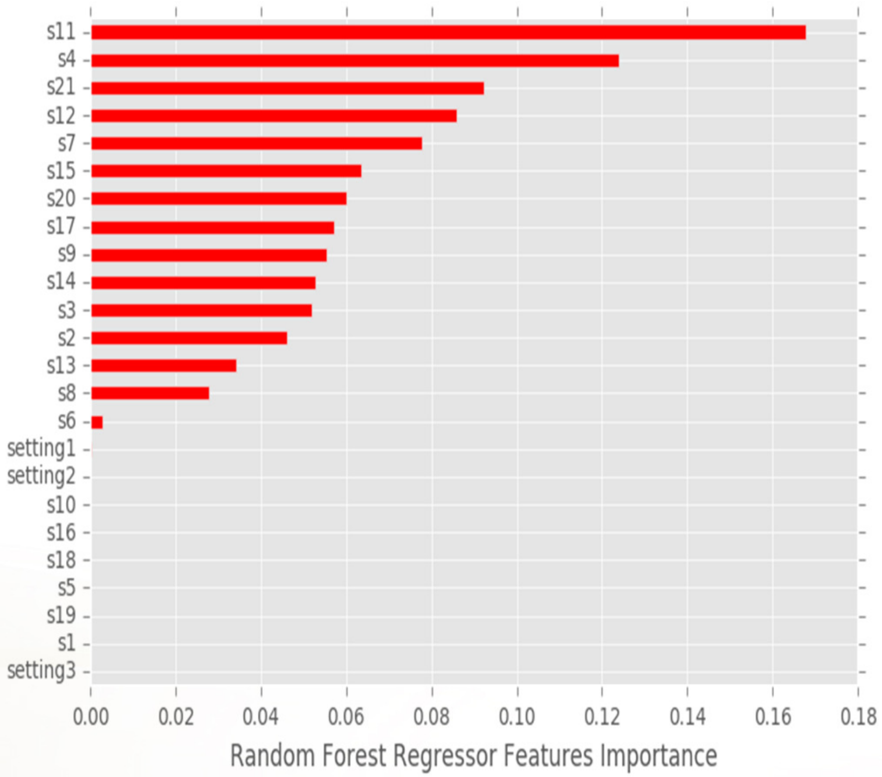
Random Forest Regressor is likely to perform better than other models

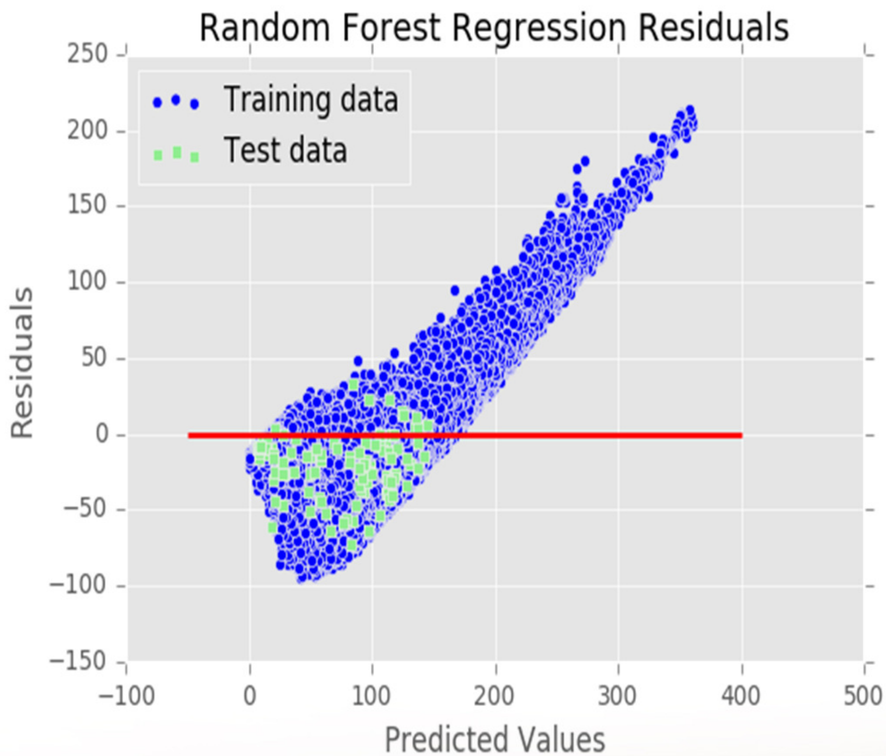
Random Forest Regressor predictions vs. actual for first 10 samples:

