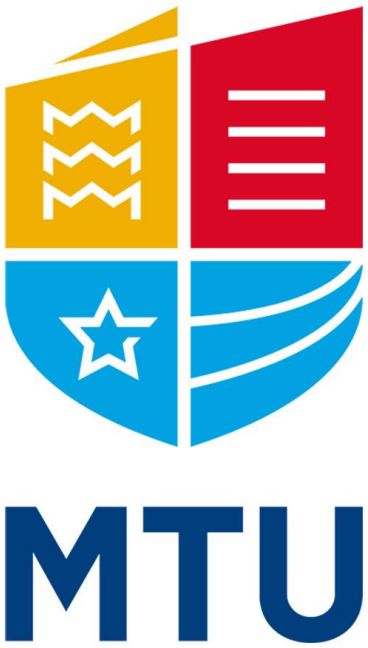

Student Name: PREM VIKAS

Student No: R00241672

A Review of DataCamp's Deep Learning Courses and Machine Learning and Prediction of Radio Masts.

For the module DATA9005 as part of the

Master of Science in Data Science and Analytics, 2023/24

Declaration of Authorship

I, PREM VIKAS, declare that the work submitted is my own.

- I declare that I have not obtained unfair assistance via use of the internet or a third party in the completion of this examination
- I have not used ChatGPT, an AI chatbot or other similar software in this assignment
- I have acknowledged all main sources of help
- I acknowledge that the Academic Department reserves the right to request me to present for oral examination as part of the assessment regime for this module.
- I confirm that I have read and understood the policy and procedures concerning academic honesty, plagiarism and infringements
- I understand that where breaches of this declaration are detected, these will be reviewed under MTU (Cork) policy and procedures concerning academic honesty, plagiarism and infringements, as well as any other University regulations and policies which may apply to the case. I also understand that any breach of academic honesty is a serious issue and may incur penalties.
- EXAMINATION/ASSESSMENT MATERIAL MAY, AT THE DISCRETION OF THE INTERNAL EXAMINER, BE SUBMITTED TO THE UNIVERSITY'S PLAGIARISM DETECTION SOLUTION
- Where I have consulted the published work of others, this is always clearly attributed
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this work is entirely my own work

Signed: PREM VIKAS

Date: 18-05-2024

Question 1.

I recently completed two comprehensive courses on DataCamp: "Introduction to Deep Learning with Keras" and "Advanced Deep Learning with Keras." These courses have significantly enriched my understanding and skills in deep learning, and I am eager to share my reflections.

Introduction to Deep Learning with Keras.

The introductory course was a perfect blend of theory and hands-on practice. It covered essential concepts such as regression, binary classification, and multi-class classification. The projects were particularly engaging; predicting asteroid trajectories and distinguishing between real and fake dollar bills made the learning experience both practical and exciting. I appreciated the clear explanations of neural network fundamentals and the practical insights into building and tuning models. Visualizing training metrics and learning about model optimization provided me with the tools to evaluate and improve my models effectively.

Advanced Deep Learning with Keras.

The advanced course took my understanding to a new level by introducing the Keras functional API. This course emphasized the versatility and power of more complex model architectures. I found the sections on building models with multiple inputs and outputs particularly valuable. The hands-on exercises involving categorical embeddings, shared layers, and merge layers were challenging yet rewarding. I also enjoyed the practical examples of multi-task learning, where models performed both classification and regression. These exercises helped me grasp the nuances of advanced model design and implementation.

Overall, both courses were well-structured and provided a thorough grounding in deep learning with Keras. The balance between theoretical concepts and practical applications was perfect. I now feel more confident in my ability to design, implement, and optimize deep learning models for various tasks.

(Word Count: 270)

Question 2.

1.Introduction

In the rapidly evolving telecommunications industry, maintaining robust and uninterrupted service is crucial. Radio Frequency (RF) radio masts, integral components of mobile communication networks, are susceptible to outages caused by adverse weather conditions, such as heavy rain and mist. These outages not only disrupt service but also impact customer satisfaction and operational efficiency. To address this challenge, Domhnall, the manager of a nationwide telecoms company, seeks to harness the power of analytical techniques and machine learning to predict the susceptibility of RF radio masts to such weather-induced disruptions.

