Team 17 – Final Report (2760 words)

[1] Introduction - Motivation

The Project is being undertaken for Queensland Rail ("QR") to model the condition of rail ballast to better inform maintenance decisions. Based on QR's work, track geometry is used as an indicator of ballast condition. The Project enhances QR's current practice by applying interactive visualisations to the inspection of multiple datasets and machine learning (ML) methods to the prediction of work order requirements and rail condition.

Improvement in maintenance could generate QR savings of $5 million p.a with an additional reduction of ~$30 million in capital expenditure. Such savings could be extrapolated for rail networks globally. Additionally, on a global scale, safety improvements via reduced accidents may be possible [15, 16].

Figure 1 illustrates some key terms used throughout.

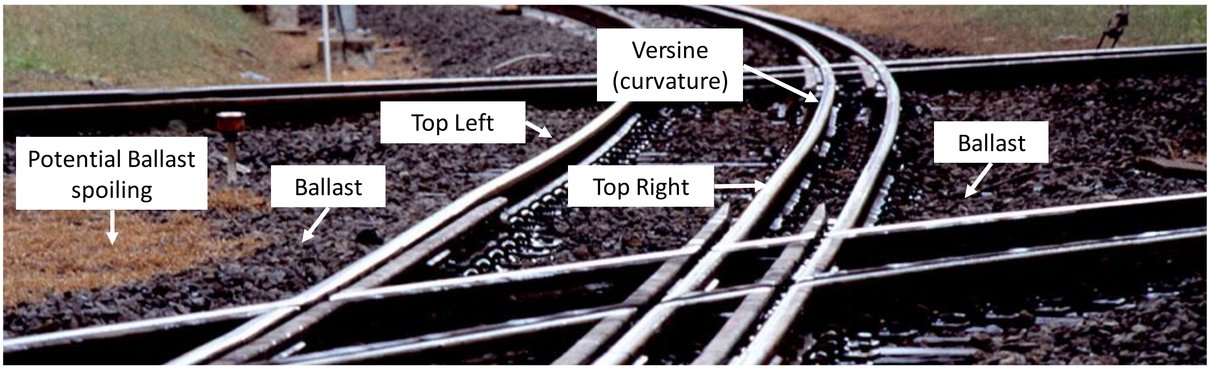


Figure 1: key terms used throughout the report[22]

[2] Problem Definition

Project scope includes:

1. Collation of raw data from various sources (e.g. ballast condition, location of drainage points etc) that may inform track condition.
2. Data manipulation:

    a. *Alignment of disparate features from multiple sources: Joining of* QR data, which separates work orders from Ground Penetrating Radar (GPR) and Track Recording Car (TRC) data respectively.

    b. *Identification of useful features, outliers and meaningful response variables:* Due to the large number of features, feature selection is imperative to prevent over-fitting. The project thus utilised LASSO regression and Linear Regression P-values to subset features. The advantage of LASSO over PCA[5] is the higher interpretability compared to eigenvectors in PCA. Previous studies on track recording data have conversely demonstrated utility of PCA [10]

3. *Applying ML algorithms to predict on collected data.* The current report focuses on predicting future track geometry and maintenance work.
4. *Interactive visualizations provide insights into QR's data that may assist decision making.*

[3] Survey Overview

Most research in ML for the rail industry relates to rollingstock and rail condition, not ballast or track geometry which is the focus of the Project.

Nakhaee et al[13], employ deep learning algorithms to detect structural defects in rail (not ballast). Several impediments were identified:

a. Class imbalance: as the vast majority (>99%) of rail is non-defective.
b. Availability of labelled datasets.
c. Model explainability.

Hajizadeh et al.[9] addressed the first two issues proposing minority over-sampling with noise to balance the labels thereby reducing bias. They also propose using semi-supervised techniques to counter the lack of labelled data.

[4] Intuition

The Project enhances current practices within QR:

- QR assesses ballast condition primarily via:
    - TRC data captured quarterly which measures rail geometry.
    - GPR collected every ~3 years.
    - As QR's network extends over 6,600 kms [17], the quantity of data collected includes 10 million TRC data points split across 16 features and ~350,000 GPR data points split across 22 features.

Since much of the current analysis undertaken by QR involves manual comparison of TRC and GPR data, it is highly labour and time intensive. The Project introduces interactive visualisations to improve the inspection process and ML to enhance predictive power.

The processes employed here extends current industry methods. Current research is focused on rail condition, whereas the Project is focussed on track geometry, a proxy for ballast quality. While Sharma[19] focussed on track geometry and utilised Markov Decision Processes to monitor maintenance actions, he only considered major defects that violate regulations. By contrast, the Project seeks to identify *any* degradation in ballast and *predict future ballast condition for all rail segments*.

Based on the survey of literature, it was believed the Project could be successful in utilising GPR data to predict rail geometry as:

- GPR is a proven technology for evaluating ballast condition [6,7,8,14]

- it has been demonstrated ~100% classification accuracy can be achieved using an SVM classifier [3,4] to detect soiling [18]
- fractal analysis on vertical TRC data has shown a moderate correlation with ballast fouling, indicating that TRC data is an indirect measure for ballast condition [1]

[5] Approach and Innovations

The project uses ML for enhanced predictions and interactive visualisations for efficient analysis and effective communication.

[5.1] Innovations

Key innovations include:

1. *Augmenting current "heatmap" processes:*
    a. Aligning TRC, GPR, work order and drainage data for comparable sections of track
    b. Automating the variance calculations to visually identify degradation in geometry
2. *Providing further insights by combining additional factors:* integrating additional attributes such as the location of drainage points and maintenance history.
3. *Enhancing decision-making through ML:* predicting track degradation over time contextualised to location to streamline decision-making.
4. *Improving user experience with interactive visualisations:* a network map for the entire state enables users to drill-down to the current status of a localised section of rail.

[5.2] Methodology

The project is planned in 6 phases (Table 1).

Process	Detail
1. Engage with QR	a. Understand current practice b. Identify opportunities for improvement c. Identify information required d. Execute non-disclosure documentation e. Obtain raw data f. Provide progress reports
2. Align features	a. Develop robust procedures to align disparate features (e.g. work orders, drainage points, TRC, GPR to track meterage)
3. Analyse data	a. Assess data and understand interrelationship between features b. Establish schema relating to features c. Finalise models to be developed d. Identify and process outliers/anomalies e. Undertake statistical analysis on the provided data
4. Build models	a. Develop various ML regression models that predict rail geometry based on features b. Develop classification models that predict future maintenance work c. Develop time-series models that extrapolate TRC features d. Perform testing and cross-validation of models e. Implement feature reduction as necessary f. Compare and select preferred models
5. Create visualisations	Several visualisations were developed as described in [5.2.1.4]. These include: a. "Heatmap" identifying ballast degradation b. Drilldown visualisation of rail geometry c. Predicted maintenance work required d. Geospatial view of GPR data e. Comparison of work order history and rail geometry measures f. Utilities to upload data and run models
6. Usability Assessment	a. Presentation of the Project to QR

Table 1. Work phases, tasks and progress.

[5.2.1] Algorithms Developed

[5.2.1.1] Features

Data manipulation and overview of available measures:

1. Data matching and joining was implemented using the pandas library. Due to overlapping distance measures, creation of the master data (Figure 2) involved segmenting into different track codes before joining on location along the track.

2. The work orders dataset required a translation from functional location to track codes prior to joining. An outline of this translation process was provided by QR and implemented in python by the team.

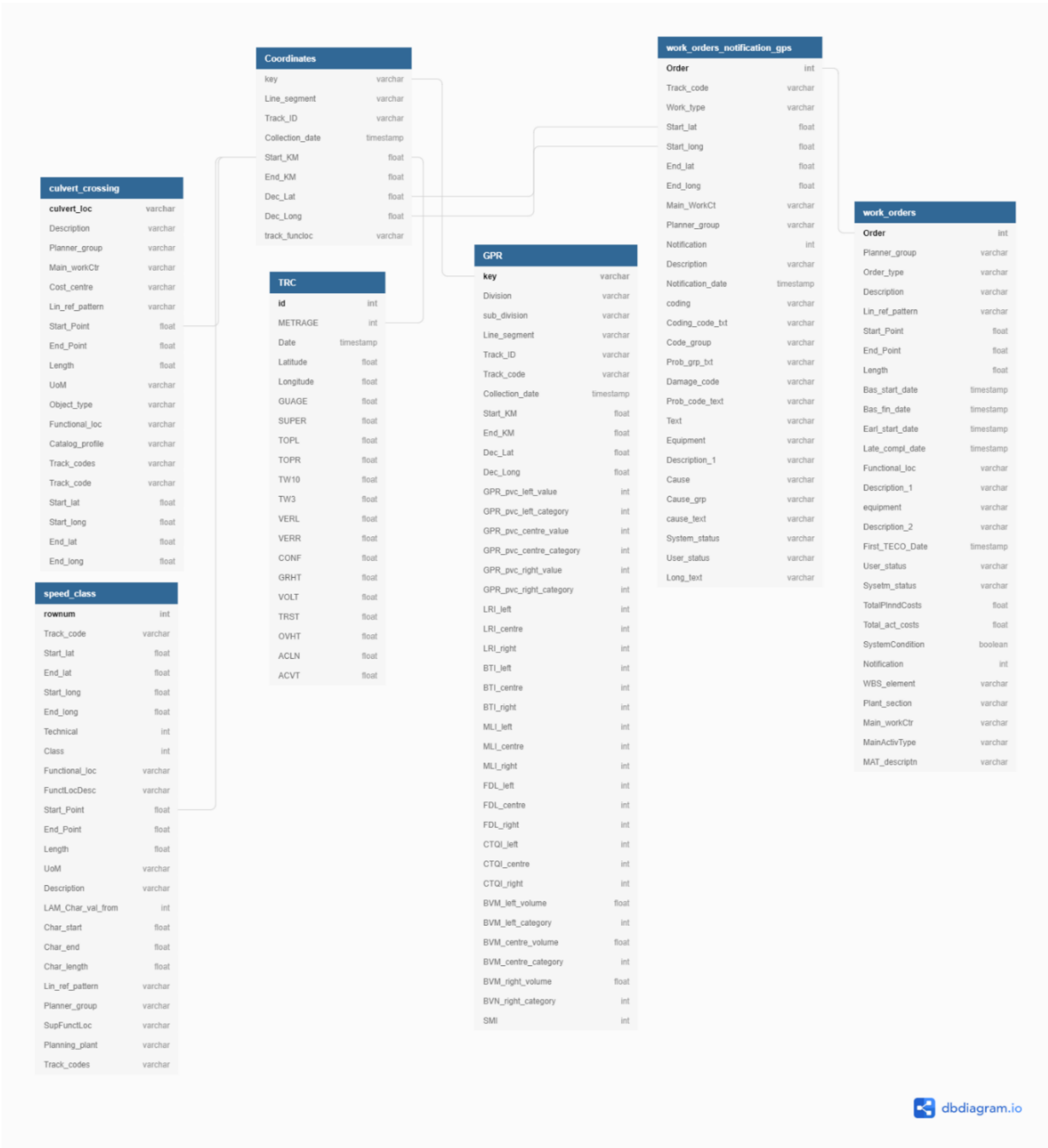


Figure 2. schema for the data provided by QR.

Appendix 2 elaborates on the pre-processing activities undertaken.