	0	1	2	3	4	5	6	7	8	9	10
Actual	112.00	98.00	69.00	82.00	91.00	93.00	91.00	95.00	111.00	96.00	97.00
Prediction	151.58	119.27	74.42	96.47	112.59	130.28	128.11	100.69	116.12	127.37	74.37
Difference	-39.58	-21.27	-5.42	-14.47	-21.59	-37.28	-37.11	-5.69	-5.12	-31.37	22.63

Regression Modeling

Random Forest Regressor: Feature Importance and Residuals





Features ranked based on their effectiveness in dividing training instances into decision tree branches.

Regression residuals are not randomly spread across the average value of the resid uals. A call for additional model/data tuning
.

Binary Classification

Used to predict if the engine will fail within specific cycles window or not

Machine learning algorithms used:

- ✓ Logistic Regression
- ✓ Decision Trees
- ✓ Support Vector Machines
- ✓ Linear Support Vector
- ✓ K Nearest Neighbors
- ✓ Gaussian Naive Bayes
- ✓ Random Forests

Classification metrics calculated:

- ✓ Area Under the Curve ROC (AUC ROC)
- ✓ Precision
- ✓ Recall
- ✓ F1 Score
- ✓ Accuracy

Binary Classification Results

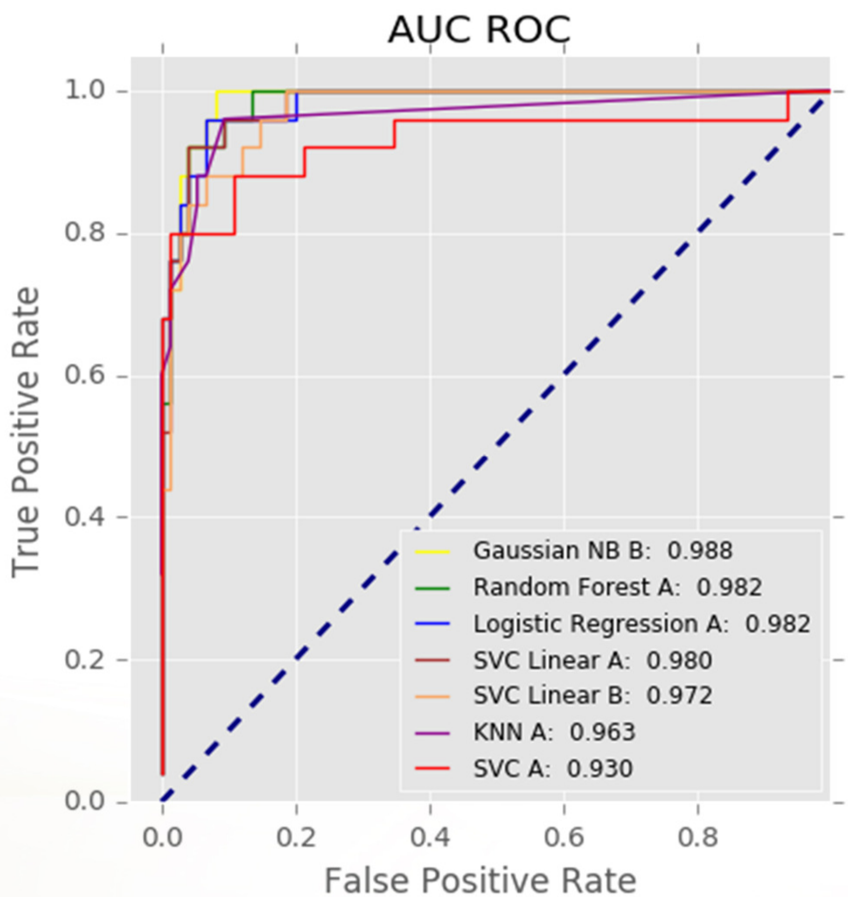
	Logistic Regression B	Logistic Regression A	Decision Tree B	Decision Tree A	Random Forest B	Random Forest A	SVC B	SVC A	SVC Linear B	SVC Linear A	KNN B	KNN A	Gaussian NB B	Gaussian NB A
ROC AUC	0.9803	0.9819	0.9451	0.9629	0.9803	0.9824	0.8917	0.9301	0.9717	0.9797	0.9352	0.9635	0.9877	0.9805
Precision	0.9333	1.0000	0.9333	0.9474	0.9444	0.9444	0.9444	0.9474	1.0000	0.4310	0.9444	0.9474	0.8276	0.8276
F1 Score	0.7000	0.8095	0.7000	0.8182	0.7907	0.7907	0.7907	0.8182	0.5714	0.6024	0.7907	0.8182	0.8889	0.8889
Accuracy	0.8800	0.9200	0.8800	0.9200	0.9100	0.9100	0.9100	0.9200	0.8500	0.6700	0.9100	0.9200	0.9400	0.9400
Recall	0.5600	0.6800	0.5600	0.7200	0.6800	0.6800	0.6800	0.7200	0.4000	1.0000	0.6800	0.7200	0.9600	0.9600