The primary objective is to develop a predictive model using the labeled training dataset, which can then be applied to the unlabeled scoring dataset to ascertain the engineering status of each mast. By doing so, the company aims to proactively manage and mitigate the risk of outages, thereby enhancing the reliability of its network. This project not only represents a technical challenge but also a significant opportunity to integrate machine learning within operational processes to yield tangible benefits in network management and service delivery. The ensuing analysis will involve cleaning the datasets, exploring the data to identify patterns and correlations, and applying suitable machine learning algorithms to develop a robust predictive model. This initiative is pivotal in transitioning from reactive to proactive management in telecom infrastructure maintenance, ensuring that service reliability is upheld even in the face of adverse environmental conditions.

1.1 Data Overview.

The training dataset consists of 2186 records, each described by 79 attributes that include technical specifications and metadata about RF radio masts. Key attributes Eng_Class, a categorical label indicating whether a mast is 'okay' or 'under' engineered which is our target variable in building a model and detailed antenna-related features such as AntennagaindBd1 and AntennagaindBd2, among others. The data types within the dataset are predominantly float64, used for technical measurements, alongside several object (string) fields used for identifiers and categorical data. The dataset shows a high variability in technical specifications, evidenced by the range of unique values across attributes, suggesting that each mast is measured with specific and detailed technical parameters.

The scoring dataset comprises 936 records, detailed across 77 attributes, which notably exclude the Eng_Class and Outcome fields found in the training dataset. It shares most attributes with the training dataset, predominantly focusing on antenna gains, heights, and other critical technical parameters relevant to the RF radio masts. Similar to the training dataset, the majority of data types are float64, which are used for recording precise technical measurements. The dataset also maintains a consistent diversity in technical specifications across its records, providing detailed and varied information crucial for the effective prediction of mast conditions in prediction tasks.

2. Exploratory Data Analysis.

To develop a predictive model for assessing the susceptibility of RF radio masts to adverse weather conditions, we conducted a comprehensive exploratory data analysis (EDA) and data preprocessing.

2.1 Data Pre-processing.

- **Loading the Dataset:** We began by loading the training and scoring datasets from the specified paths using the pandas library, ensuring the data was readily accessible for processing.
- **Standardizing Column Names:** To facilitate easier data manipulation, we standardized the column names across both datasets. This involved converting all column names to lowercase and replacing spaces with underscores, making the data more uniform and easier to handle programmatically. We addressed any inconsistencies in column naming within the datasets, such as renaming columns to ensure uniformity. This step is crucial for merging data or comparing columns later in the analysis.
- **Handling Missing Values:** A critical step in our EDA was identifying missing values within the datasets. Initially there 3 columns dpq_r2, fullmin and outcome in training dataset with missing values of 9, 6, 105 respectively. Since the percentage of missing values if columns dpq_r2 an fullmin was very less, those rows was dropped using dropna() function and the outcome column was dropped from the dataset since it was redundancy of the eng_class columns.
- **Dropping Irrelevant Columns:** Similarly, the antennafilename1 and antennafilename1 was also dropped from the dataset as it was irrelevant in training our model. A total of **75** is remaining after this approach.
- **Removing Highly Correlated Features (Numeric):** To understand the relationships between numerical features, we calculated a correlation matrix. This helped us identify highly correlated features, which can lead to multicollinearity issues in predictive modelling and can also mitigate the issues caused by redundant data in addressing the aspects of **bijection**. Features with high correlations (above a predefined threshold of 0.90) were removed from the dataset to reduce redundancy and improve model performance. After this step a total of **53** columns remained in our training dataset.
- **Identifying Duplicate Columns:** We used a custom function to identify any duplicate columns within the dataset. The function to identify duplicate columns in a dataset by comparing each column with every other column. If two columns are identical across all rows, they are considered duplicates and noted. Once identified, specific duplicate columns deemed redundant are listed and removed from the dataset, streamlining it by eliminating unnecessary redundancy. A total of **49** columns was left after this action with int and float type columns.
- **Removing Highly Correlated Features (Categorical):** To analyse the relationships between categorical features in a dataset using Cramer's V statistic, which is a measure of association between two nominal variables. Since the basic cor-relation function cannot be used categorical, we use the Cramer's V Calculation Function and

initialize the DataFrame to store Cramer's V values between all pairs of categorical columns and Iterating over pairs of categorical columns and computes Cramer's V for each pair, excluding self-comparisons, and stores the results in the matrix to identify Highly Correlated Features i.e. correlation of 88% and above, then dropping them from the dataset. Removing highly correlated categorical features reduces the risk of overfitting and increases model generalizability. A new total of **40** columns in the training dataset.