References to GPR and TRC data below mean:

- TRC for C195: 93 files representing 5 years of TRC history
- TRC for C138: 120 files representing 5 years of TRC history
- TRC for C139: 4 files representing 4 quarters of TRC history
- GPR for C138: for 2015 and 2018
- GPR for C195: for 2015
- GPR for C139: for 2018

Feature	Data Source	Expected Use
Ballast fouling Percentage Void Contamination (PVC) • Left • Centre • Right	• GPR Data	Correlated with ballast condition
Layer roughness Index (LRI) • Left • Centre • Right	• GPR Data	Assess feature importance
Ballast Thickness Index (BTI) • Left • Centre • Right	• GPR Data	Assess feature importance
Moisture Likelihood Index (MLI) • Left • Centre • Right	• GPR Data	Assess feature importance
Fouling Depth Layer (FDL) • Left • Centre • Right	• GPR Data	Assess feature importance
Ballast Volume Metric (BVM) • Left • Centre • Right	• GPR Data	Assess feature importance
Ballast Deficit Metric (BDM) • Left	• GPR Data	Assess feature importance

• Centre • Right		
Track Drainage Index (TDI) • Left • Right	• GPR Data •	Assess feature importance
Surface Mudspot Index (SMI)	• GPR Data	Assess feature importance
Rail Top Left	• TCR Data	Compute standard deviation over adjacent 5m segments
Rail Top Right	• TCR Data	Compute standard deviation over adjacent 5m segments
Rail Twist 10	• TCR Data	Compute standard deviation over adjacent 5m segments
Rail Twist 3	• TCR Data	Compute standard deviation over adjacent 5m segments
Rail Versine Left	• TCR Data	Compute standard deviation over adjacent 5m segments
Rail Versine Right	• TCR Data	Compute standard deviation over adjacent 5m segments
Drainage points	• Track Culvert and Level Crossing Data	
Maintenance history on track segment	• Work orders (C138, C195) • QR Translation Process • QR Track Code List • LRP Details • 4^{th} Level LRP to 3 digit TC • 5^{th} Level LRP to 4^{th} digit TC • Maintenance Codes	Correlated with ballast condition – must be controlled

Table 2: key features

[5.2.1.2] Response variables

Response variables to be predicted include:

Response Variable	Expected Use	Model Form
Combined rail geometry - future quarter(s)	Prediction of combined standard deviations for Top Left, Top Right and Twist 3	Regression
Maintenance requirement	Prediction on whether maintenance will be required on a segment of track	Classification
TRC features	Prediction of TRC features using historic values	Time-series

Table 3: response variables

[5.2.1.3] Statistical and ML Algorithms

The algorithms developed during the Project include:

Purpose	Algorithm	Source
1. Align TRC quarterly datasets	Minimise standard deviations of semi-static features	Custom developed
2. Estimate underlying distribution	Exponentiated Weibull Fit	stats; exponweib
3. Feature selection	LASSO	sklearn: linear_model.Lasso
4. Feature selection	OLS p-values	Stats; OLS
5. Feature selection	Elastic Net	sklearn: linear_model. ElasticNet
6. Regression prediction	Linear Regression	sklearn: linear_model. LinearRegression and statsmodels.api: sms
7. Regression prediction	Random Forest Regression	sklearn.ensemble: RandomForestRegressor
8. Regression prediction	Support Vector Regression – various kernels	sklearn.svm: SVR
9. Regression prediction	K-NN Regression	sklearn; neighbors; model_selection.GridSearchCV
10. Regression prediction	Artificial Neural Network	keras.wrappers.scikit_learn ; KerasRegressor; keras.layers; Dense ; keras.models ; Sequential
11. Regression prediction	Artificial Neural Network (with early stopping)	keras.models ; Sequential; keras.callbacks EarlyStopping, ModelCheckpoint
12. Regression prediction	Multilayer Perceptron	sklearn.neural_network; MLPRegressor
13. Feature transformation	Transformation of time-series variables to coexist with semi-static variable	Custom developed
14. Classification prediction of upcoming maintenance work	Logistic Regression	sklearn; linear_model.LogisticRegression
15. Classification prediction of upcoming maintenance work	Support Vector Machine	sklearn.svm; SVC, GridSearchCV
16. Classification prediction of upcoming maintenance work	Random Forest Classification	sklearn.ensemble; RandomForestClassifier; GridSearchCV

17. Regression prediction of quarterly maintenance work count (zone-wise).	Random Forest Regression	sklearn.ensemble: RandomForestRegressor
18. Regression prediction of TRC metric **combined**	Random Forest Regression	sklearn.ensemble: RandomForestRegressor
19. Multi-processor Feature Transformation	Aggregation of time-series data into zones and quarters; Transformation of time values as row indices to columns to be used as features	Custom developed (used parallel processing to speed up computation; 16X speed up for a 16-core machine)

Table 4: algorithms developed for project

[5.2.2] User Interfaces Developed

Interactive displays of data and predictions derived from the ML models are presented using the Django framework, with embedded plots from Bokeh, a python visualization library. These are hosted on an Amazon Elastic Compute Cloud (Amazon-EC2) instance [21].

The main webpage (Figure 3) provides users the ability to:

1. Interact with existing GPR data to gain insight from visualisations (Figure 3B)
2. Interact with existing calculated track geometry calculated using TRC data (Figure 3A, Figure 5)
3. Execute a random forest ML model on user-input data to generate predictions of upcoming maintenance work (Figure 3C) .
4. Visualise predictions in a geospatial context (Figure 3D)
5. Interact with work order data to identify the impact maintenance has on rail geometry
6. Review the results of analysis undertaken by the Project (Figure 4)
7. Predict number of zone-wise work orders required for the upcoming 1-4 quarters.

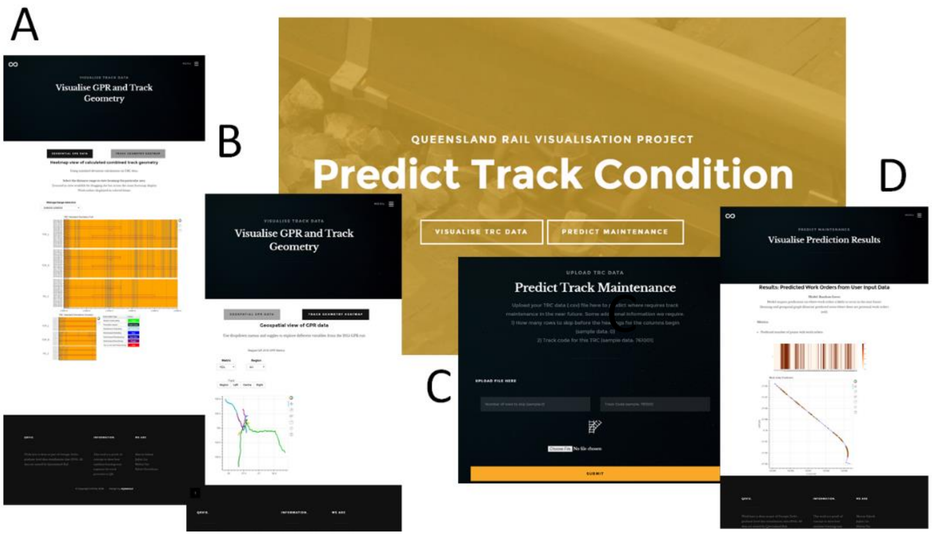


Figure 3. Developed web application views hosted on [aws](). Web application consists of 2 main functions. 1) visualising GPR data in an interactive geospatial graph, and calculated TRC combined metric with overlays of work orders in an interactive heatmap. 2) predicting upcoming track maintenance using user-input data to generate visualisations from a random forest output.

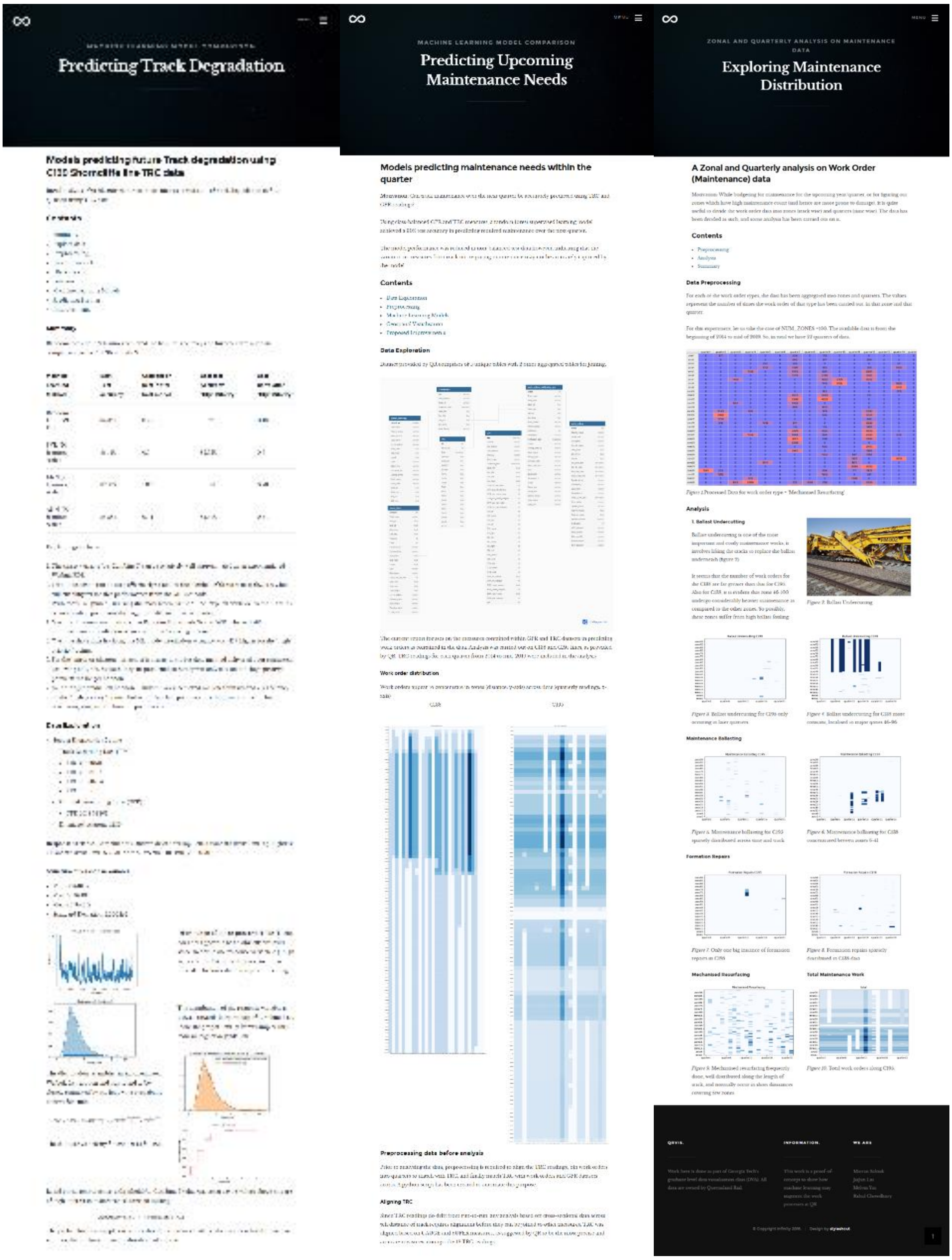
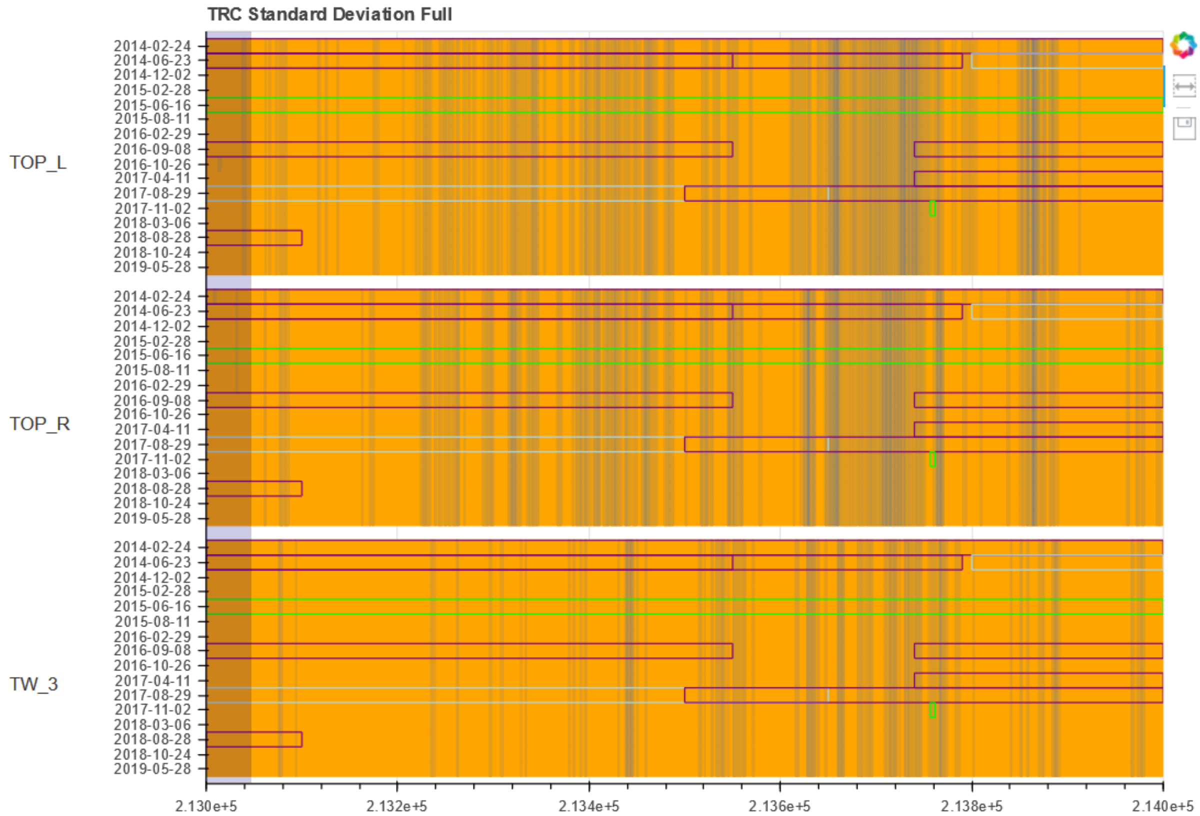
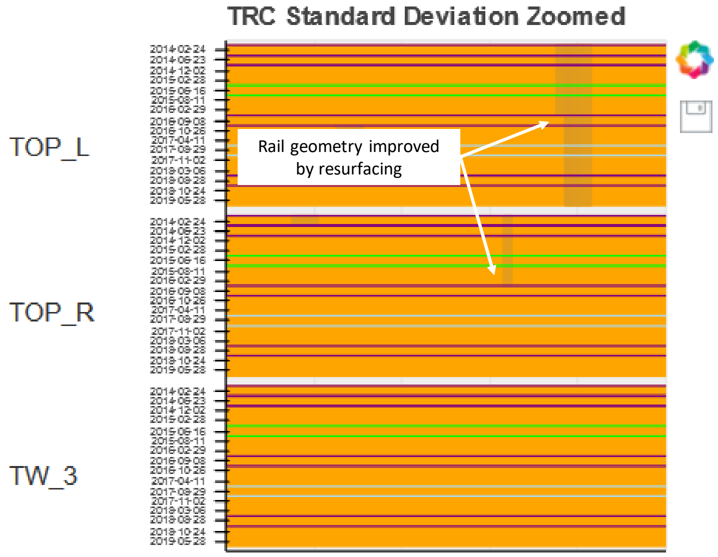