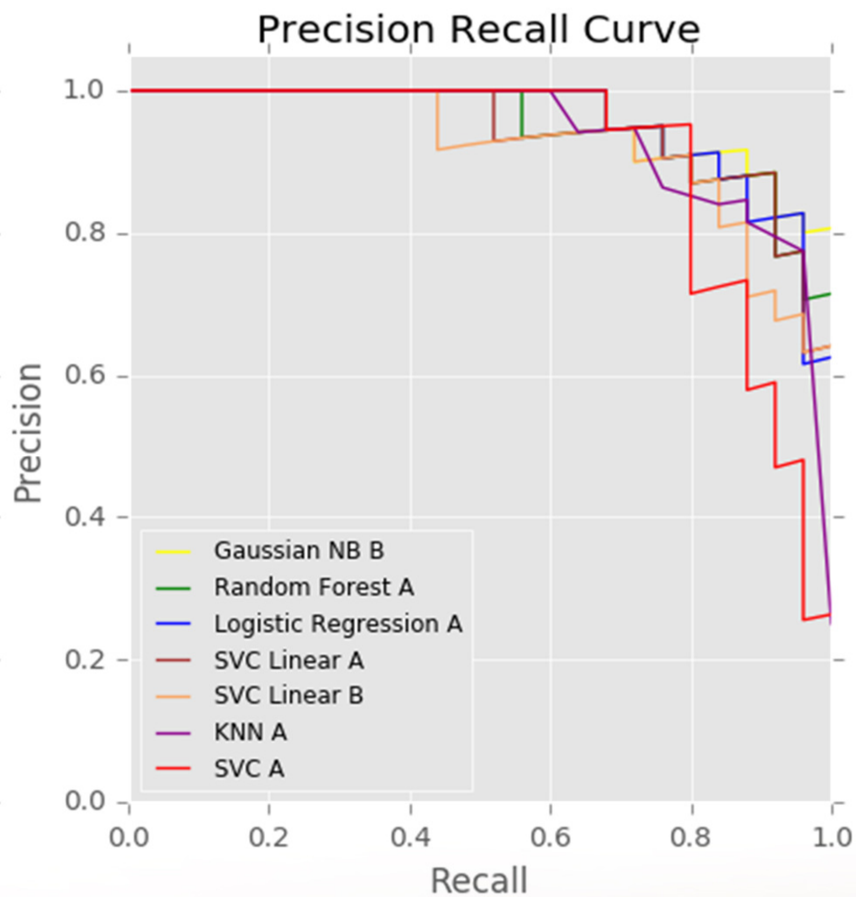
A: **A**fter Feature Engineering

B: **B**efore Feature Engineering

- ✓ Naïve Bayes and Random Forests scoring best results for AUC are likely to perform better than others
- ✓ Most of the binary classifiers showed better performance metrics with addition of new features
- ✓ Linear SVC has totally different metrics before and after FE, switching between Precision and Recall

