**2800 words, 12pt font**

**[2%] Introduction - Motivation**

**[3%] Problem definition**

**[5%] Survey**

**Proposed method**

* **[10%] Intuition - why should it be better than the state of the art?**
* **[35%] Description of your approaches: algorithms, user interfaces, etc.**

**Experiments/ Evaluation**

* **[5%] Description of your testbed; list of questions your experiments are designed to answer**
* **[25%] Details of the experiments; observations (as many as you can!)**

**[5%] Conclusions and discussion**

**[-5% if not included] Distribution of team member effort.**

**Team 17 – Final Report (2266 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 current practices, track geometry is used as a direct indicator of ballast condition. The Project enhances QR’s current practice by applying interactive visualisations to the inspection of multiple datasets and machine learning 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 report22*

**[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:
3. *Alignment of disparate features from multiple sources:* QR separated their data into multiple files, with work orders separate from Ground Penetrating Radar (GPR) and Track Recording Car (TRC) data respectively. The track recording files were further segmented into multiple track ID and time of collection. To compile the data into a master file, common indices are required.
4. *Identification of useful features, outliers and meaningful response variables:* Due to the comparatively large number of features to time points available, feature selection is imperative to prevent over-fitting. To this end, the project utilised LASSO regression and the P-values derived from Linear Regression to select a subset of features. The advantage of LASSO over PCA5 is the higher interpretability of features compared to eigenvectors in PCA. Previous studies on track recording data have conversely demonstrated utility of PCA 10
5. Applying Machine Learning (ML) algorithms to predict on collected data. The current report focuses on predicting future track geometry and maintenance work.
6. Interactive visualizations provide insights into QR’s data that may assist decision making.

**[3] Survey Overview**

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

Nakhaee et al13, employ deep learning algorithms to detect structural defects in rail (not ballast). Several impediments were identified:

1. Class imbalance: as the vast majority (>99%) of rail is non-defective, classifiers are biased.
2. Availability of labelled datasets: it is time- and skill-intensive to label thousands of kilometres of rail.
3. Model explainability: algorithms typically utilise “black box” solutions, such as convolutional neural networks, which are difficult to debug and explain to management.

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 across multiple datasets and Machine Learning to enhance predictive power.

The processes employed by the Project extends current methods within the industry. Much of current research is focused on locomotives and rail condition. By contrast, the Project is focussed on track geometry which is a proxy for ballast quality. While Sharma19 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 (a proxy for ballast condition), 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 Machine Learning for enhanced predictions and interactive visualisations for efficient analysis and effective communication.

**[5.1] Innovations**

Key innovations include:

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

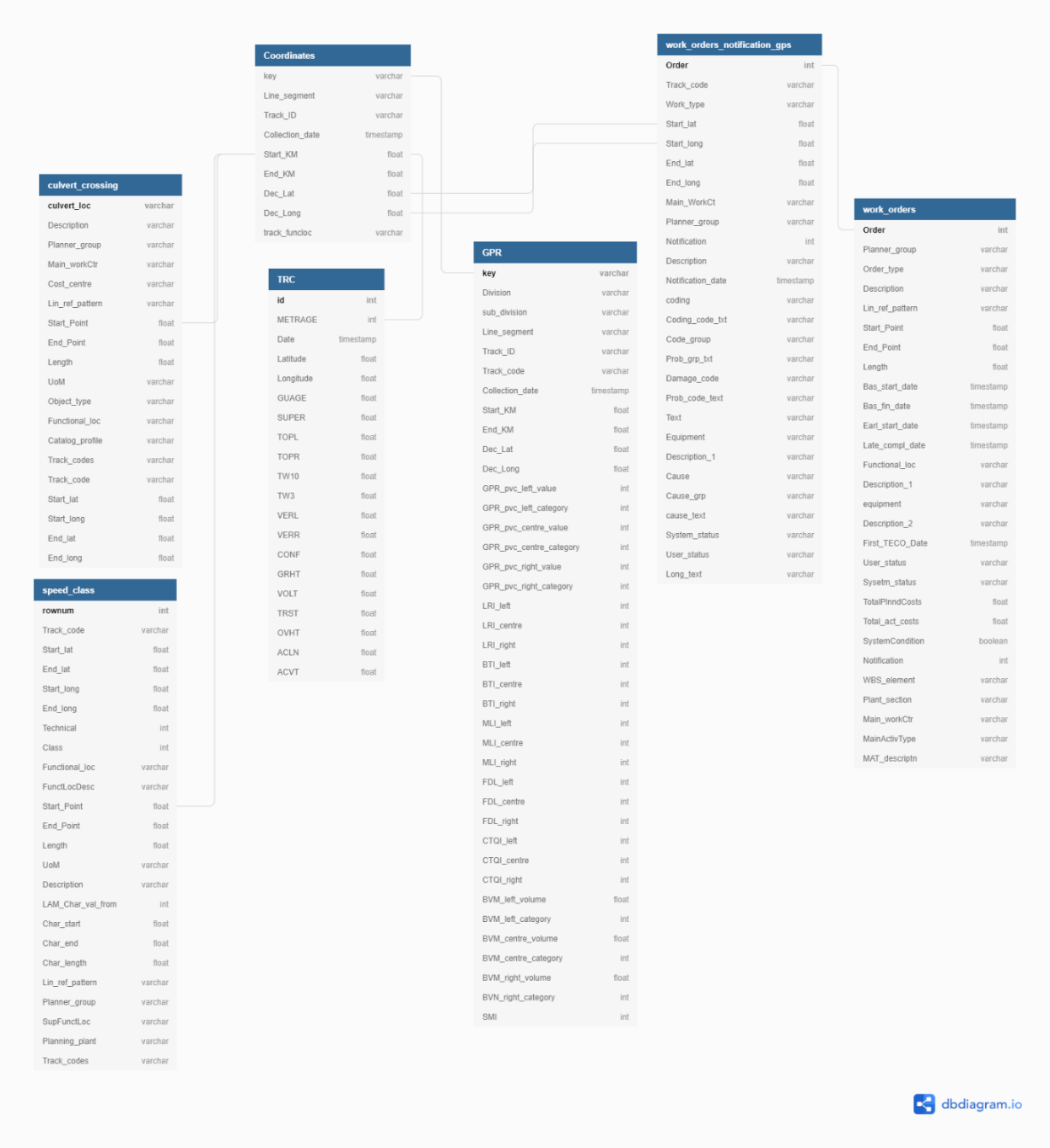
*Table 1. Work phases, tasks and progress.*

**[5.2.1] Algorithms Developed**

**[5.2.1.1] Features**

Data manipulation and overview of available measures:

1. Data matching and joining was predominantly implemented using the pandas library (version 0.25.1) in python. Due to the overlapping distance measures along different segments of tracks, creation of the master data (Figure 2) involved first segmenting into the different track codes before joining on the distance of each value measured along the track.
2. The work orders dataset further 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 * Centre * Right | * GPR Data | Assess feature importance |
| 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 * 4th Level LRP to 3 digit TC * 5th Level LRP to 4th 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 Machine Learning Algorithms**

The algorithms developed during the Project include:

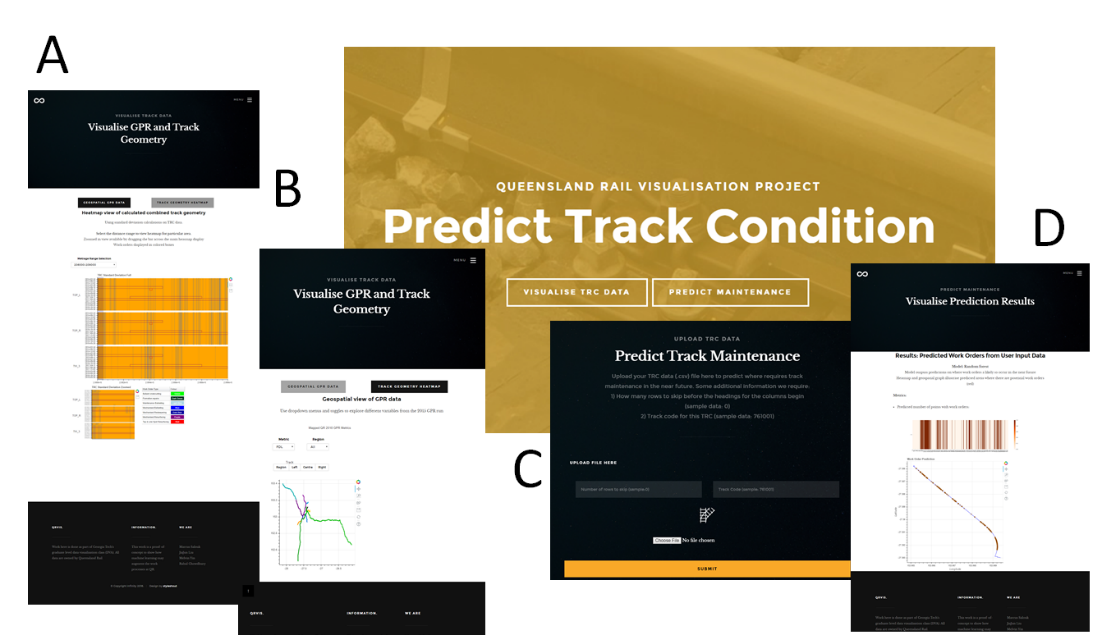
|  |  |  |
| --- | --- | --- |
| **Purpose** | **Algorithm** | **Source** |
| 1. Align TRC quarterly datasets | Minimise standard deviations of semi-static features | Custom developed |
| 1. Estimate underlying distribution | Exponentiated Weibull Fit | stats; exponweib |
| 1. Feature selection | LASSO | sklearn: linear\_model.Lasso |
| 1. Feature selection | OLS p-values | Stats; OLS |
| 1. Feature selection | Elastic Net | sklearn: linear\_model. ElasticNet |
| 1. Regression prediction | Linear Regression | sklearn: linear\_model. LinearRegression and statsmodels.api: sms |
| 1. Regression prediction | Random Forest Regression | sklearn.ensemble: RandomForestRegressor |
| 1. Regression prediction | Support Vector Regression – various kernels | sklearn.svm: SVR |
| 1. Regression prediction | K-NN Regression | sklearn; neighbors; model\_selection.GridSearchCV |
| 1. Regression prediction | Artificial Neural Network | keras.wrappers.scikit\_learn ; KerasRegressor; keras.layers; Dense ; keras.models ; Sequential |
| 1. Regression prediction | Artificial Neural Network (with early stopping) | keras.models ; Sequential; keras.callbacks EarlyStopping, ModelCheckpoint |
| 1. Regression prediction | Multilayer Perceptron | sklearn.neural\_network; MLPRegressor |
| 1. Feature transformation | Transformation of time-series variables to coexist with semi-static variable | Custom developed |
| 1. Classification prediction of upcoming maintenance work | Logistic Regression | sklearn; linear\_model.LogisticRegression |
| 1. Classification prediction of upcoming maintenance work | Support Vector Machine | sklearn.svm; SVC, GridSearchCV |
| 1. Classification prediction of upcoming maintenance work | Random Forest Classification | sklearn.ensemble; RandomForestClassifier; GridSearchCV |
| 1. ???Rahul’s models – time series? |  |  |