Figure 4: Reports hosted on the web application to explain the models and steps behind preparation of the techniques used within the app.





Work Order Type	Colour
Ballast Undercutting	Green
Formation repairs	Dark Green
Maintenance Ballasting	Light Blue
Mechanised Ballasting	Blue
Mechanised Resleepering	Dark Blue
Mechanised Resurfacing	Purple
Top & Line Spot Resurfacing	Red

Figure 5: interactive drill-down allowing inspection of maintenance impact. Sliding the grey view-bar changes the zoomed-in area of track shown, while the dropdown box selects the section of track presented, denominated in kilometres.

Interactivity is enhanced via:

1. Ability to select specific geographic segments to drill-down (Figure 6)
2. Utilities that enable uploading of new data to be visualised (Figure 5)

3. Functionality enabling ML and time-series algorithms to be run on selected datasets in real-time

The visualisations implemented (Figure 5) were designed following consultation with QR engineers and management with a view to enhancing their future decision making. The visualisations were made interactive using custom Javascript call-backs in Bokeh.

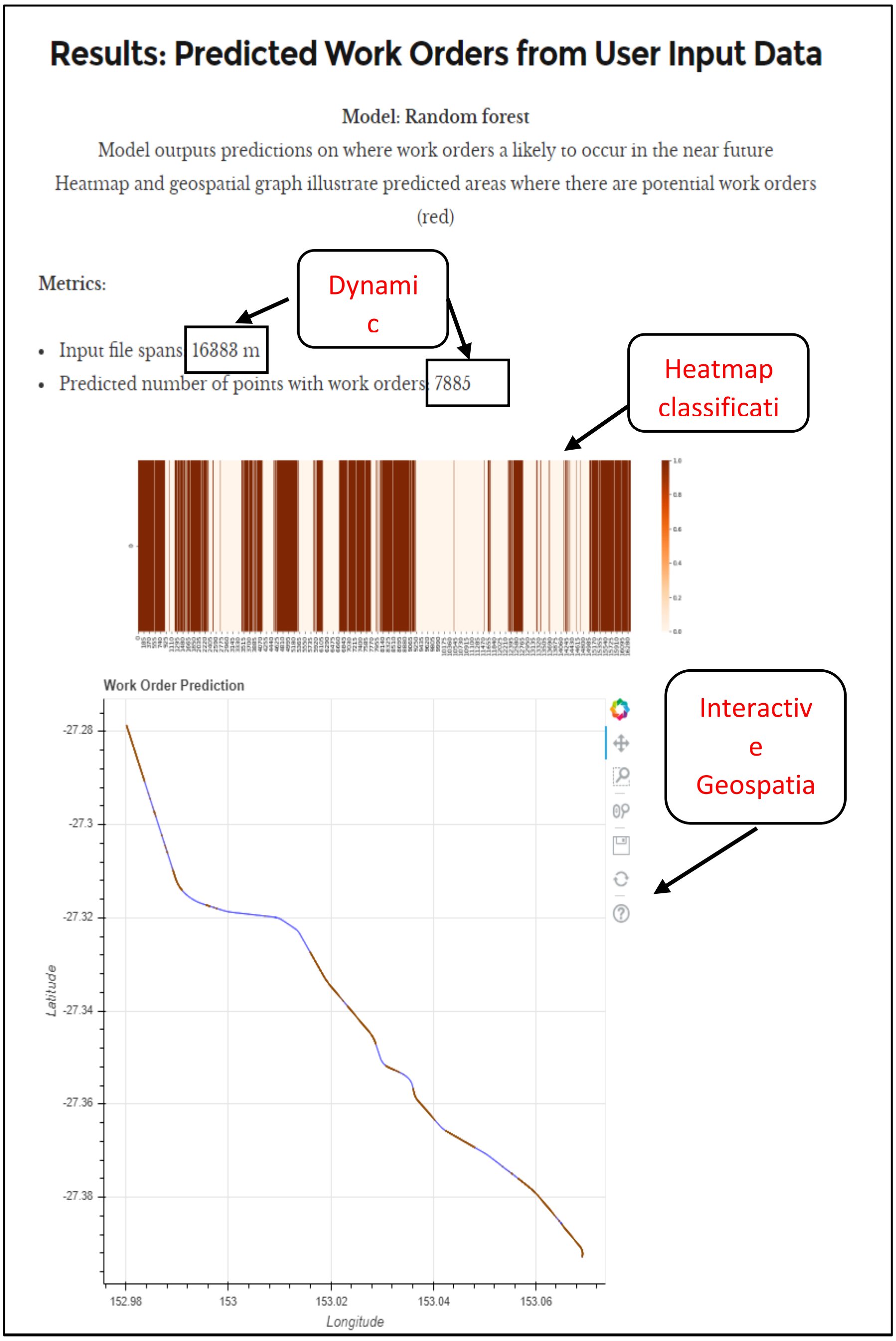


Figure 6: utility enabling predictions from upload of new feature data. Results page displays dynamic statistics about the uploaded file, heatmap classification of the data and an interactive geospatial view of the predictions.

[6] Testbed Description

The key questions addressed during experimentation include:

Key Questions	Considerations
Most meaningful response variables	Which variables best serve as the dependent variables: • Track geometry measures (i.e. "Combined") • Maintenance requirement
Dataset alignment and statistical analysis	• Can different datasets be effectively aligned using semi-static features? • Can the distribution of the target variable be effectively estimated?
Baseline prediction accuracy	• What is the baseline predictive performance of the non-ML methods (i.e. what could QR expect to achieve without applying ML)?
Validity of features	• Which features explain variance in the response variables? • Are there highly correlated predictors that need to be removed? • Should feature reduction be performed (e.g. LASSO)?
Predictive power of the models	• Do the ML models provide any predictive benefit over the baseline? • What is the test/cross validation errors of the models?
Preferred model(s)	• Which model is preferred wrt predictive power? • Which model is preferred wrt communicability (e.g. to management)?
Feature transformation	• Can/should features in the prediction datasets be transformed to improve predictive accuracy?
Longer-horizon predictions	• Are the models effective in making predictions in longer-term horizons? • Do the ML models outperform baseline predictions in longer-term horizons?
Usability of visualisations	• Are the visualisations meaningful to end-users wrt decision making? • Are the visualisations usable and "attractive" to users?
Future improvements	• What additional improvements/extensions could be made in the future?

Table 5: testbed description

[7] Experiments and Observations

[7.1] Overall data pre-processing

[7.1.1] Alignment of TRC data

Around 200 TRC files were aligned and joined by locating the meterage offset that minimised differences in standard deviations across the semi-static features "Gauge" and "Super". Once aligned, the datasets were joined with other feature sets such as GPR, drainage points and maintenance history.

Figure 7 illustrate clear continuity in rail geometry across time periods, suggesting robust alignment.

Missing value imputation was accomplished by filling with '0' value.

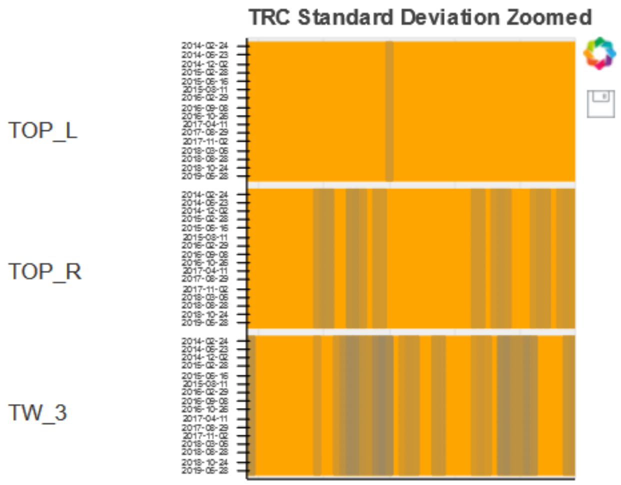


Figure 7: aligned TRC datasets using custom python utility provided in application

[7.1.2] Combined TRC variable calculation and analysis

One key response variable is "Combined" which reflects a linear combination of track geometry features calculated by the TRC:

$$Combined \equiv \frac{Std.DEV(TopLeft) + Std.Dev(TopRight)}{2} + Std.DEV(Twist3)$$

The combined metric was used by QR engineers as an indicator of **track geometry**, most associated with ballast condition.

The distribution of track geometry indicates no apparent structure associated with the response variable (figure 8).

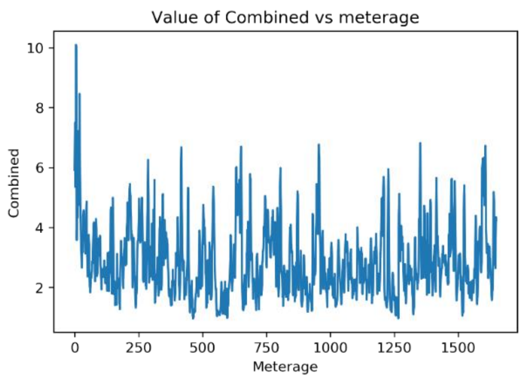
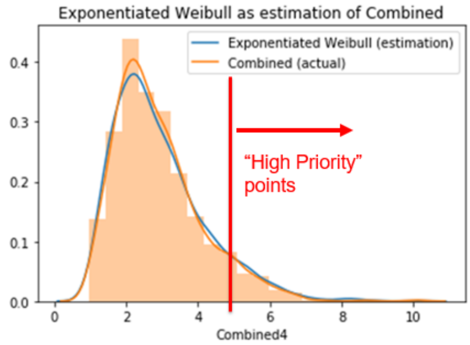


Figure 8: high variance seen in the Combined metric

As shown in Figure 9, an exponentiated Weibull distribution probability density function (pdf) fits the combined variable closely.

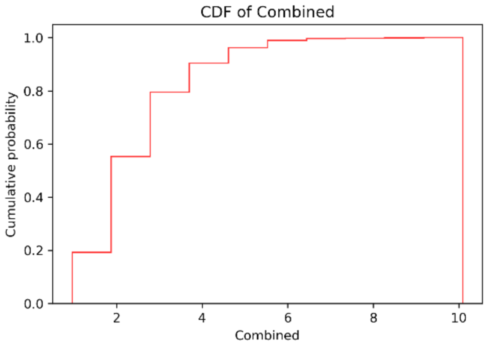
Based on current practice, a 4.9 threshold value for track geometry, where values above this are of high interest in maintenance decision-making ("high priority").