- **Encoding Categorical Columns:** Categorical variables were encoded using label encoding to convert them into a format that could be easily used by machine learning algorithms. This step was crucial for preparing the data for model training. A function is created which iterates over columns that are of the object data type, which typically includes text or mixed data types. For eligible columns, the text data is first converted to string type to ensure consistency. The LabelEncoder from sklearn.preprocessing is then applied to transform the textual data into a numeric format. Each unique category in the column is assigned a unique integer. This is done for both training and scoring data to maintain the consistency between the two datasets.
- **Encoding target variables:** In the above step when we encoded categorical variables, the target variable eng_class was also treated resulting a numeric value 1 for the term ‘okay’ and numeric value 2 for the term ‘under’ is encoded.
- **Final Preparation and Clean-up:** Before starting to split the data for modelling one last check for any zero values row in the target variable is done and 3 zero values was found. The removal of those rows was performed as it would alter the results while model building and prediction. (Word Count: 740)

3. Methodology.

This section outlines the methodology used to train, tune, and evaluate three different machine learning models to predict the engineering class ('okay' or 'under') of RF radio masts based on their attributes. The goal is to identify the best performing model based on accuracy and other relevant metrics.

3.1 Feature and Target Preparation.

- From the fully encoded dataset, features (X) are separated from the target variable (y). The target variable eng_class indicates the susceptibility of radio masts to adverse conditions, and the features consist of all other columns after encoding categorical variables into numerical formats.
- The dataset is split into training (X_train, y_train) and testing sets (X_test, y_test) using a stratified approach to ensure that both sets are representative of the overall dataset. The test size is set to 20% of the total data, maintaining sufficient data for both training and validation.

3.2 Model Selection.

- **Random Forest Classifier:** A robust ensemble technique that uses multiple decision trees to make predictions. Initialized with 100 trees (n_estimators=100), providing a good balance between training time and model complexity.
- **Logistic Regression:** A statistical model for binary classification problems. It is initialized with max_iter=1000 to ensure convergence and the 'saga' solver, which is efficient for large datasets and supports various penalty types.
- **Gradient Boosting Classifier:** An ensemble technique that builds sequential models, focusing on correcting the errors of the previous models. It uses a decision tree as the base learner and is set with a random state for reproducibility.

Using a combination of these models can be particularly powerful. While Logistic Regression offers a baseline understanding and quick implementation for binary classification, Random forest add interpretability regarding feature importance and decision criteria. Gradient Boosting builds upon these by iteratively improving model accuracy and handling the misclassifications effectively. This combination allows for robust generalization across different types of data scenarios and provides a comprehensive toolkit for tackling binary classification problems with nuanced complexities like those seen in predicting mast susceptibility.

3.3 Model Evaluation.

The performance of the three models—Random Forest, Logistic Regression, and Gradient Boosting—has been evaluated based on several metrics: accuracy, precision, recall, and the F1-score for two classes (labelled as 1 for 'okay' and 2 for 'under'). Below is an in-depth analysis of the results and a rationale for selecting Gradient Boosting as the best model for this task.

Random Forest Performance:

- Precision: High for class 1 (0.86) and reasonably good for class 2 (0.78).
- Recall: Excellent for class 1 (0.96) but moderate for class 2 (0.47).
- F1-Score: Strong performance for class 1 (0.91) indicates good balance between precision and recall, whereas for class 2 it is lower (0.59) due to the lower recall.
- Accuracy: Overall, the model achieves 85.23%, which is a solid performance.

Logistic Regression Performance:

- Precision: Lower than the Random forest for both classes, with 0.79 for class 1 and significantly lower at 0.53 for class 2.
- Recall: Very high for class 1 (0.98), yet very low for class 2 (0.09), indicating a skew towards predicting class 1 correctly but failing to identify class 2 effectively.
- F1-Score: Decent for class 1 (0.87) but poor for class 2 (0.15), reflecting the imbalance in recall.
- Accuracy: At 78%, it is the lowest among the three models, reflecting its struggle to adequately model the diversity within the classes, especially the minority class.