*Table 4: algorithms developed*

**[5.2.1.4] User Interfaces Developed**

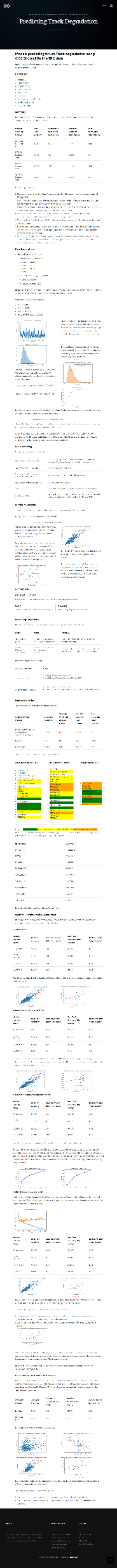
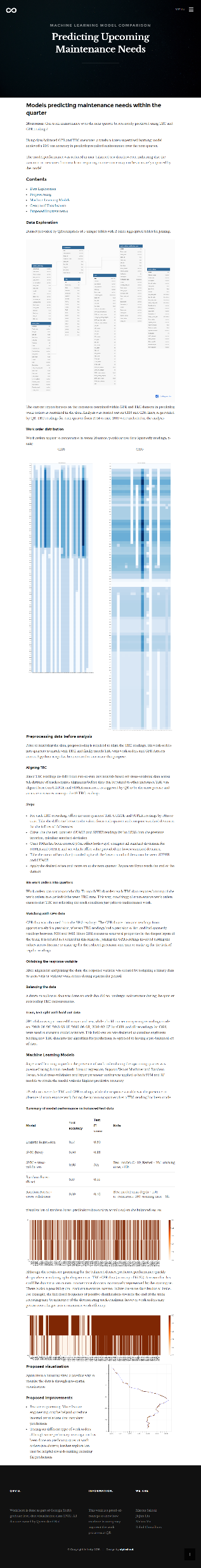
Interactive displays of input features in addition to predictions deriving from the machine learning models are presented using the Django framework, with embedded plots from Bokeh, a python visualization library. These are hosted on an Amazon Elastic Compute Cloud (Amazon-EC2) instance 21.

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

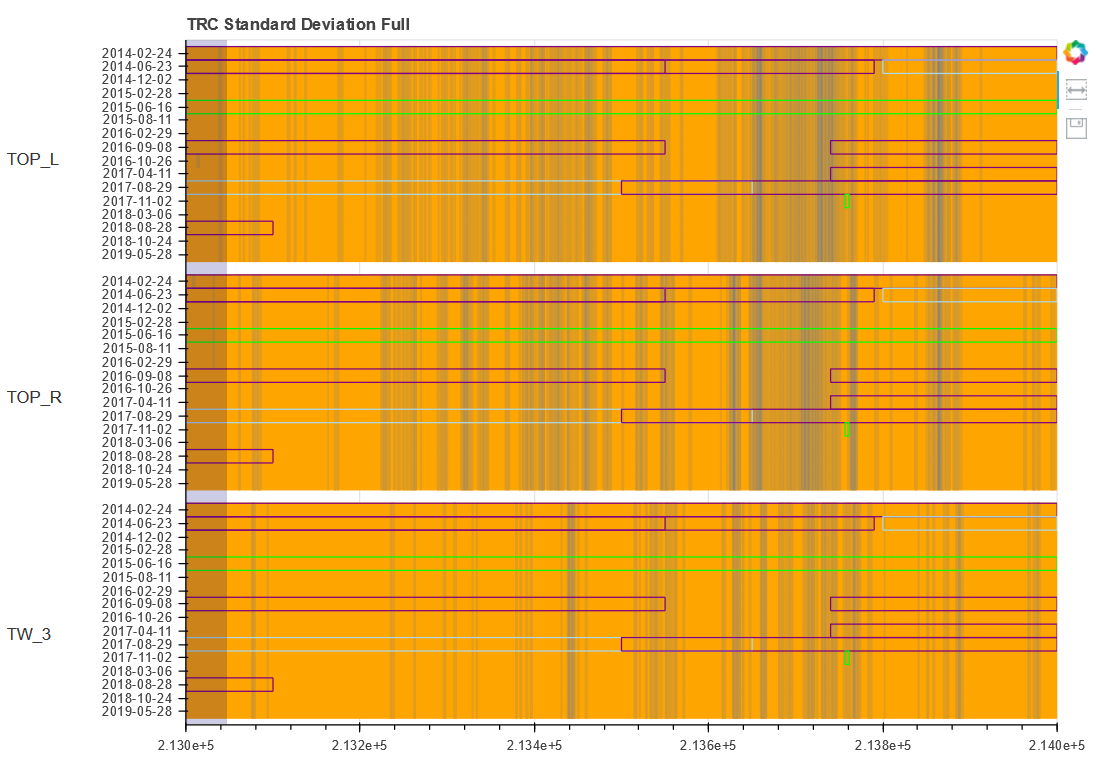
1. interact with existing GPR data to gain insight from visualisations (Figure 3B)
2. interact with existing calculated track geometry calculated using TRC data (Figure 3A, Figure 5)
3. execute a random forest machine learning model on user-input data to generate predictions of upcoming maintenance work (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)