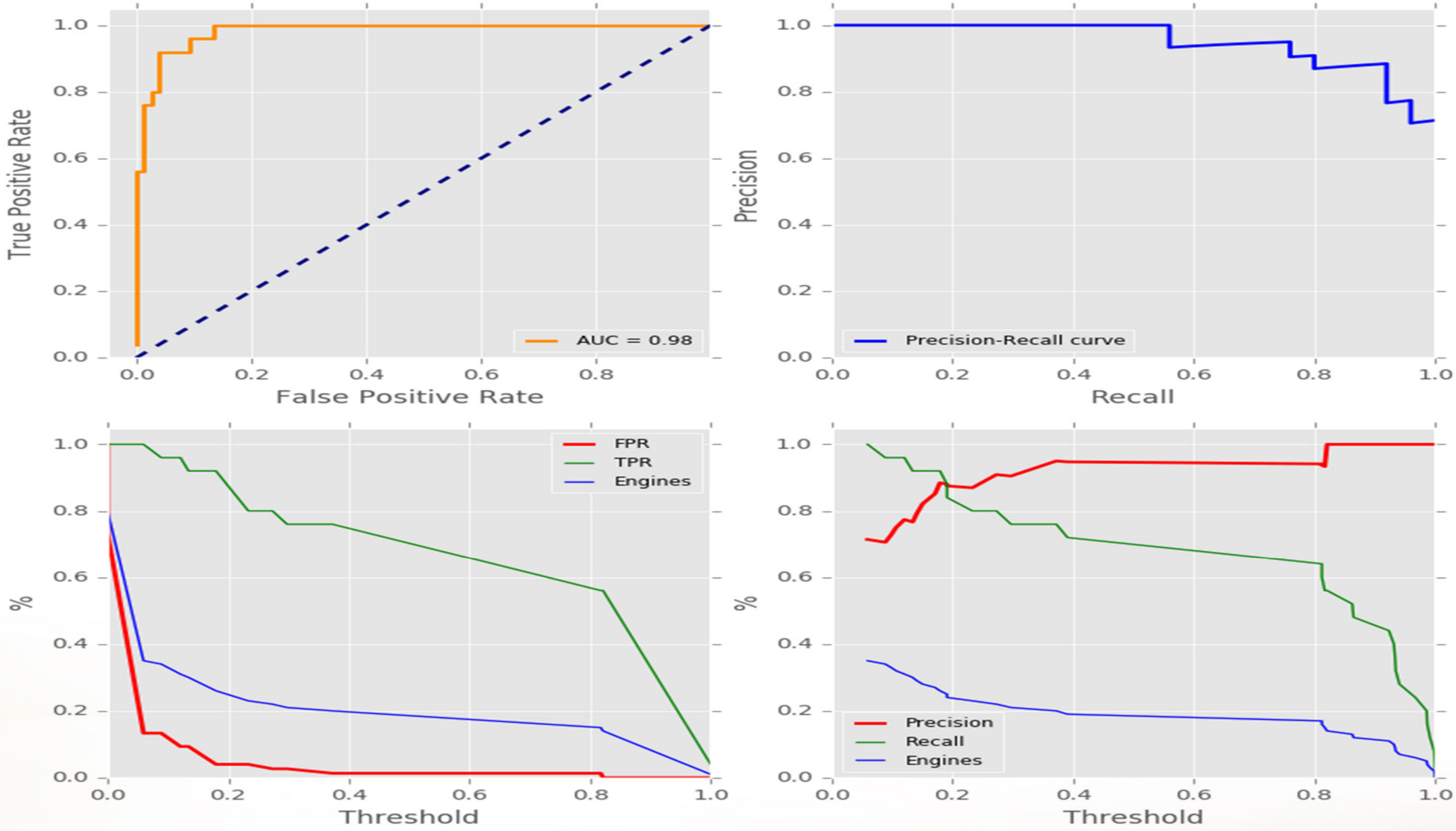
Binary Classification Results





Naïve Bayes and Random Forests have the best curves for AUC ROC and Precision-Recall

Binary Classification (Random Forest)



Threshold at different level facilitate maximum gain based on business capacity (number of engines that could be inspected in specific period i.e. Queue)