Gradient Boosting Performance:

- Precision: Similar to the Random Forest for class 1 (0.86) and higher for class 2 (0.84), indicating very effective classification for both classes.
- Recall: Very high for class 1 (0.98) and nearly balanced for class 2 (0.45), showing an improvement over both the Random Forest and Logistic Regression in handling class 2.
- F1-Score: High for class 1 (0.91) and moderately good for class 2 (0.58), showing better overall balance in classification performance than the other two models.
- Accuracy: The highest at 85.2%, indicating that this model most effectively combines accuracy with balance across class identification.

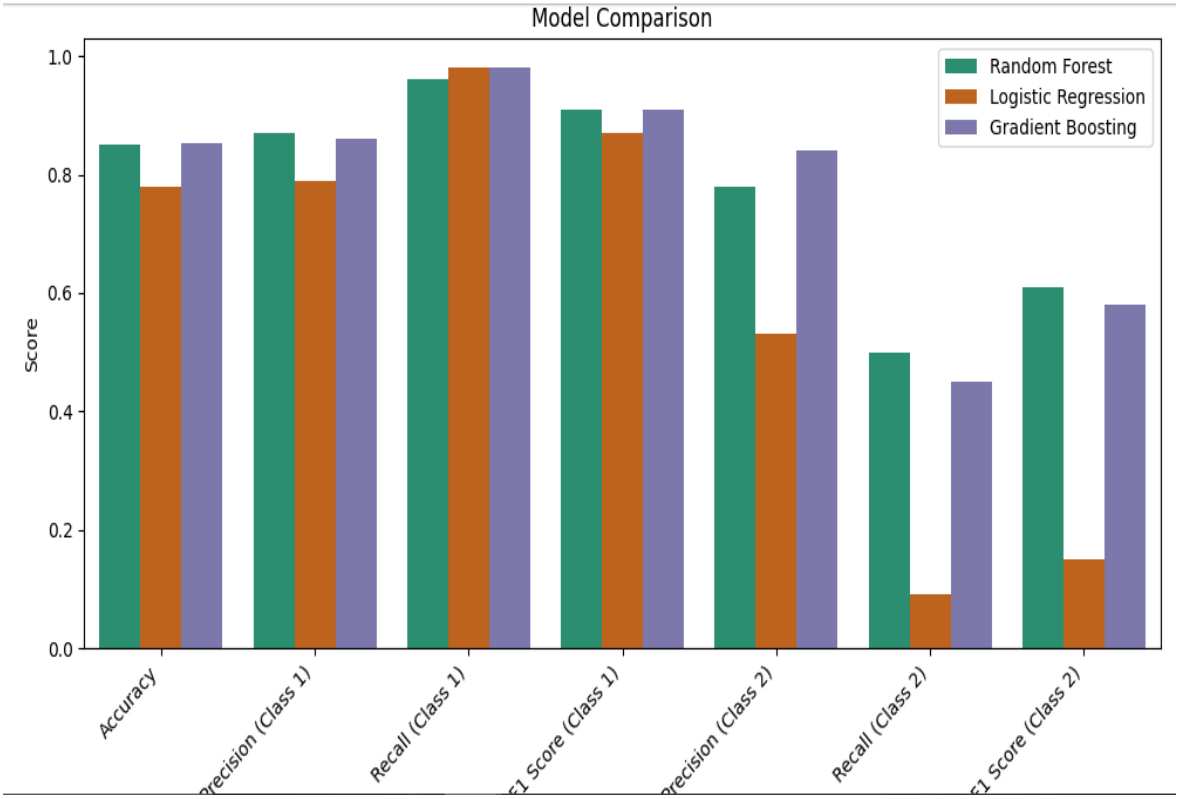

Figure 1. Graphical Analysis for Model Comparison.

Considering the higher accuracy, balanced class performance, and robustness in handling complex data patterns, Gradient Boosting stands out as the most suitable model for predicting the susceptibility of RF radio masts to adverse weather conditions in this case study.

3.4 Hyper-parameter Tuning with Grid Search.

- **Defining the Parameter Grid.**

A parameter grid is set up to specify different values for key hyperparameters of the Gradient Boosting model:

n_estimators: Number of trees in the ensemble. More trees can lead to better performance but might increase computation time and risk overfitting.

learning_rate: Shrinks the contribution of each tree and can lead to a more robust model. Lower rates typically require more trees but can generalize better.

max_depth: Maximum depth of individual trees. Deeper trees can learn more complex patterns but might overfit.

subsample: The fraction of samples to be used for fitting individual base learners. Less than 1.0 leads to a reduction of variance and an increase in bias.

min_samples_split: The minimum number of samples required to split an internal node. Higher values prevent the model from learning overly specific patterns, thus lowering overfitting.