C

*Figure 3. Developed web application views hosted on [aws](http://ec2-3-18-150-48.us-east-2.compute.amazonaws.com:8000/). 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.*





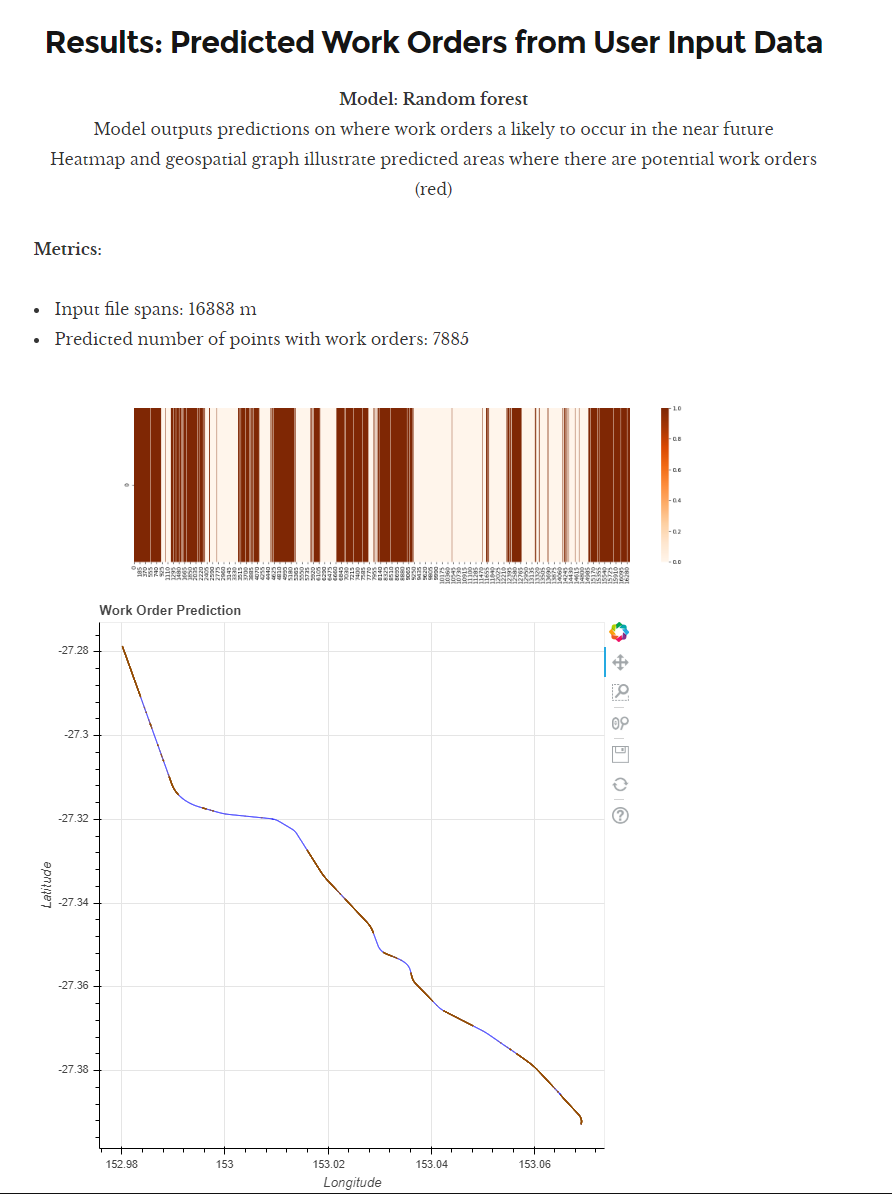
*Figure 5: interactive drill-down allowing inspection of maintenance impact*

Interactivity is enhanced via:

1. ability to select specific geographic segments to drill-down
2. utilities that enable uploading of new data to be visualised (Figure 5)
3. functionality enabling machine learning and time-series algorithms to be run on selected datasets in real-time

The visualisations implemented 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.