The significance of this analysis: i) the combined variable follows a similar distribution to part failure in common maintenance settings, and
ii) based on the cumulative distribution function (*cdf;* Figure 10), only 7.64% of samples was labelled "high priority".



$$f(x) = 1.750(x/1.449)^{0.208}\left[1-e^{-(\frac{x}{1.449})^{1.208}}\right]^{1.099} e^{-(\frac{x}{1.449})^{1.208}}$$

Figure 9: response variable estimated PDF



$$\text{"High Priority" points} = \{Combined_i(t) > 4.9\}$$

Figure 10: response variable estimated CDF

As the dataset is highly biased, a trivial classifier that only predicts {Combined < threshold} would

[7.2] Experiment 1: Predicting combined variable using historic values (Track: C195)

Hypothesis: historic track geometry results may contain information about future values 1 quarter ahead.

[7.2.1] Features and Response Variable

The TRC data for C195 contains 81 unique. The metric value taken on the last date has been selected as the response variable, while the remaining 80 dates were features for the model.

[7.2.2] Data Pre-processing and Train/Test Split

The total number of datapoints for TRC C195 are ~3.1 million.

Using the calculated combined variable, the values from each date are collated into one row in the processed dataset. The final data contains 154900 points (figure 11):

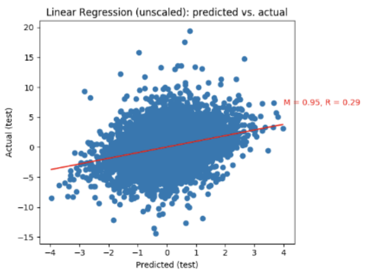
	2014-01-31	2014-02-11	2014-02-14	2014-05-16	2014-06-04	2014-10-04	2014-10-31	2015-01-20	2015-02-05
26.114	0.85000	0.00000	1.85000	-0.50000	0.00000	-1.35000	0.00000	-0.10000	7.65000
26.115	1.15000	0.00000	0.15000	1.35000	0.00000	5.35000	0.00000	-0.10000	3.45000
26.116	0.30000	0.00000	-1.20000	0.60000	0.00000	5.95000	0.00000	-1.35000	2.50000
26.117	-0.50000	0.00000	-2.40000	1.75000	0.00000	0.95000	0.00000	0.15000	0.95000
26.118	-0.55000	0.00000	-1.40000	-0.10000	0.00000	1.40000	0.00000	2.20000	-0.40000
26.119	-1.55000	0.00000	0.45000	-1.85000	0.00000	-0.40000	0.00000	1.65000	-1.55000
26.12	-3.85000	0.00000	2.30000	-2.75000	0.00000	-0.90000	0.00000	-2.60000	-3.00000
26.121	-3.85000	0.00000	2.05000	-3.25000	0.00000	-2.65000	0.00000	-2.60000	-2.10000
26.122	0.35000	0.00000	2.05000	-0.50000	0.00000	-3.75000	0.00000	-1.75000	-0.40000
26.123	1.00000	0.00000	-2.00000	4.35000	0.00000	-2.55000	0.00000	0.55000	0.55000
26.124	0.45000	0.00000	-2.15000	5.75000	0.00000	-0.20000	0.00000	0.80000	-2.30000
26.125	1.40000	0.00000	5.50000	2.30000	0.00000	1.75000	0.00000	0.80000	-2.45000
26.126	2.05000	0.00000	7.60000	-3.30000	0.00000	0.80000	0.00000	-3.90000	1.75000
26.127	1.80000	0.00000	6.65000	-5.00000	0.00000	-4.90000	0.00000	-4.90000	0.80000
26.128	-0.20000	0.00000	2.45000	-2.75000	0.00000	-3.05000	0.00000	-2.60000	0.80000
26.129	-2.40000	0.00000	-3.20000	-2.95000	0.00000	0.80000	0.00000	1.00000	-1.45000
26.13	0.25000	0.00000	-8.55000	-2.80000	0.00000	1.35000	0.00000	4.20000	-1.55000
26.131	0.25000	0.00000	-1.00000	-2.15000	0.00000	1.70000	0.00000	4.20000	1.65000
26.132	4.40000	0.00000	1.45000	1.05000	0.00000	2.00000	0.00000	3.25000	3.20000

Figure 11 Pre-processed C195 dataset with Combined track data for each quarter

[7.2.3] ML Models

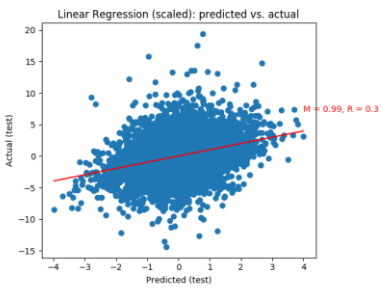
Linear regression

Both scaled and unscaled data was using for Linear regression. The scaled model achieved slightly better accuracy. However, both models suffered from low coefficient of determination (Figure 12, 13).



Metric	**Score**
Coefficient of determination	0.1036
accuracy	95.14 %

Figure 12. Scatterplot for Unscaled Linear Regression, and Evaluation metrics

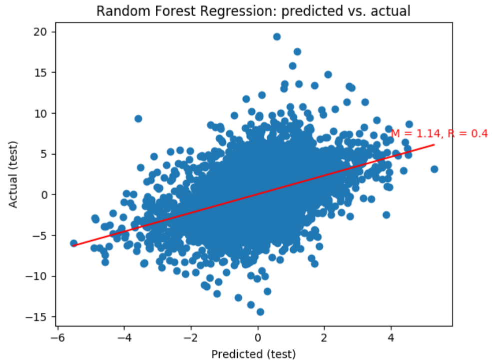


Metric	**Score**
Coefficient of determination	0.1036
accuracy	95.5 %

Figure 13. Scatterplot for Scaled Linear Regression, and Evaluation metrics

Random Forest Regression

A Random Forest Regressor, with hyperparameter num_trees = 1000 (after experimenting with values from 10,20, 50, 100,1000 and 10000) was used to predict (Figure 14).



Metric	Score
RF accuracy	97.53 %
RF test score	0.161
Out of bag score	0.1696

Figure 14. Scatterplot for Random Forest Regressor, and Evaluation metrics

[7.3.4] Predicting "high priority" points

As "high priority" points alert QR engineers where ballast degradation is most likely, predictions were made on these points for each ML model. Random Forest achieved the best model performance (accuracy: 82.81%, R^2: -0.081, OOB: 0.0041). This result suggests potential utility in QR's track maintenance decision making for 1 quarter ahead.

[7.3.4] Summary and Further Work

The models had moderately high accuracy, but low R^2 score due to inability to explain the variance around the mean. This approach requires extensive testing on other models such as ANN, LSTM (since it is of time series nature) to provide useful and accurate results that can be used for filling gaps left by the TRC runs.

[7.3] Experiment 2: Feature selection and Model comparison (Track: C139)

Hypothesis: Feature selection may further improve predictions on track geometry.

[7.3.1] Baseline models achieve up to 80.92% accuracy in predicting track geometry 1-quarter ahead

To set a baseline, the most trivial regression model: $Combined(t) = mean(Combined(t - 1))$, achieved 63.86% test accuracy. By projecting the most recent quarter Combined value i.e. $Combined_i(t) = Combined_i(t-1)$. The baseline achieved and improved 80.92% test accuracy. Note heteroskedasticity in Figure 16 for "high priority" points.

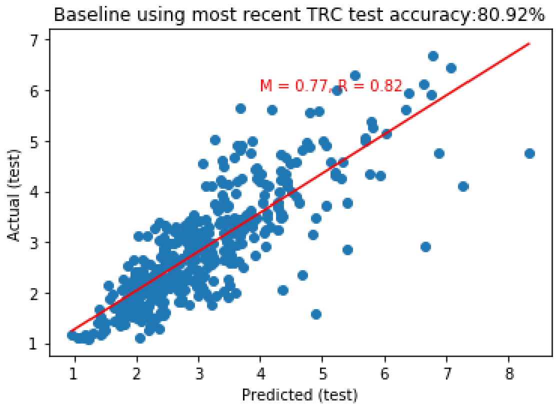


Figure 15: Baseline predictive accuracy

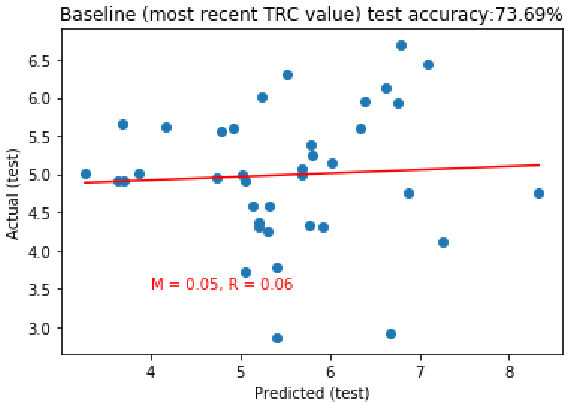


Figure 16: Baseline prediction "high priority"

[7.2.2] Feature assessment reveals redundancy in measures

The feature selection results are summarised as:

Feature Selection Method	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
Linear regression (using coefficient p-values)	83.05%	0.98	80.52%	0.18
LASSO	81.78%	1.17	80.27%	0.22
Elastic Net[1]	83.21%	1.01	80.74%	0.16

Table 6: feature engineering results. Note 1: not technically a feature selection method as includes coefficient shrinkage

Feature selection summary performance:

Objective	Performance
Test accuracy	~2% improvement
Gradient line-of-best-fit	High improvement
Test accuracy "High Priority" points	~6% improvement
"High Priority" prediction correlation	Moderate improvement

Table 7: feature selection summary of performance

[7.2.3] Assessing ML models designed to predict track geometry 1 quarter ahead

The ML results are summarised as:

ML Method	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
Random Forest (9 features)	84.35%	0.99	86.2%	0.48
SVR (20 features, scaled)	85.14%	0.9	85.25%	0.2
KNN (9 features, scaled)	82.21%	1.02	81.64%	0.23
ANN (20 features, scaled)	85.88%	0.94	85.54%	0.34

Table 8: ML summary of performance

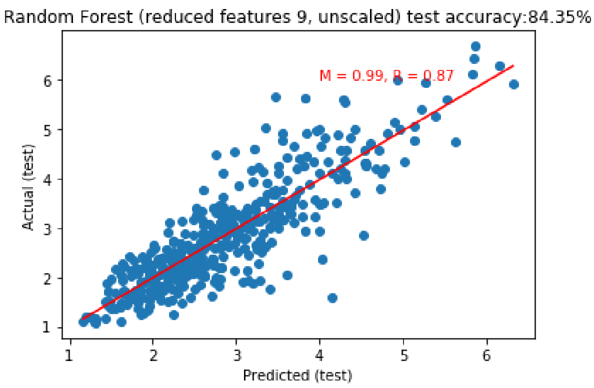


Figure 17: Random Forest Regression

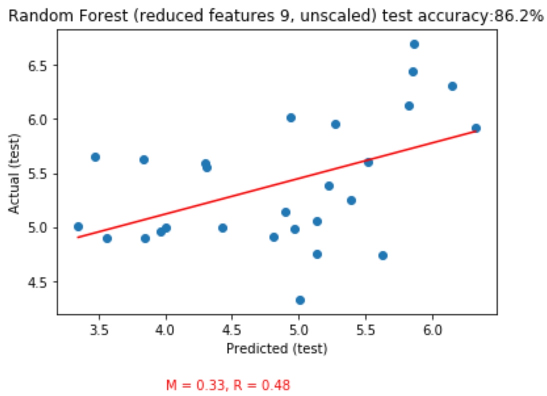


Figure 18: RF Regression "high priority"

Random Forest Regression (Figure 17, 18) and Sequential ANN achieved improved prediction accuracy on the "high priority" points compared to baseline.

The network architecture of the ANN comprised 6 hidden layers using the Adam optimiser. Note that implementing early stopping for the Sequential ANN (Figure 18) was key to optimising performance.

The Random Forest was fitted on 1,000 trees with features$^{1/2}$ random features.