- **Initializing the Gradient Boosting Classifier.**

A base Gradient Boosting Classifier is initialized with a fixed random state to ensure reproducibility.

- **Performing Grid Search with Cross-Validation.**

Grid Search: This technique systematically works through multiple combinations of parameter values, cross-validating as it goes to determine which tune gives the best performance.

Cross-Validation: The dataset is split into k smaller sets (folds). The model is trained on k-1 of these folds, with the remaining part used as a test set to compute a performance measure. This process is repeated k times.

Execution: The grid search is configured to use 5-fold cross-validation (cv=5), assessing all combinations (243 total fits) across the defined grid, running in parallel (n_jobs=-1) for faster execution and providing detailed logs (verbose=2).

- **Evaluating Results.**

Best Parameters and Score: The best performing set of parameters (best_params) and the highest cross-validation accuracy (best_score) are identified from the grid search.

Training on Best Parameters: A new Gradient Boosting model is instantiated with the best-found parameters and fitted on the entire training dataset.

```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits
Best Parameters: {'learning_rate': 0.01, 'max_depth': 3, 'min_samples_split': 10, 'n_estimators': 200, 'subsample': 0.8}
Best Cross-Validation Score: 0.8607690195001373
Accuracy: 0.8523002421307506
Classification Report:
              precision    recall  f1-score   support

           1       0.85      0.99      0.91       321
           2       0.92      0.37      0.53        92

    accuracy                           0.85       413
   macro avg       0.88      0.68      0.72       413
weighted avg       0.86      0.85      0.83       413
```

Figure 2. Output of Hyper-parameter Tuning.

The hyperparameter tuning process led to the identification of an optimal set of parameters that achieved a cross-validation score of approximately 86.08% and a test accuracy of 85.25%. While precision is generally high, recall for class 2 is significantly lower, indicating potential issues model sensitivity to class 2 features.

4. Feature Selection.

The feature selection process using the tuned Gradient Boosting model aimed to identify and focus on the most influential features that impact the prediction of RF radio mast susceptibility. This approach was intended to optimize model complexity and potentially enhance model performance.

- **Train the Model with Optimal Parameters**.
  A Gradient Boosting Classifier was initialized with previously determined best parameters and fitted to the entire training dataset.
- **Extract and Rank Feature Importance's**.
  Feature importance's were extracted from the trained model. These importance's represent the relative contribution of each feature to the model's prediction accuracy. A DataFrame was created to store these importance's alongside their corresponding feature names, sorted in descending order to highlight the most significant features.
- **Visual Representation of Feature Importance's**.
  The feature importance's were visualized using a horizontal bar chart, providing a clear and intuitive depiction of each feature's contribution to the model.
- **Select Top N Features**.
  Based on the feature importance rankings, the top N features were selected. This subset of features is considered to have the most impact on the model's predictions. The training and testing datasets were then subset to include only these top features, reducing the dimensionality and focusing the model on the most relevant predictors.
- **Retrain the Model with Selected Features**.
  The Gradient Boosting Classifier was retrained using only the top features to assess whether focusing on these key predictors would maintain or improve model performance.

```
Accuracy with top features: 0.8547215496368039
Classification Report with top features:
              precision    recall  f1-score   support

           1       0.85      0.99      0.91       321
           2       0.92      0.38      0.54        92

    accuracy                           0.85       413
   macro avg       0.88      0.69      0.73       413
weighted avg       0.86      0.85      0.83       413

Confusion Matrix with top features:
[[318   3]
 [ 57  35]]
```

Figure 3. Output of Feature selection.

The confusion matrix showed that while the model is excellent at identifying class 1 (true positives), it struggles somewhat with class 2, missing a significant number of true class 2 predictions (false negatives).