The page is manually tested to determine the number of data points presented, to provide acceptable end-user loading times. Figures 5 demonstrates an example of user interactivity, where sliding the grey viewbar changes the zoomed-in area of track shown, while the dropdown box selects the section of track presented, denominated in kilometres.



Dynamic Statistics

Heatmap classification view

Interactive Geospatial classification view

*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-machine learning methods (i.e. what could QR expect to achieve without applying machine learning)? |
| 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 machine learning 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 machine learning 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] Data preprocessing alignment and statistical analysis**

|  |  |
| --- | --- |
| For the C138 corridor of rail alone, 120 TRC files needed to be aligned and joined. The alignment was undertaken by locating the meterage offset that minimised the 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 work order history.  Figure 7 suggests the alignment process appears robust as there is clear continuity in the rail geometry across time periods.  Missing value imputation was accomplished by filling with ‘0’ value. | *Figure 7: aligned TRC datasets* |

One key response variable is “Combined” which reflects a linear combination of rail geometry features calculated by the TRC:



The distribution of the Combined variable was analysed and presented in Figure 8:

|  |  |
| --- | --- |
| There is no apparent structure associated with the response variable e.g. high values of the Combined metric are often preceded by low values in adjacent meterage.  The distribution of the response variable is skewed towards lower values of Combined. i.e. there are proportionally fewer samples that indicate higher degradation. | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\E5CC3D61.tmp  *Figure 8: high variance seen in the Combined metric* |

|  |  |
| --- | --- |
| As shown in Figure 9, the Project was able to effectively estimate the probability density function (PDF) of the Combined variable as an exponentiated Weibull distribution.  Its cumulative density function (CDF) (Figure 10). Based on current practice, a threshold for Combined value was set at 4.9 (“high priority”) where values above this are of high interest in maintenance decision making.  The significance of this analysis is i) the Combined response variable follows a similar distribution to part failure in many maintenance settings, and ii) based on the CDF, the probability of a sample exceeding the threshold was only 7.64%. | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\8B045253.tmp  *Figure 9: response variable estimated PDF* |
| C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\CAB192D9.tmp  *Figure 10: response variable estimated CDF* |

As the dataset is highly biased, a trivial classifier that only predicts {Combined < threshold} would achieve 92.36% accuracy.

**[7.2] Baseline prediction accuracy on predicting future track geometry**

The most trivial regression model: , achieved 63.86% test accuracy. The baseline prediction was improved by projecting the most recent quarter Combined value i.e. . The baseline achieved 80.92% test accuracy. Note heteroskedasticity in Figure 12 for “high priority” points.

|  |  |
| --- | --- |
| *C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A13F319B.tmp*  *Figure 11: Baseline predictive accuracy* | *C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\EF8CD961.tmp*  *Figure 12: Baseline prediction “high priority”* |

**[7.3] Feature assessment**

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 Net1 | 83.21% | 1.01 | 80.74% | 0.16 |

Note 1: not technically a feature selection method as includes coefficient shrinkage

*Table 6: feature engineering results*



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*

**[7.4] Machine Learning model performance (1 quarter prediction horizon)**

**[7.4.1] Regression experiments designed to predict “Combined”**

The machine learning results are summarised as:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning 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: Machine Learning summary of performance* | | |
| ***C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\66BC317.tmp***  *Figure 13: Random Forest Regression* | | ***C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B561743D.tmp***  *Figure 14: RF Regression “high priority”* |
|  |  | |
| Random Forest Regression (Figure 13, 14) and Sequential ANN achieved materially improved prediction accuracy on the “high priority” points than the baseline.  It was noted that implementing early stopping for the Sequential ANN (Figure 15) was key to optimising its performance.  The Random Forest was fitted on 1,000 trees with features1/2 random features.  The network architecture of the ANN comprised 6 hidden layers using the Adam optimiser. | *C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\99AFF52D.tmp*  *Figure 15: importance of early stopping in ANN* | |