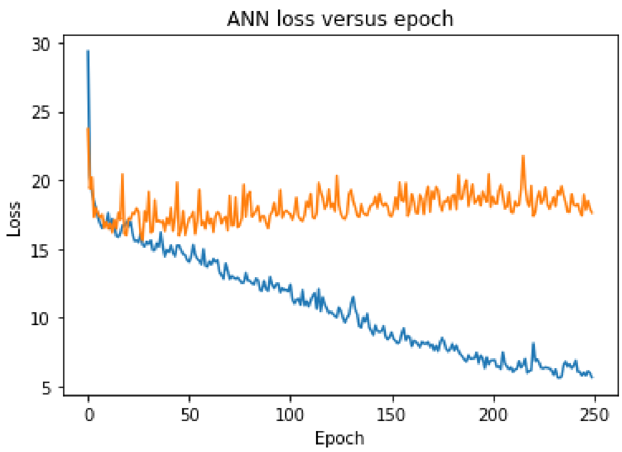


Figure 19: importance of early stopping in ANN

The results of the ML experiments predicting future-quarter track geometry are summarised below:

Objective	Performance
Test accuracy	~3% improvement
Gradient line-of-best-fit	High improvement
Test accuracy “High Priority” points	~12% improvement
“High Priority” prediction correlation	High improvement

Table 9: ML summary of performance

Additional experiments and observations are outlined in Appendix 3.

[7.3.2] Assessing ML models designed to predict track geometry 2 quarters ahead

Predictions were made for a 2-quarter time horizon. The baseline performance was compared with Random Forest regression. The ML method clearly outperformed the baseline in the longer time horizon.

Additionally, Random Forest produced significantly higher test accuracy on the “high priority” predictions.

Prediction Method	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy “High Priority”	Best Correlation “High Priority”
Baseline	51.86%	0.23	28.78%	-0.56
Random Forest[2]	74.19%	1.29	70.93%	0.15

Table 10: RF Regression outperformed the baseline for 2-quarter horizon. Note 2: Random Forest used 9 features, unscaled data.

Unlike the baseline method, Random Forest Regression produced predictions for 2 future quarters that could be useful (Figure 20).

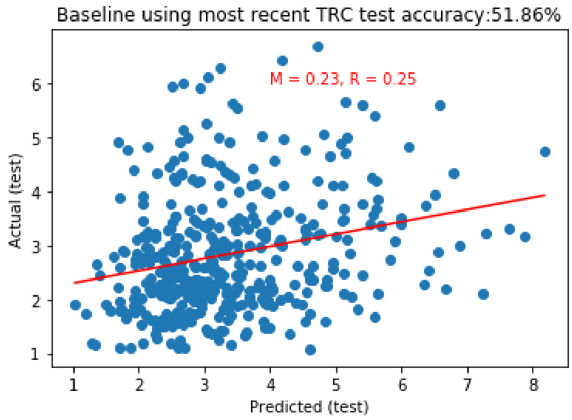
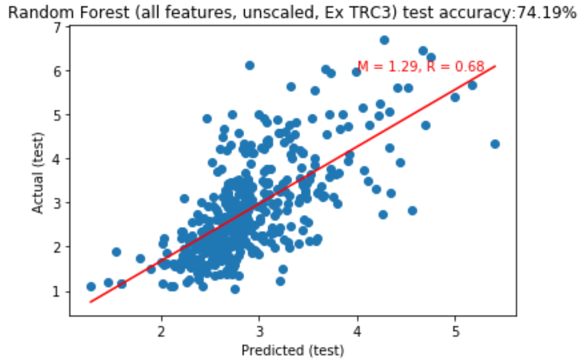
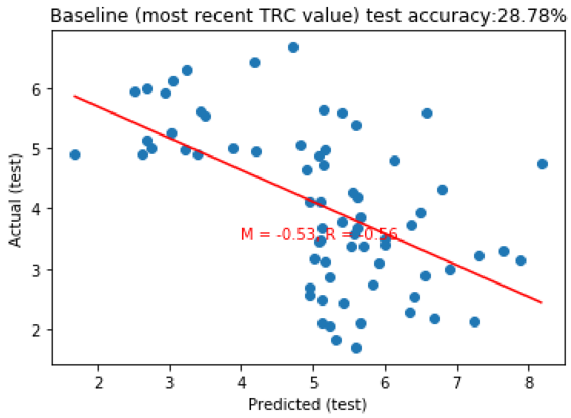
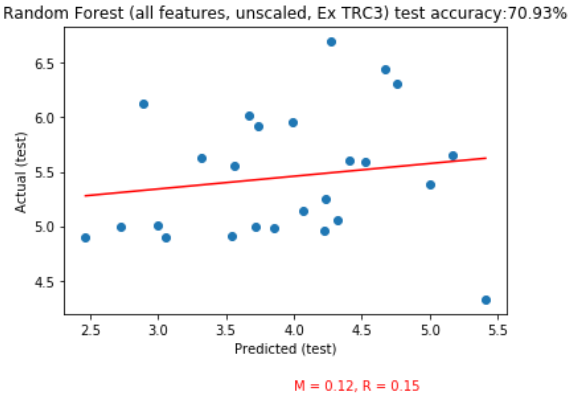
Baseline prediction: $Combined_i(t) = Combined_i(t-1)$	**Random Forest Regression (9 features)**
	
	

Figure 20: RF Regression outperforms baseline prediction in 2-quarter horizon

[7.3.3] Summary and future work

The results indicate a potential for Random Forest to be applied to predicting track condition and supplement maintenance decision making up to 2 quarters ahead.

However, the high accuracy and low correlation may relate to relatively low variance in the target variable enabling high accuracy despite the low correlation. This should be further investigated.

[7.4] Experiment 3: Use of Feature transformation

As observed in 7.3.2, the datasets contain redundancy in features, a common issue with time-series data. Thus, further exploration of feature transformation was conducted.

[7.4.1] Feature transformation of TRC data

The ML Regression models used features sourced from different time horizons: GPR (12-months old), TRC (quarterly) and drainage points (relatively fixed). To assess the impact of this, the TRC features were transformed:

i. the most recent TRC was retained to preserve currently known information regarding rail geometry, and

ii. derived features intended to capture the rate of change over prior TRC runs, calculated as:

$$\Delta Top\ Left_i = \{1.0[\sigma Top\ Left_i(t) - \sigma Top\ Left_i(t-1)] + \gamma[\sigma Top\ Left_i(t-1) - \sigma Top\ Left_i(t-2)] + \gamma^2[..]\}/(1 + \gamma + \gamma^2 + ..)$$

where σ refers to the standard deviation across 5 metres for the feature and γ is a decay coefficient reducing the impact of historic rates of change.

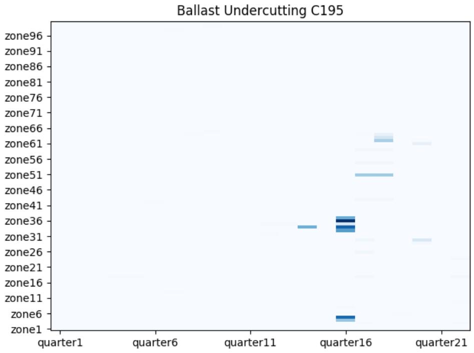
Results of the feature transformation experiments:

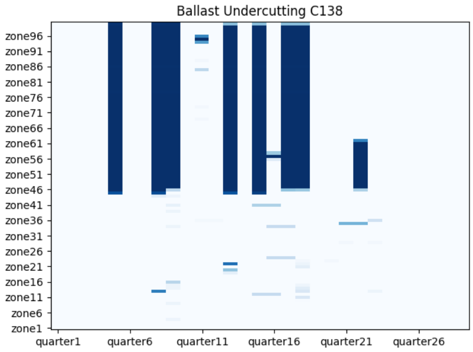
1. the most useful rate of change features used γ=0 i.e. using only the most recent rate of change.
2. the ML models did not demonstrate improved performance using this feature transformation.

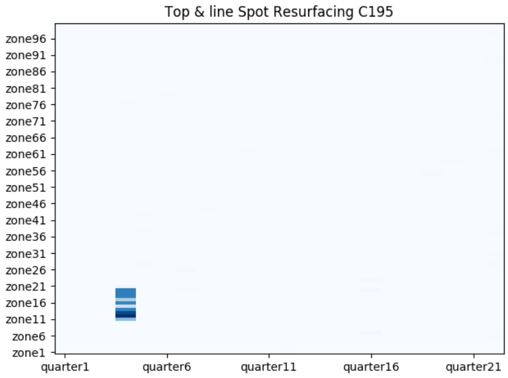
[7.4.2] Feature transformation of maintenance data by binning into zones and quarters

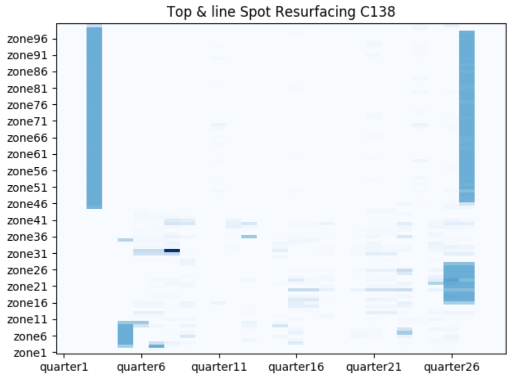
Binning of maintenance work for C138 and C195 data into zones and columns revealed trends in maintenance distribution (Figure 21).

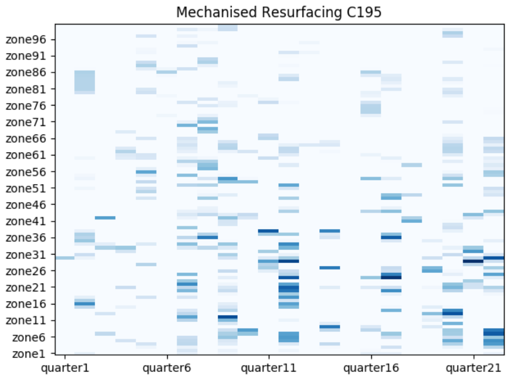
The project team has written these tools in Python which use parallel processing to quickly perform this feature transformation. Future usage of this may include real-time experimentation on large track data covering the full QR network (millions of rows).

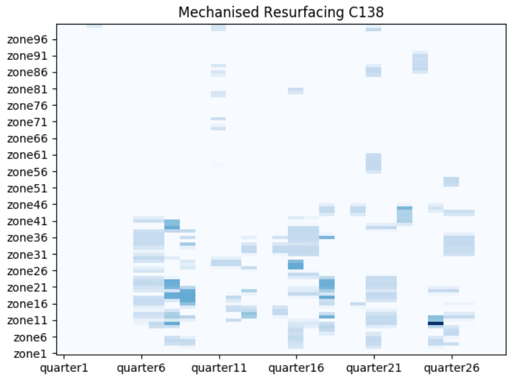

Ballast Undercutting C195
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21

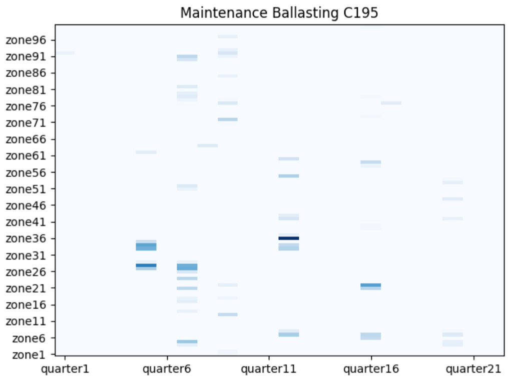

Ballast Undercutting C138
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21
quarter26

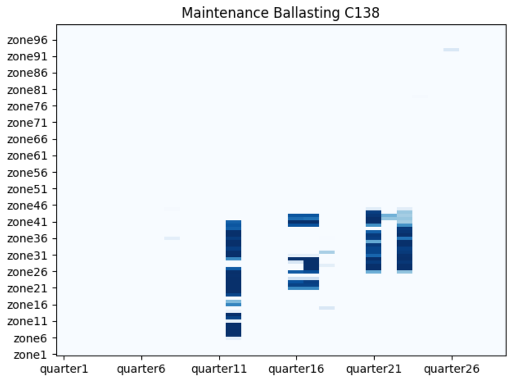

Top & line Spot Resurfacing C195
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21


Top & line Spot Resurfacing C138
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21
quarter26


Mechanised Resurfacing C195
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21


Mechanised Resurfacing C138
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21
quarter26


Maintenance Ballasting C195
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21