Expected Profit

Rank	Profit	Model	Queue	Threshold	TP	FP	TN	FN	TPR	FPR	TNR	FNR
0	19.00	Gaussian NB B	0.31	0.09	25	0	69	6	1.00	0.08	0.92	1.00
1	18.69	Logistic Regression B	0.28	0.11	24	1	71	4	0.96	0.05	0.95	0.99
2	18.69	Gaussian NB A	0.28	0.97	24	1	71	4	0.96	0.05	0.95	0.99
3	17.70	Logistic Regression A	0.29	0.06	24	1	70	5	0.96	0.07	0.93	0.99
4	17.35	Random Forest A	0.26	0.18	23	2	72	3	0.92	0.04	0.96	0.97
5	17.35	SVC Linear A	0.26	0.78	23	2	72	3	0.92	0.04	0.96	0.97
6	17.00	Random Forest B	0.33	0.10	25	0	67	8	1.00	0.11	0.89	1.00
7	15.72	KNN A	0.31	0.08	24	1	68	7	0.96	0.09	0.91	0.99
8	13.05	SVC Linear B	0.27	(0.59)	22	3	70	5	0.88	0.07	0.93	0.96
9	12.16	SVC A	0.21	(0.23)	20	5	74	1	0.80	0.01	0.99	0.94
10	12.08	SVC B	0.28	(0.94)	22	3	69	6	0.88	0.08	0.92	0.96
11	10.70	KNN B	0.26	0.31	21	4	70	5	0.84	0.07	0.93	0.95
12	10.14	Decision Tree A	0.30	0.18	22	3	67	8	0.88	0.11	0.89	0.96
13	7.82	Decision Tree B	0.29	0.08	21	4	67	8	0.84	0.11	0.89	0.94

By assigning cost and benefit monetary value for False and True predictions respectively, algorithms have been ranked based on expected profit / engine calculations (for unconstrained queue):

Expected Value ***= Prop(+ve) x [TPR x benefit(TP) + FNR x cost(FN)] + Prob(-ve) x [TNR x benefit(TN) + FPR x cost(FP)]***

- ✓ 100 Test samples (positive class = 25 : negative class= 75):
- ✓ prob_positive = 0.25, prop_negative = 0.75, TPb = \$300, TNb = \$0, FPc = \$-100, FNc = \$-200
- ✓ These monetary values should be provided by business domain experts

Multiclass Classification

Used to predict in which cycles window will the engine fail

Machine learning algorithms used:

- ✓ Logistic Regression
- ✓ Decision Trees
- ✓ Linear Support Vector
- ✓ K Nearest Neighbors
- ✓ Gaussian Naive Bayes
- ✓ Random Forests
- ✓ Neural Networks

Classification metrics calculated:

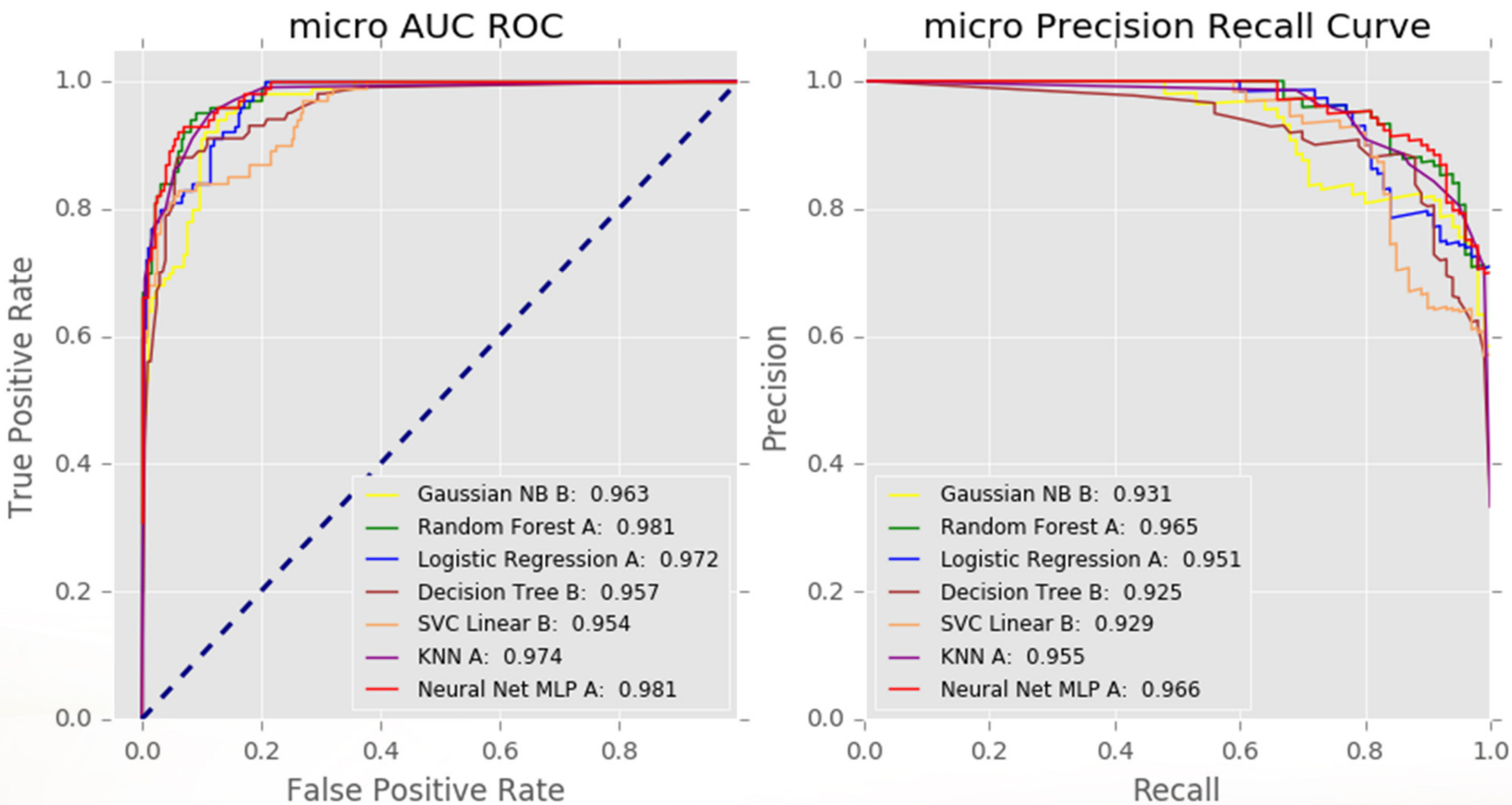
- ✓ AUC ROC – micro, macro
- ✓ Precision – micro, macro
- ✓ Recall – micro, macro
- ✓ F1 Score – micro, macro
- ✓ Accuracy