The results of the Machine Learning experiments implemented to predict future-quarter Combined 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: Machine Learning summary of performance*

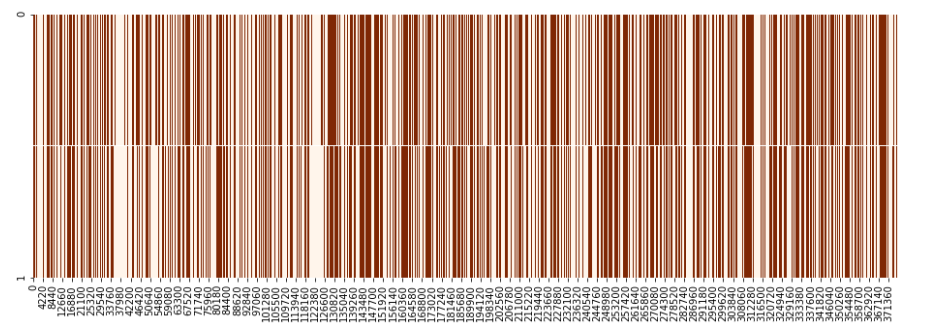
Additional experiments and observations are outlined in Appendix 3.

**[7.4.2] Classification experiments that predict maintenance requirement**

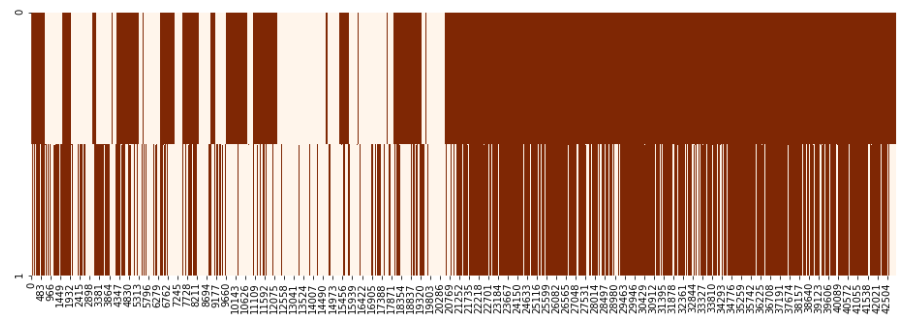
Several models were developed to assess whether track maintenance over the next quarter could be accurately predicted using TRC and GPR readings.

Using class-balanced GPR and TRC measures, a random forest supervised learning model achieved a 90% test accuracy in predicting required maintenance over the next quarter (Figure 16).

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 17).



*Figure 16: ~90% accuracy achieved using class balanced datasets*



*Figure 17: ~72% accuracy achieved using raw (unbalanced) datasets*

The results of the work order classification experiments are summarised below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 10: work order classification accuracy*

**[7.4.3] Time-series experiments that predict TRC features**

Rahul’s summary

**[7.5] Feature transformation experiments**

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:

1. the most recent TRC was retained to preserve currently known information regarding rail geometry, and
2. derived features intended to capture the rate of change over prior TRC runs, calculated as:

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.

The results of the feature transformation experiments are summarised below:

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 approach

**[7.6] Experiments assessing prediction accuracy in longer term horizons**

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 Forest2 | 74.19% | 1.29 | 70.93% | 0.15 |

Note 2: Random Forest used 9 features, unscaled data.