Maintenance Ballasting C138
zone96
zone91
zone86
zone81
zone76
zone71
zone66
zone61
zone56
zone51
zone46
zone41
zone36
zone31
zone26
zone21
zone16
zone11
zone6
zone1
quarter1
quarter6
quarter11
quarter16
quarter21
quarter26

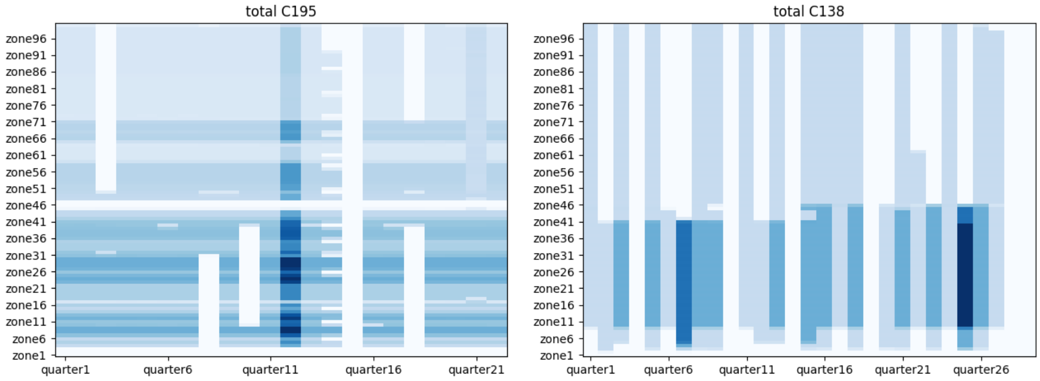


Figure 21. Trends emerge in work order data, on being aggregated into zones and quarters

Furthermore, visual observation reveals a high correlation between the **work order count** for C195 and the **squared** track geometry metrics **topL** and **topR :**

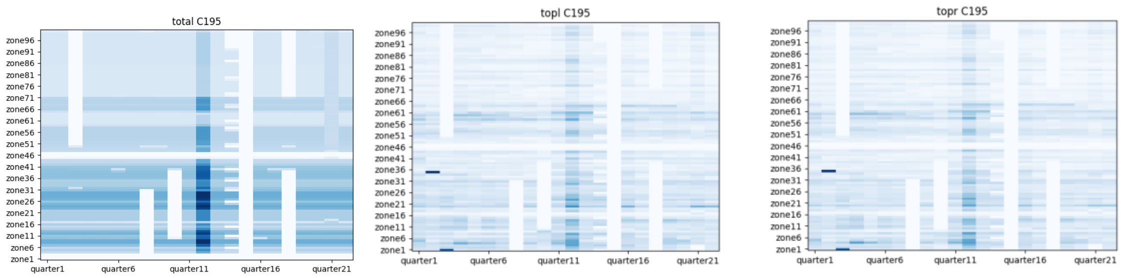


Figure 22. (from left) total work order count, Top Left(TRC) and Top Right (TRC) binned into zones and quarters

Indeed, values are highly correlated using the Spearman's Rank Correlation Coefficient, indicating that maintenance work impacts track geometry (Table 11)

Metric	**Spearman's Rank Correlation Coefficient**
Top L	0.64
Top R	0.63
Twist 3	0.72

Table 11: spearman correlation of maintenance data

[7.5] Experiment 4: Using feature transformation to conduct Zone-wise Maintenance Count Prediction for future quarters (Track: C138, C195)

Hypothesis: zone-wise analysis of data reveal maintenance load over the upcoming time periods.

[7.5.1] Data Pre-processing

For each maintenance type, the data has been aggregated into zones and quarters. The values represent the amount of maintenance work carried out during the quarter per zone (Figure 23).

	quarter1	quarter2	quarter3	quarter4	quarter5	quarter6	quarter7	quarter8	quarter9	quarter10	quarter11	quarter12	quarter13	quarter14	quarter
zone1	0	471	0	0	0	0	424	0	174	0	0	0	2	0	0
zone2	0	0	0	0	0	0	840	0	377	0	0	254	0	0	0
zone3	0	0	0	0	252	0	345	0	0	0	0	0	0	91	0
zone4	0	0	0	0	1012	0	1497	0	615	0	0	754	0	1405	0
zone5	0	0	0	1006	0	0	1440	0	834	0	0	2984	0	0	0
zone6	0	0	0	0	0	0	1278	0	2156	0	0	3150	0	0	0
zone7	0	0	1453	0	0	0	0	0	1823	2181	0	3002	0	0	0
zone8	0	0	0	0	0	0	0	0	1143	3135	0	0	0	1500	0
zone9	0	0	0	0	0	0	0	0	0	0	0	0	0	3915	0
zone10	0	0	0	0	0	0	0	0	813	0	0	0	0	576	0
zone11	0	0	0	0	0	0	1623	0	2460	0	0	351	0	0	0
zone12	0	0	0	0	0	0	2988	0	4830	0	0	0	0	0	0
zone13	0	0	954	0	0	0	1458	0	0	0	0	0	0	0	0
zone14	0	0	0	0	0	0	886	0	1632	0	0	0	0	0	0
zone15	0	1440	0	404	0	0	0	0	878	0	0	2380	0	0	0
zone16	0	3492	0	0	0	0	0	0	566	0	0	3004	0	0	0
zone17	0	1734	0	0	0	0	0	0	0	0	0	948	0	0	0
zone18	0	516	0	0	1018	0	914	0	0	0	0	3280	0	0	0
zone19	0	0	0	0	0	0	174	0	174	0	0	1892	0	0	0
zone20	0	0	0	0	0	0	2476	0	1322	0	0	3956	0	0	0
zone21	0	0	0	1268	0	0	1444	0	864	0	0	4380	0	638	0
zone22	0	0	0	0	0	0	3510	0	248	0	0	2984	0	0	0
zone23	0	0	0	0	0	0	2059	0	0	0	0	0	0	0	0
zone24	0	0	0	0	0	0	0	0	543	0	0	4563	0	0	0
zone25	0	0	0	0	0	0	2415	0	0	0	0	1563	0	0	0
zone26	0	0	0	0	0	0	0	0	1353	0	567	1293	0	0	0
zone27	0	0	0	0	0	0	0	0	0	0	1872	0	0	3879	0
zone28	0	0	0	0	1644	0	0	0	0	0	1611	0	0	0	0
zone29	0	0	0	0	0	0	0	0	0	0	3084	4950	0	0	0
zone30	1941	333	0	0	0	0	0	0	954	0	711	1929	0	0	0
zone31	0	1212	0	0	0	0	0	0	1803	0	0	381	0	0	0
zone32	0	0	0	974	0	0	0	0	0	0	1906	0	0	0	0
zone33	0	0	1812	2050	0	0	772	1530	766	0	0	2556	0	0	0

Figure 23. Processed Data for work order type = 'Mechanised Resurfacing' for C195. For this experiment, tracks were binned into 100 zones each over 22 quarters of data for C195 and 26 quarters of data for C138.

[7.5.2] Features and Response variables

The zone and the quarter (modulo 4) were used as features. The response variable is the amount of maintenance for each quarter and zone. The data is split into 75% training, and 25% test.

[7.5.3] Model and Evaluation

A Random Forest Regression Model was used. The **n_estimators** hyperparameter was tuned to 10. Random Forest achieved ~45% test accuracy (table 12, figure 24), suggesting a possibility of using previous data to predict future zonal maintenance load.

Metric	Score
RF accuracy	79.91%
RF test score	0.454
Out of bag score	0.2413

Table 12. Accuracy score for maintenance predictions using Random Forest

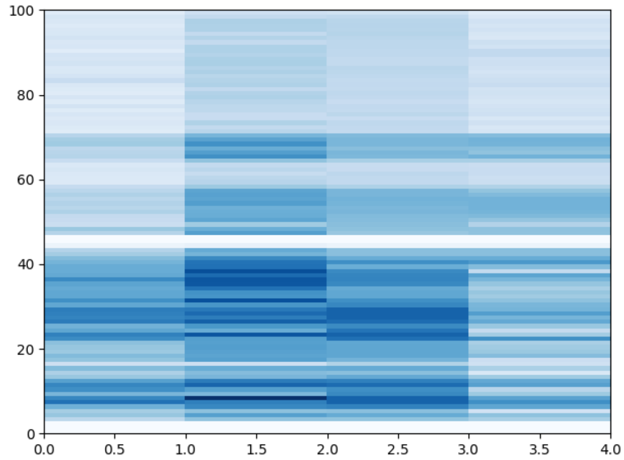


Figure 24 Predictions for the upcoming 4 quarters

[7.6] Experiment 4: Predicting maintenance load 1 quarter ahead using TRC and GPR data as model features (Track: C138)

Hypothesis: TRC and GPR features capture information about track condition; are predictive of upcoming maintenance.

[7.6.1] Data Pre-processing and train, test, hold-out split

C138 track data amounted to 1,047,771 rows and 77 columns. Presence of maintenance work over a location was encoded as a binary response variable.

To simulate an engineer uploading raw TRC readings, a hold-out test set (unbalanced data) consisted raw data from 4 quarters. Due to class-imbalance, down-sampling of non-maintained track was performed on the remaining data (balanced data), then partitioned into 50% training and 50% test sets.

[7.6.2] Random Forest with hyperparameter tuning achieves best results in predicting upcoming maintenance load

Using balanced GPR and TRC, a random forest model achieved a 90% test accuracy in predicting required maintenance over the next quarter (Figure 25).

The model performance was reduced in non-balanced test data however, indicating that the variation in measures from track not requiring maintenance was not robustly captured by the model (Figure 26).

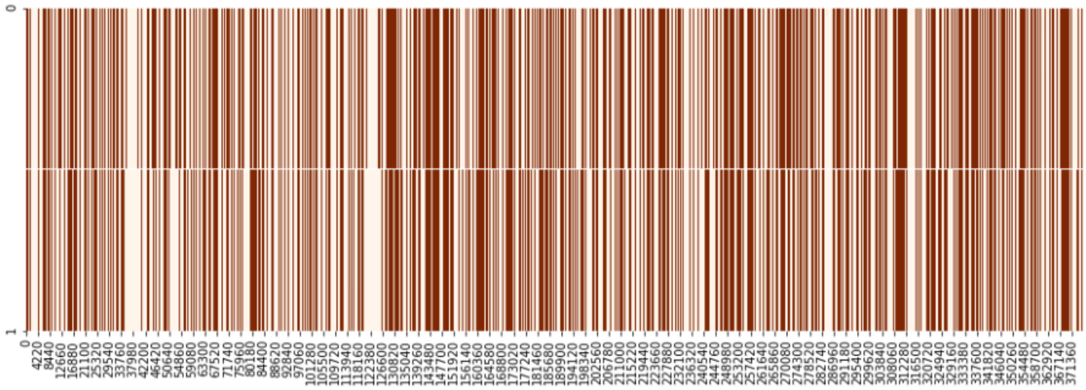


Figure 25: Actual (top) vs predicted (bottom) classification of maintenance work 1 quarter ahead. Random Forest + hyperparameter tuning achieved ~90% accuracy achieved using class balanced datasets.

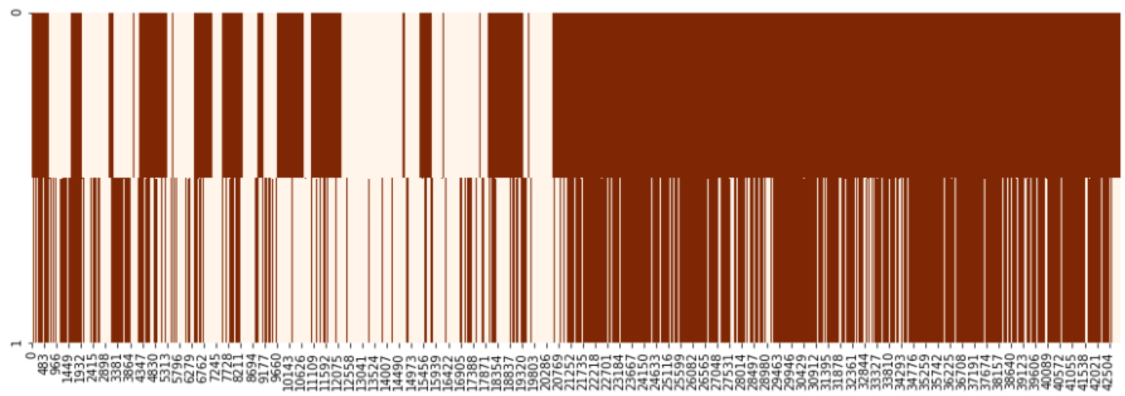


Figure 26: Actual (top) vs predicted (bottom) classification of maintenance work 1 quarter ahead. Random Forest + hyperparameter tuning achieved ~72% accuracy achieved using raw (unbalanced) datasets

The results of the work order classification experiments are summarised in Table 13.