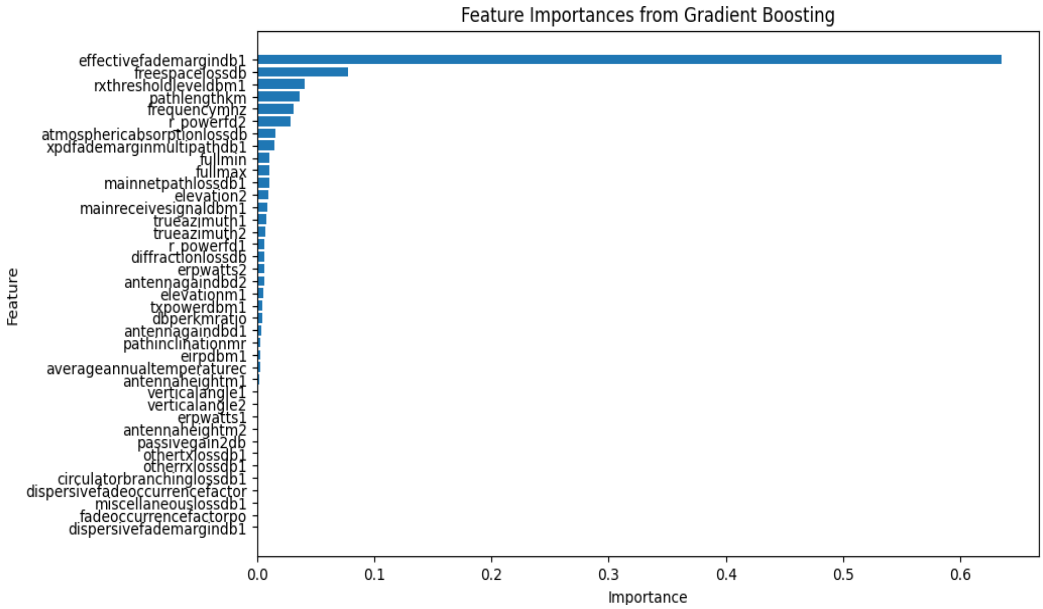

Figure 4. Feature Importance's from Gradient Boosting.

The retrained model achieved an accuracy of approximately 85.47%, slightly above the accuracy obtained with the full set of features (85.2%). However, the reduction in complexity by using fewer features may benefit the model by reducing overfitting and improving generalization on new, unseen data. Precision, recall, and F1-scores, especially for class 2, indicate that the model, while slightly less effective in predicting class 2 accurately, maintains a reasonable balance between error types.

The feature selection was performed because of following reasons:

Reduce Model Complexity: Simplifying the model can decrease training time and improve model interpretability.

Avoid Overfitting: By reducing the number of features, the model may generalize better to new data, as it must focus on the most informative attributes.

5. Model Explanation and Cost/loss Function.

Gradient Boosting is a powerful machine learning technique that combines multiple weak prediction models, into a stronger model in a sequential manner. The primary goal of this ensemble technique is to convert many weak learners (models that perform slightly better than random guessing) into a collectively strong learner that achieves high accuracy in prediction tasks.

5.1 How Gradient Boosting Works.

Initialization:

The model starts with a single weak learner, usually a logistic regression to make initial predictions. These predictions are typically very simple, often resulting in substantial errors or residuals (the difference between the predicted and actual values).

Iterative Improvement:

The algorithm then enters a loop where it continually tries to improve upon the residuals left by the previous model.

In each iteration, a new decision tree is trained, but instead of learning from the original data, it learns to predict the residuals from the previous model. The idea is to stepwise minimize these errors.

Combining the Learners:

Each tree is added to the ensemble with an associated weight that helps to reduce the overall prediction error. The weighting of each tree is influenced by a parameter called the learning rate, which controls how fast the model learns. A lower rate requires more trees to be effective but typically results in a more robust model that generalizes better.

Stopping Criteria:

The process continues until a predetermined number of trees are added, or the improvement in model accuracy becomes negligible. This prevents the model from overfitting.

5.2 Cost/Loss Function.

Gradient Boosting specifically minimizes a cost function during training. In classification tasks, such as predicting if a radio mast is 'okay' or 'under', the loss function often used is the logarithmic loss (log loss), defined as:

$$\text{Log Loss} = -1/N \sum N[y_i \log(p_i) + (1-y_i)\log(1-p_i)]$$

Where:

N is the number of samples.

y_i is the actual label of the ith sample.

p_i is the predicted probability of the ith sample being in the positive class.

This loss function penalizes incorrect classifications by comparing the predicted probability assigned to the correct class with the actual class. The closer the probability is to 1 for the actual class, the lower the loss; the farther it is (especially toward 0), the higher the loss, exponentially growing as predictions approach the wrong class with high certainty.

The primary costs associated with using Gradient Boosting involve computational resources and time. Training this model can be computationally expensive, especially with large datasets and a high number of iterations (trees). However, the trade-off typically favors better performance and deeper insights into feature impacts, which can be crucial for strategic planning and operational efficiency.