*Table 12: RF Regression outperformed the baseline for 2-quarter horizon*

Unlike the baseline method, Random Forest Regression produced predictions for 2 future quarters that could be useful (Figure 18).

|  |  |
| --- | --- |
| **Baseline prediction: Combinedi(t) = Combinedi(t-1)** | **Random Forest Regression (9 features)** |
| C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\6D73C941.tmp | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\CF384425.tmp |
| C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\BB149DF7.tmp | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3FD3AE7B.tmp |

*Figure 18: RF Regression outperforms the baseline prediction in 2-quarter horizon*

**[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 13: assessment of the usability of visualisations*

Starting from the premise of user-driven software development, we built our visualisations and interactivity based on what the QR engineers would find helpful. A save function is provided on every interactive plot for users to adjust their plot view until it’s focused on what they want, before conveniently taking a snapshot for subsequent use. A balance is struck between displaying extensive plots that allow users to find patterns in their data, and website load time. Optimisation methods such as caching html pages for display instead of having Django rebuild the html for every access, merging low-value heatmap points into the background, and serialising large plots as json for Django to load directly, are performed to further reduce loading time, a non-trivial task considering the combined TRC-C195 file alone is nearly half a Gb in size.

In addition to catering to non-technical users, we expect QR’s software engineers to integrate our tools into their current toolkits. These include both deploy-as-is on their servers, and further development work. For the former, the software is packaged in a Docker container that allows for trivial cross-platform deployment, with clearly-marked data file paths to replace with their own files for use on other rail sections. For the latter, we provide standard build instructions on a Linux platform to compile on their systems, and fairly decoupled code if they wish to make changes. We would provide test suites, linters and better code documentation, and build instructions in other platforms in future.

**[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 14: 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 |
|  |  |
| ??Rahul summary time-series | ?? |
| 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 15: summary of key findings*

**[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 machine learning algorithms, Django app development |
| **Chowdhury** | Visualisations, time-series algorithms |
| **Yin** | Visualisations, Bokeh web app development, webhosting |
| **Salouk** | Interface with Queensland Rail, machine learning algorithms |

*Table 16: key team member roles*

## **Appendix 1 - 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." *Machine learning* 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 machine learning 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 Machine Learning 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 2 - 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 |
| 1. Alignment of GPR to TRC | using meterage measures |
| 1. Calculate standard deviations and the Combined metric | across 20 metre sections of TRC measures |
| 1. Alignment of drainage points | using meterage measures |
| 1. Train/test split | the dataset was split into training (75%) and test (25%) sets |
| 1. Standardisation | feature data was standardised to mean 0, standard deviation of 1 (as required by models such as K-NN and SVR) |

**Appendix 2 – Machine Learning 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 | | BDMRight | | BVMCentreCategory | | BVMCentreVolume | | BVMLeftCategory | | BVMRightCategory | | Drainage | | LRICentre | | LRILeft | | SDTopLeft1 | | SDTopLeft2 | | SDTopLeft3 | | SDTopRight2 | | SDTopRight3 | | SDTwist103 | | SDTwist33 | | SDVersL3 | | SDVersR3 | | TDILeft | | |  | | --- | | BTILeft | | BVMLeftCategory | | BVMLeftVolume | | BVMRightCategory | | LRICentre | | LRILeft | | PVCCentre | | PVCLeft | | PVCRight | | SDTopLeft1 | | SDTopLeft2 | | SDTopLeft3 | | SDTopRight1 | | SDTopRight3 | | SDTwist101 | | SDTwist103 | | SDTwist33 | | SDVersL1 | | SDVersL3 | | TDILeft | | |  | | --- | | PVCCentre | | PVCLeft | | PVCRight | | SDTopLeft3 | | SDTopRight3 | | SDTwist101 | | SDTwist103 | | SDTwist33 | | SDVersL1 | |

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

## **Random Forest Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning 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.

|  |  |
| --- | --- |
| **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\66BC317.tmp** | **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B561743D.tmp** |
| **Random Forest: 9 features** | |

**Support Vector Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning 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.

|  |  |
| --- | --- |
| **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\E9A7CF4F.tmp** | **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A9A535.tmp** |
| **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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\4E0E820B.tmp | | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\2DD13351.tmp | | | |
| **Machine Learning 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. | C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\99AFF52D.tmp |

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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning 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.

|  |  |
| --- | --- |
| **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\10C546C9.tmp** | **C:\Users\Marcus\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\82BD5CBF.tmp** |
| **ANN: 20 features** | |