Model	**Accuracy (bal.)**	**F1 Score (bal.)**	**Accuracy (unbal.)**	**F1 Score (unbal.)**
Logistic Regression	57.4%	0.61	NA	NA
SVM	80.6%	0.38	NA	NA
Random Forest (base)	87.6%	0.44	65.3%	0.37
Random Forest + CV	89.5%	0.40	71.87%	0.40

Table 13: work order classification accuracy

[7.7] Assessing the usability of visualisations

The usability of visualisations was assessed using the following metrics:

Objective	Metric	Observation
Web page speed	Average page load time across 6 runs using 3 different browsers (seconds)	3.7 seconds
Additional insights enabled from visualisations	Number of insights derived from interactive visualisations unavailable in raw data	1. Correlation (or lack thereof) between work order history and rail geometry 2. Change of rail geometry across time and meterage
Provision of visualisations in a format useful to QR	In-person feedback	Satisfactory, based on Skype conference call with QR engineers and executives.
Completeness of capture of key visualisations and drill-downs	Confirmation to be provided by QR during demonstration	Scheduled (December 2019)
Ease of use	Subjective test on independent party to assess their ease of navigation	

Table 14: assessment of the usability of visualisations

Through user-driven software development, visualisations and interactivity were based on QR use-cases. A save function is provided on every for users to adjust plot view before taking a snapshot. A balance is struck between displaying extensive plots and website load time. Optimisation methods to reduce loading time include caching html pages, merging low-value heatmap points into the background, and serialising large plots as json to load directly - a non-trivial task since the TRC-C195 file was ~0.5 Gb in size.

Two approaches were used to facilitate QR's software engineers integrating our tools:

1) Deploy-as-is on their servers - the software is packaged in a Docker container for trivial cross-platform deployment, and
2) Further development work - provide standard build instructions on a Linux platform to compile on their systems.

Test suites, linters and better code documentation, and build instructions in other platforms are planned for future iterations.

[7.8] Future improvements

Several potential extensions to the experiments were identified:

Proposed Extension	Objectives
Testing on different types of work orders	Extend preliminary investigation on predicting types of work orders (not discussed)
Control for maintenance work	Control for maintenance work in the predictions of "Combined" response variable
Extend longer-horizon predictions	Continue experiments in longer-term prediction horizons

Table 15: potential extensions

[8] Conclusions and Discussion

Key findings include:

Key Finding	Discussion
Data alignment was effective	pre-processing activities appear to effective in aligning track segments and disparate feature datasets
PDF of response variable was estimated	the “Combined” target variable was approximated as an exponentiated Weibull PDF, consistent with PDFs of wearing parts in other maintenance settings
Feature selection was effective	LASSO regression was used to reduce the number of features to as few as 9 while still enabling reasonable performance from the ML methods
Baseline prediction (without ML) is accurate in 1-quarter horizon when properly aligned	utilising the most recent value of the target variable delivered ~80% test accuracy when predicting the target variable in the next quarter
ML regression delivered slight improvements over the baseline in predictions in the 1-quarter horizon	Random Forests, ANNs and SVR delivered ~3% improvement in prediction accuracy for the following quarter
ML regression delivered high improvements over the baseline in “high priority” predictions in the 1-quarter horizon	in the 1-quarter horizon, the ML models’ prediction accuracy was ~12% higher for the “high priority” points
ML regression delivered high improvements over the baseline in predictions in the 2-quarter horizon	over 2 quarters, the baseline method achieved only 52% test accuracy versus 74% for RF regression
ML regression delivered high improvements over the baseline in “high priority” predictions in the 2-quarter horizon	over 2 quarters, the baseline method achieved only 29% test accuracy versus 71% for RF regression on the “high priority” points
ML delivered high accuracy on a balanced dataset predicting work order requirements	RF classification delivered 90% test accuracy when using a balanced dataset
ML delivered moderate accuracy on a balanced dataset predicting work order requirements	RF classification delivered 72% test accuracy when using the raw (unbalanced) dataset
Visualisations provided high levels of insight into the raw data	The ability to visualise and interact with joined work order/TRC data enabled the impact of maintenance on rail geometry to become immediately apparent
Aggregating work order data into zones and quarters and	It became evident that there is a high correlation between work orders and track metrics, proving the fruitfulness of maintenance work. Also, zones prone to

visualizing them as heatmaps revealed interesting insights.	maintenance could be identified and predictions for work orders for the upcoming quarters could be predicted with sufficient accuracy.
Random forest regressions on TRC data, using time series values as features, gave moderate accuracy.	The need to predict TRC metrics for unmapped regions for a quarter, when historical data is available, without depending on other metrics can provide useful. This model used only historical values of the same metric as predictors.
Several opportunities for extension and improvement were identified	These include expanding the use of work order in the ML models and extending the work on longer-horizon predictions

Table 16: summary of key findings

[8.1] Future enhancement

As the current implementation is at a proof-of-concept stage, future enhancements to the application may include:

1) Further tuning of algorithms to provide greater accuracy (and other model metrics)

2) Test and extend use of feature transformation across models

3) Deploying additional ML algorithms to the application to suit QR's use cases

4) Integrating data preparing scripts to application

5) Conduct user testing to further improve application

6) Incorporating additional considerations to deployment in QR's systems

[8.2] Conclusion

Receiving data of track measurement and track properties from QR, the team showed two key findings:

1) Compared to a baseline model, random forest may predict the presence of changes to track geometry (attributed most to ballast condition) up to two quarters ahead.

2) Location of upcoming maintenance may be predicted from current data.

Together, these findings suggest that predictive ML may be applied to further augment QR's decision-making around maintenance and ballast condition.

The team also provided a proof-of-concept Django application incorporating preliminary workflows for the use of ML by QR, allowing users to upload data to run predictions. The application further seeks to augment QR's processes via the use of interactive visualisations to both TRC and GPR datasets, allowing users to explore track geometry changes quickly and view GPR measures in a geospatial context.

[9] Distribution of Team Member Effort

All team members have contributed similar amount of effort.

Key roles of team members are summarised below:

Team Member	Key Role
Liu	Pre-processing of data, visualisations and ML algorithms, Django app development
Chowdhury	Visualisations, time-series algorithms
Yin	Visualisations, Bokeh web app development, webhosting
Salouk	Interface with Queensland Rail, ML algorithms

Table 17: key team member roles

Bibliography

1. Asadzadeh, Seyed Mohammad, Roberto Galeazzi, Emil Hovad, J. F. Andersen, Camilla Thyregod, and A. F. S. Rodrigues. "Ballast Degradation Modeling for Turnouts based on Track Recording Car Data." In *PHM Society European Conference*, vol. 4, no. 1. 2018.
2. Bryant, Fred B., and Paul R. Yarnold. "Principal-components analysis and exploratory and confirmatory factor analysis." (1995).
3. Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144-152. ACM, 1992.
4. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *ML* 20, no. 3 (1995): 273-297.
5. Dunteman, George H. *Principal components analysis*. No. 69. Sage, 1989.
6. Eriksen, A., J. Gascoyne, and W. Al-Nuaimy. "Improved productivity and reliability of ballast inspection using road-rail multi-channel GPR." *Railway Engineering* 14 (2004).
7. Eriksen, Asger, Ben Venables, Jon Gascoyne, and Shubho Bandyopadhyay. "Benefits of high speed GPR to manage trackbed assets and renewal strategies." In *PWI Conference, June*, vol. 19. 2006.
8. Eriksen, Asger, Jon Gascoyne, Cathal Mangan, and Ron Fraser. "Practical applications of GPR surveys for trackbed characterisation in the UK, Ireland, USA and Australia." *CORE 2010: Rail, Rejuvenation and Renaissance* (2010): 380.
9. Hajizadeh, Siamak, Alfredo Núnez, and David MJ Tax. "Semi-supervised rail defect detection from imbalanced image data." *IFAC-PapersOnLine* 49, no. 3 (2016): 78-83.
10. Kemsley, E. K. "Discriminant analysis of high-dimensional data: a comparison of principal components analysis and partial least squares data reduction methods." *Chemometrics and intelligent laboratory systems* 33, no. 1 (1996): 47-61.
11. Kelleher, Christa, and Thorsten Wagener. "Ten guidelines for effective data visualization in scientific publications." *Environmental Modelling & Software* 26, no. 6 (2011): 822-827.
12. Lasisi, Ahmed, and Nii Attoh-Okine. "Principal components analysis and track quality index: A ML approach." *Transportation Research Part C: Emerging Technologies* 91 (2018): 230-248.
13. Nakhaee, Muhammad Chenariyan, Djoerd Hiemstra, Mariëlle Stoelinga, and Martijn van Noort. "The Recent Applications of ML in Rail Track Maintenance: A Survey." In *International Conference on Reliability, Safety, and Security of Railway Systems*, pp. 91-105. Springer, Cham, 2019.
14. Olhoeft, Gary R., and Ernest T. Selig. "Ground-penetrating radar evaluation of railway track substructure conditions." In *Ninth International Conference on Ground Penetrating Radar*, vol. 4758, pp. 48-53. International Society for Optics and Photonics, 2002.
15. Peng, Fan. "Scheduling of track inspection and maintenance activities in railroad networks." PhD diss., University of Illinois at Urbana-Champaign, 2011.

16. Peng, Fan, Seungmo Kang, Xiaopeng Li, Yanfeng Ouyang, Kamalesh Somani, and Dharma Acharya. "A heuristic approach to the railroad track maintenance scheduling problem." *Computer-Aided Civil and Infrastructure Engineering* 26, no. 2 (2011): 129-145.
17. Queensland Rail (2019) ‘Queensland Rail Annual Report 2018-2019’
18. Shao, Wenbin, Abdesselam Bouzerdoum, Son Lam Phung, Lijun Su, Buddhima Indraratna, and Cholachat Rujikiatkamjorn. "Automatic classification of ground-penetrating-radar signals for railway-ballast assessment." *IEEE Transactions on Geoscience and Remote Sensing* 49, no. 10 (2011): 3961-3972.
19. Sharma, Siddhartha. "Data-Driven Optimization of Railway Track Inspection and Maintenance Using Markov Decision Process." PhD diss., The State University of New York at Buffalo, 2016.
20. Stone, Mervyn. "Cross-validatory choice and assessment of statistical predictions." *Journal of the Royal Statistical Society: Series B (Methodological)* 36, no. 2 (1974): 111-133.
21. http://ec2-3-18-150-48.us-east-2.compute.amazonaws.com:8000/QRvisualisation
22. https://www.queenslandrail.com.au/Community/Projects/Pages/QueenslandRailStablingProgram.aspx

Appendix 1 - Pre-processing activities

[A2.1] C138 and C195 data pre-processing

Objective

Prior to analysing the data, pre-processing was undertaken to align the TRC readings, bin work orders into quarters to match with TRC and match TRC with work orders and GPR datasets. A python script was created to automate this purpose.

Aligning TRC

As TRC readings drift from run-to-run, any analysis based off cross-sectional data across the distance of track requires alignment before they can be joined to other measures. TRC was aligned based on “GAUGE” and “SUPER” measures. This was suggested by QR as the most precise measures amongst the 18 TRC readings.

Process

1. For each TRC recording, offset the next quarters' TRC GUAGE and SUPER readings by 50 metres. Take the difference between the values from each quarter and compute standard deviation for the full set of differences
2. Offset the next quarters' GUAGE and SUPER readings by 1 metre less than the previous iteration, calculate standard deviation
3. Once 100 metres has been crossed (i.e. 50 metre offset both ways), compare all standard deviations for SUPER and GUAGE and see which offset value provided the lowest standard deviation
4. Take the mean offset value (rounded up) with the lowest standard deviation between SUPER and GUAGE
5. Apply the desired offset and move on to the next quarter. Repeat until the end of the dataset has been reached

Bin work orders into quarters

Work orders can occur sporadically. To match Work orders with TRC data requires binning of the work orders into periods between TRC runs. This way, matching of maintenance work orders ensures that the rail geometry measures from the TRC data reflect the track condition just prior to maintenance work.