Conclusively, the choice of Gradient Boosting is justified by its robustness, accuracy, and the detailed insights it offers into the factors affecting mast reliability. This knowledge allows for targeted improvements and preventive measures, potentially saving costs associated with unexpected failures and service interruptions.

5.3 Why this model.

- Handling Imbalanced Data:
  Gradient Boosting inherently manages class imbalance by focusing more on harder-to-classify instances, which are typically under-represented classes in many datasets.
- Feature Importance:
  An intrinsic benefit of using Gradient Boosting is its ability to rank features by their importance in making predictions. This is particularly useful for understanding which aspects of a radio mast's design or environment most significantly impact its susceptibility to adverse weather conditions.
- Predictive Power:
  Due to its iterative correction of errors, Gradient Boosting often results in higher accuracy and better handling of varied data patterns than other models, making it particularly suitable for complex problems like predicting mast failures.
- Scalability and Flexibility:
  Gradient Boosting can handle different types of predictor variables and relationships, making it adaptable to diverse datasets and scenarios without extensive pre-processing.

6. Cost-sensitive

For each cost ratio (1:5, 1:10, and 1:20), the model was adjusted to penalize the misclassification of under-engineered masts ('2') more heavily compared to misclassifying okay masts ('1'). Here's how the model performance and results can be interpreted given the correct class labels:

- **For cost ratio 1:5:**

Confusion Matrix: 298 true negatives (correctly predicted okay), 23 false negatives (under predicted as okay), 43 false positives (okay predicted as under), 49 true positives (correctly predicted under).

Metrics.

Accuracy: 84.02% shows good overall performance.

Precision for 'under' (Class 2): High precision indicates that when the model predicts a mast as under-engineered, it is likely correct.

Recall for 'under' (Class 2): The recall rate for identifying under-engineered masts is relatively low, suggesting some under-engineered masts are still being missed.

- **For cost ratio 1:10:**

Confusion Matrix: More aggressive in predicting masts as under-engineered, evident from increased false positives (okay predicted as under) and decreased false negatives.

Metrics.

Accuracy: 78.93% indicates a decrease due to the model's aggressive prediction of under cases.

Precision for 'under': Remains high, meaning the predictions of under-engineered are often correct.

Recall for 'under': Improved slightly, indicating fewer under-engineered masts are missed compared to the 1:5 ratio.

```
Results for cost ratio 1:5
Confusion Matrix:
[[298  23]
 [ 43  49]]
Accuracy: 0.8401937046004843, Precision: 0.873900293255132, Recall: 0.9283489096573209, F1 Score: 0.9003021148036254

Results for cost ratio 1:10
Confusion Matrix:
[[272  49]
 [ 38  54]]
Accuracy: 0.7893462469733656, Precision: 0.8774193548387097, Recall: 0.8473520249221184, F1 Score: 0.8621236133122028

Results for cost ratio 1:20
Confusion Matrix:
[[204 117]
 [ 27  65]]
Accuracy: 0.6513317191283293, Precision: 0.8831168831168831, Recall: 0.6355140186915887, F1 Score: 0.7391304347826086
```

Figure 5. Output of Cost ratio Analysis.

- **For cost ratio 1:20:**

Confusion Matrix: Significantly more aggressive in predicting masts as under-engineered, with a substantial increase in false positives.

Metrics.

Accuracy: Drops to 65.13%, reflecting the trade-off of aggressively trying to catch all under-engineered masts.

Precision for 'under': Still high, which is good, but comes at the cost of many okay masts being incorrectly labeled.

Recall for 'under': The best among the ratios, indicating a high sensitivity in detecting under-engineered masts.

6.1 Analysis and Recommendation.

As the cost ratio increases, the model becomes more focused on not missing any under-engineered masts, even at the expense of incorrectly predicting many okay masts as under-engineered. This results in:

Higher Recall for 'under' (Class 2): Ensuring that almost no under-engineered mast goes undetected.

Decrease in Precision for 'okay' (Class 1): Many okay masts are wrongly identified as under-engineered, which could lead to unnecessary inspections or maintenance.

6.3 Recommendation for Domhnall.

Choosing the right cost ratio depends on the operational tolerance for false positives (okay masts predicted as under). If the operational cost of missing an under-engineered mast is very high, a higher ratio like 1:20 may be justified despite its lower overall accuracy.