Multiclass Classification Results

	micro ROC AUC	micro F1	Accuracy	micro Precision	macro Recall	macro F1	micro Recall	macro ROC AUC	macro Precision
Logistic Regression B	0.9705	0.8438	0.8100	0.8804	0.5622	0.5579	0.8100	0.9452	0.5564
Logistic Regression A	0.9718	0.8526	0.8100	0.9000	0.5333	0.5517	0.8100	0.9415	0.5869
Decision Tree B	0.9566	0.8615	0.8400	0.8842	0.6689	0.6841	0.8400	0.9056	0.8190
Decision Tree A	0.9736	0.8571	0.8400	0.8750	0.6511	0.6079	0.8400	0.9499	0.8521
Random Forest B	0.9785	0.8542	0.8200	0.8913	0.5733	0.6125	0.8200	0.9643	0.7767
Random Forest A	0.9806	0.8673	0.8500	0.8854	0.6622	0.7058	0.8500	0.9677	0.8008
SVC Linear B	0.9537	0.8000	0.6800	0.9714	0.4178	0.4825	0.6800	0.9347	0.5949
SVC Linear A	0.9172	0.6301	0.0200	0.4792	0.7333	0.5011	0.9200	0.9433	0.6611
KNN B	0.9548	0.8557	0.8300	0.8830	0.5956	0.6417	0.8300	0.9049	0.8008
KNN A	0.9735	0.8731	0.8600	0.8866	0.6844	0.7099	0.8600	0.9499	0.7934
Gaussian NB B	0.9627	0.8520	0.7400	0.7724	0.9778	0.7579	0.9500	0.9503	0.6556
Gaussian NB A	0.9429	0.8493	0.7400	0.7815	0.9333	0.7550	0.9300	0.9448	0.6645
Neural Net MLP B	0.9833	0.8763	0.8500	0.9043	0.6689	0.7320	0.8500	0.9706	0.8736
Neural Net MLP A	0.9813	0.8990	0.8800	0.9082	0.7622	0.7981	0.8900	0.9709	0.8902

Neural Network and Random Forest classifiers are likely to perform better than other classifiers.

Multiclass Classification Results



Neural Network and Random Forest classifiers have the best curves for micro - AUC ROC and micro -.Precision Recall.

Next Steps

- ✓ Enhance regression modeling by model tuning, fixing data, or trying other models
- ✓ Perform features selection and dimensionality reduction techniques to enhance models performance metrics and speed
- ✓ Deploy selected models to be accessible online