Matching with GPR data

GPR data was obtained from the 2015 readings. The GPR dataset contains readings in approximately 5 metre precision, whereas TRC readings had a precision of 1 metre, and had quarterly readings between 2014 and 2019. Since GPR measures structural properties in the deeper layers of the track, it is treated as a control in the analysis of C138 and C195. Joining the GPR readings involved casting the values across distance to accommodate the reduced precision, and time to make up for the lack of regular readings.

Obtaining the work order classification response variable

After alignment and joining the data, the response variable was created by assigning a binary class to areas with or without work orders during a particular period.

Balancing the data

A down-sampling of data was done on track that did not undergo maintenance during the quarter succeeding TRC measurements.

Train, test split with hold out data

50% of data was partitioned for train and test, while a hold-out set comprising recordings made on: '2018-06-26', '2015-05-15', '2016-06-03', '2019-02-07' for C138, and all recordings for C195, were used to measure model accuracy. This hold-out set was designed to simulate engineers feeding new TRC data into the algorithm for predictions, as opposed to having a pre-balanced set of data.

[A2.2] C139 data pre-processing

Objective

Key pre-processing actions included:

1. Alignment of TRC datasets	As described above for C138 and C195
2. Alignment of GPR to TRC	using meterage measures
3. Calculate standard deviations and the Combined metric	across 20 metre sections of TRC measures
4. Alignment of drainage points	using meterage measures
5. Train/test split	the dataset was split into training (75%) and test (25%) sets
6. Standardisation	feature data was standardised to mean 0, standard deviation of 1 (as required by models such as K-NN and SVR)

Appendix 2 – ML Regression Models

Feature Engineering

The features selected by the models are as follows:

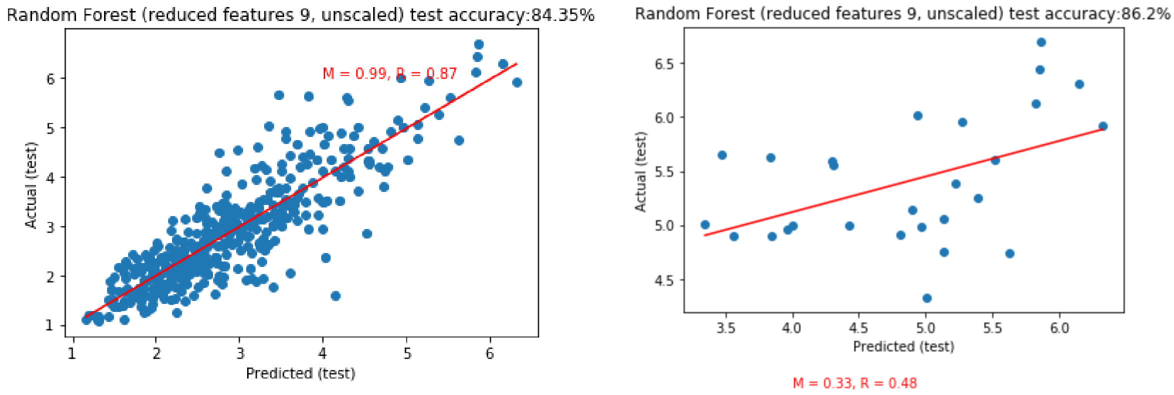
Linear Regression ("OLS")	LASSO alpha=0.01 ("20")	LASSO alpha=0.1 ("9")
BDMCentre	BTILeft	PVCCentre
BDMRight	BVMLeftCategory	PVCLeft
BVMCentreCategory	BVMLeftVolume	PVCRight
BVMCentreVolume	BVMRightCategory	SDTopLeft3
BVMLeftCategory	LRICentre	SDTopRight3
BVMRightCategory	LRILeft	SDTwist101
Drainage	PVCCentre	SDTwist103
LRICentre	PVCLeft	SDTwist33
LRILeft	PVCRight	SDVersL1
SDTopLeft1	SDTopLeft1	
SDTopLeft2	SDTopLeft2	
SDTopLeft3	SDTopLeft3	
SDTopRight2	SDTopRight1	
SDTopRight3	SDTopRight3	
SDTwist103	SDTwist101	
SDTwist33	SDTwist103	
SDVersL3	SDTwist33	
SDVersR3	SDVersL1	
TDILeft	SDVersL3	
	TDILeft	

Legend: common to all 3, common between LASSO 20 and OLS, common between LASSOs

Random Forest Regression

ML Model	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
• All features	85.42%	1.02	84.75%	0.3
• "OLS" features	85.06%	1.01	84.39%	0.37
• LASSO "20"	85.09%	1.01	85.51%	0.33
• **LASSO "9"**	**84.35%**	**0.99**	**86.2%**	**0.48**

Results are for unscaled data. Random Forest fitted for 1,000 trees using a random sample of (number of features)$^{1/2}$.

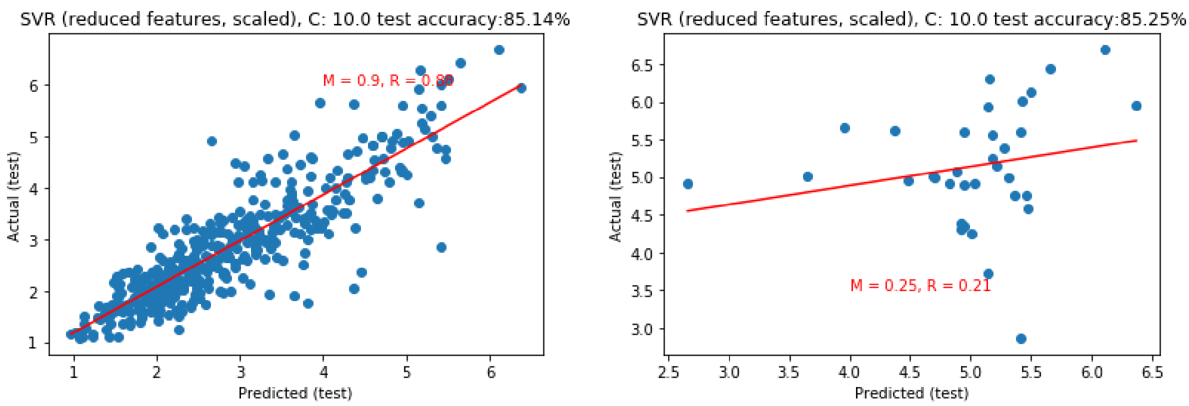


Random Forest: 9 features

Support Vector Regression

ML Model	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
• All features	84.8%	0.92	84.70%	0.03
• "OLS" features	83.29%	0.89	84.72%	0.22
• **LASSO "20"**	**85.14%**	**0.9**	**85.25%**	**0.2**
• LASSO "9"	82.9%	0.9	84.44	0.17

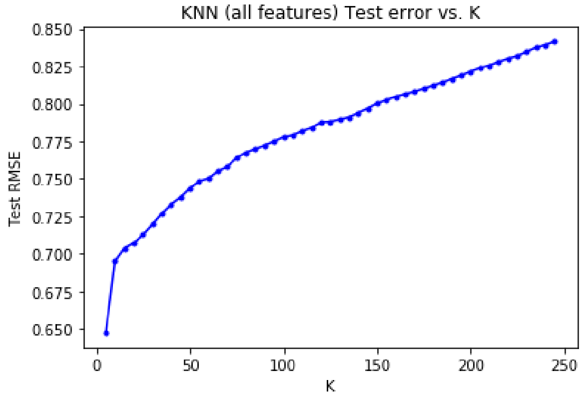
Results are for scaled data and radial basis function (RBF) kernel which outperformed sigmoid and polynomial kernels. Optimal regularisation parameter (C) = 10 i.e. the model traded a relatively small margin for higher training accuracy.

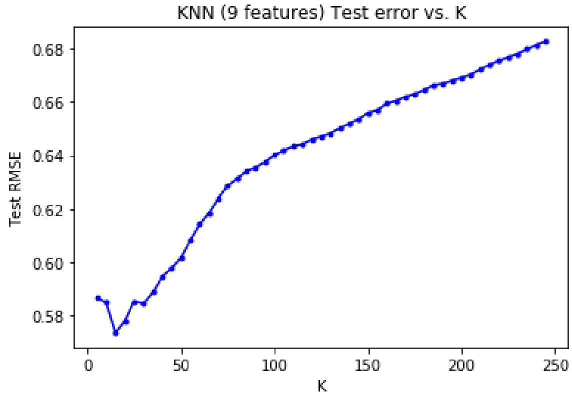


Support Vector Regression: 20 features

K-NN Regression

The lack of "elbow" in the Test RMSE versus K plot using all features indicated the target variable, Combined, is not consistently correlated with a similar set of features i.e. the combination of features and response are relatively unique. On this basis, it was not expected that KNN using all features would perform well on the test dataset. This is contrasted with a clear optimal K (15 neighbours) when using only 9 features.





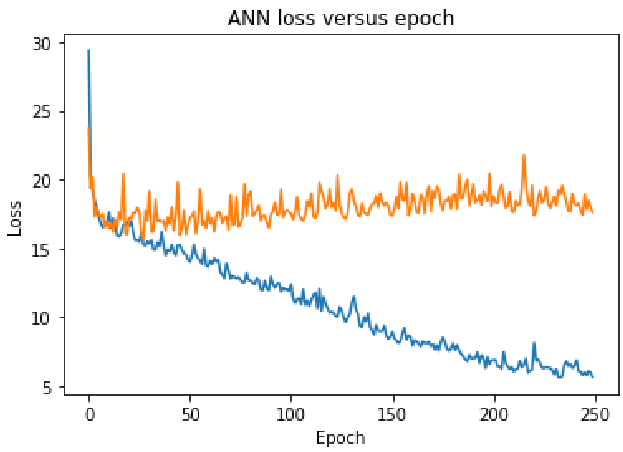
ML Model	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
• All features	81.68%	1.04	80.08%	0.13
• "OLS" features	83.59%	0.99	83.19%	0.08
• LASSO "20"	83.33%	1.0	80.93%	0.08
• **LASSO "9"**	**82.21%**	**1.02**	**81.64%**	**0.23**

Results are for scaled data using a ball tree algorithm and Manhattan distance.

Artificial Neural Networks (ANN)

ANNs were developed using the KerasRegressor, Sequential (from the Keras library in Python) and MLPRegressor (from sklearn neural_network in Python).

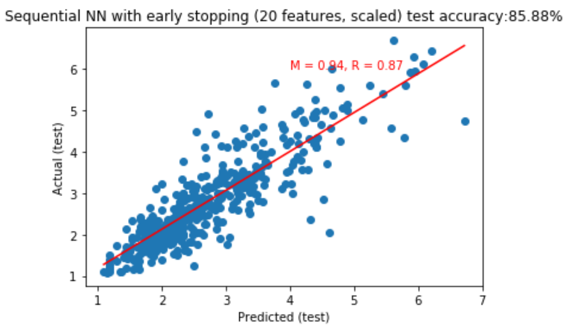
Results are shown for the Sequential model implemented with early stopping.

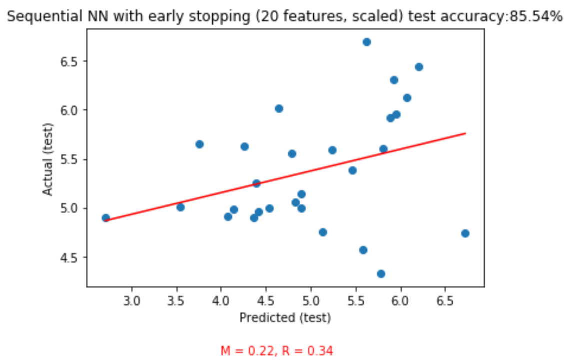


The importance of early stopping is evident where the test error (orange) starts to rise after relatively few epochs despite training loss continuing to fall.

ML Model	Best Test Accuracy	Gradient of best-fit for Best Model	Best Test Accuracy "High Priority"	Best Correlation "High Priority"
• All features	85.53%	0.88	83.09%	0.05
• "OLS" features	85.33%	0.90	85.08%	0.32
• **LASSO "20"**	**85.88%**	**0.94**	**85.54%**	**0.34**
• LASSO "9"	84.16%	0.91	80.69%	0.31

Results are for scaled data and network architecture comprising 6 hidden layers using the Adam optimiser.





ANN: 20 features