However, a balanced approach such as a 1:10 ratio might provide a more practical compromise, improving the detection of under-engineered masts while maintaining reasonable accuracy and reducing the risk of unnecessary interventions.

In conclusion, adjusting the model to prioritize the detection of under-engineered masts aligns well with the operational priorities, ensuring that the most critical failures are detected promptly while managing the trade-offs involved in predictive accuracy and operational efficiency. (Word Count:510)

7. Prediction on Scoring Dataset.

After training the Gradient Boosting Classifier with the best parameters identified earlier, the model was used to predict the engineering class of RF radio masts in the scoring dataset. The aim was to classify each mast as either 'okay' or 'under engineered', based on the features provided.

- **Model Training:**

The model was trained using the entire training dataset (X, y) with optimal parameters to ensure it learned the most effective patterns and relationships. The parameters used include a low learning rate for gradual learning without skipping over important details, moderate tree depth to capture sufficient complexity, and a subsample rate that helps in reducing variance and improving model robustness.

- **Prediction on Scoring Dataset:**

The trained model was then applied to the scoring dataset to predict the engineering class of each mast. It's important that the features in the scoring dataset were aligned with those in the training dataset to ensure compatibility and accuracy in predictions.

- **Integration of Predictions:**

Predictions were integrated back into the scoring dataset as numerical labels. These labels were then mapped to human-readable strings (**'okay'** for **1**, **'under engineered'** for **2**) for clarity and ease of interpretation.

- **Analysis of Predicted Classes:**

The final predictions were analysed to determine the distribution of predicted classes. The results indicated that out of **936** masts, **862** were predicted to be **'okay'** and **74** were predicted to be **'under engineered'**.

7.1 Interpretation of Results.

The model predicts that the vast majority of masts are in good condition ('okay'), with a smaller portion deemed to be 'under engineered'. This suggests that most of the masts are expected to withstand adverse weather conditions effectively, while a few may require attention to avoid potential failures.

Operational Efficiency: The predictions can help prioritize inspections and maintenance. Masts classified as 'under engineered' can be targeted first to prevent outages, optimizing resource allocation and potentially reducing operational costs.

Preventive Measures: By identifying potentially vulnerable masts, preventive measures can be more effectively implemented, enhancing the overall reliability of the network.

7.2 Additional Uses of Scoring Data.

Beyond classifying the masts, the scoring data can be leveraged in several other ways:

Trend Analysis:

Analysing patterns or trends in the features associated with under-engineered masts can help in identifying common factors contributing to vulnerabilities.

Feature Engineering:

Further analysis could lead to the discovery of new features or interactions between features that significantly impact mast performance, which could improve model accuracy or provide insights for engineering improvements.

Geographical Analysis:

Combining mast location data with the predictions could help in geographical plotting to visualize areas with higher risks, aiding strategic planning and regional prioritizations.

Feedback Loop:

As more data becomes available (e.g., actual performance during adverse weather), it can be used to refine the model further. Comparing predicted outcomes with actual outcomes can help in assessing the model's effectiveness and guiding continuous improvement.

Resource Allocation:

Predictive data can guide decisions on where to allocate budget and resources most effectively to enhance network stability and customer satisfaction.

8. Conclusion.

This report detailed a comprehensive analysis of RF radio masts using machine learning techniques to predict their susceptibility to adverse weather conditions, particularly focusing on distinguishing between masts that are well-engineered ('okay') and those that are under-engineered ('under'). Through meticulous data preparation, model selection, and

optimization, we aimed to provide actionable insights to support the operational priorities of Domhnall's telecom company. the project not only addressed the immediate need to identify under-engineered masts but also highlighted the strategic benefits of integrating advanced analytics into routine operational practices. This approach not only mitigates immediate risks but also enhances overall operational efficiency and customer satisfaction in the long term.

9. Future Work.

Continuous Model Refinement: As new data becomes available, continuously updating and refining the model will ensure its relevance and accuracy.

Integration of Additional Data Sources: Incorporating more diverse data, such as real-time weather data and maintenance history, could further improve the predictive accuracy and utility of the model.

Deployment in Operational Processes: Implementing the model within the existing operational workflows through automated scoring and alert systems could help in real-time decision-making and operational responsiveness.

Expansion to Other Infrastructure Components: Applying similar analytical techniques to other critical components of the telecom network could provide a comprehensive risk management framework.