Data Analysis of dataset containing metrics that influence internet speed.

# **Description of the dataset**

## **Dataset Overview**

The dataset that this data analysis report is on is related to metrics that contribute to internet performance. The dataset contains 20,000 entries spread across seven columns which include the average download and upload speed in kilobits per second (avg\_d\_kbps, avg\_u\_kbps) the average latency and the average latency download and upload (avg\_lat\_ms, avg\_lat\_down\_ms, avg\_lat\_up\_ms) and the network type (net\_type).

## **Categorical Data**

The first step of my data cleaning was to encode the nominal categorical data of the network type into a numerical form so it can be processed for ML models. In the dataset there are only two network values either “moblie” or “fixed”. Using binary encoding I turned these into two separate columns net\_type\_fixed and net\_type\_moblie. The column net type “moblie” is a typographical error for “mobile” which requires a name correction to ensure consistency.

## **Missing Values**

The next part was to investigate any missing values within the dataset and to choose a method to handle them. Average latency down contained 903 of the missing values and average latency up contained 873.

Instead of dropping these values which would affect the accuracy of the dataset I opted for the K-nearest neighbors imputation technique which “works by classifying the nearest neighbours of missing values and use those neighbours for imputation using a distance measure between instances missing values” (Emmanuel, 2021). This method reduces bias and has the flexibility to handle mixed data types while also preserving the integrity of the dataset which is crucial for ML/DL processing.

**A graph with numbers and lines

Description automatically generated with medium confidence**

The figure shows a heatmap of the rows where there are missing values.

## **Outliers**

Following the missing values, the next step was to identify outliers and handle them if there are any to mitigate the influence of them. A common approach to identify outliers in a dataset is by using boxplots in the figures below. It can be seen that the distribution is heavily skewed meaning that there is a significant amount of outliers. To solve this the interquartile range (IQR) can be used to set thresholds on the upper and lower quarters of the data’s values to determine what is classed as an outlier and impute those values that exceed those boundaries to fit within.

## **Feature engineering**

The last stage was to engineer new features that may provide further insight and assist the ML/DL algorithms to find patterns. The two new features were “speed\_ratio” and “latency\_ratio” these are features that represent the ratio of download to upload for both speed and latency.

A graph of a graph

Description automatically generated with medium confidence A diagram of a bar graph

Description automatically generatedA graph of a graph

Description automatically generated with medium confidenceA diagram of a graph

Description automatically generated

A graph of a graph of a number of data

Description automatically generated with medium confidence A diagram of a bar graph

Description automatically generatedA graph with numbers and a line

Description automatically generatedA graph with a blue and black line

Description automatically generated with medium confidence

A graph of a graph with numbers and a line

Description automatically generated with medium confidence A diagram of a bar graph

Description automatically generated A blue square with white text

Description automatically generated A graph of a number of objects

Description automatically generated with medium confidence A graph of a speed ratio

Description automatically generatedA diagram of a graph

Description automatically generated A graph of a latency ratio

Description automatically generated A diagram of a line

Description automatically generated

Before and after removal of outliers for each feature showing the distribution of values change

# **Comprehensive Dataset Analysis**

A group of numbers on a white background

Description automatically generated

The comprehensive dataset analysis of the cleaned internet performance metrics reveals the average download speeds mean was 102,277 kbps which is significantly higher than that of its counter part of the average upload speed 23,452 kbps. This finding can be interpreted as the consumers of the internet service provider have a higher demand and use of downloading services rather than uploading services. The median download speed is 53,976 which is below the mean, suggesting that the distribution of the data is right skewed which is a common pattern throughout all of the continuous features. This is further proved by the skewness statistics that add evidence to a right skewed distribution. A takeaway from this is that the majority of the representatives of the data use low internet speeds and may have no reason to pay for the premium to be in the minority of higher speeds.

The mean of the speed ratio feature is 5.956249 which if rounded to six means that on average download speeds are six times as fast as upload speeds. The latency ratio here is 1.275872 showing not much difference in scale between them.

Variance within the dataset shows a notable spread in consumers internet performance which is shown by that of download speeds variance at 11.94billion and upload speed of 569million. Among the latency features upload latency has the highest variance at 652,511. These statistics suggest that the representatives of the data have significant differences in that of the download and upload speed quality.

A table of numbers and symbols

Description automatically generated

Table of the skewness and Kurtosis values for each continuous type

To address the issue of the data being right skewed a normlisation technique of logarithm transformation was used this transformation is widely used in statistical analysis due to its ease of use to normalize data (Changyong FENG, 2014) which is key when processing in ML algorithms. However, this source states that there is no guarantee log transformation can make the data less skewed and could make it more skewed which emphasizes the need to check the distribution of data after transformation.

The mode of the categorical values of net type mobile/fixed can be inferred as the same, as both values have a mean of 0.5. This in turn gives an unbiased comparison between fixed and mobile networks when comparing the performance of each.

A screenshot of a computer

Description automatically generated

Cross tabulation table showing the count of each value in the net type fixed feature.

To explore the relationships between the features I visualised a correlation matrix using a heatmap. The key points shown from this are that download speed and upload speed have a moderate positive correlation (0.72), latency upload and latency download also have a moderate positive correlation (0.58) and lastly latency upload and latency ratio have moderate negative correlation (-0.61). An insight from the correlation matrix suggests that higher speeds do not correlate with lower latency.

A colorful squares with white text

Description automatically generated with medium confidence

The figure shows the correlation values visualised using in a heatmap with stronger correlated values being represented with lighter or darker colours.

The covariance between the download and upload speeds is high at 1,571,304,070 which reinforces the statistic shown by the correlation matrix which further shows the relationship between download and upload speeds.

# **Hypothesis Definition and testing**

The hypothesis chosen to test aimed to determine if there are significant differences in the upload speed over the network types of the data which will allow consumers to have insights over decisions when choosing a network type. This will be performed by statistical analysis to calculate the z statistic. The z-test was chosen over the t-test as its suitable for “testing significance of single population mean and difference of two population means if our sample is taken from a normal distribution with known variance or if our sample size is large enough to invoke the Central Limit Theorem (usually𝓃 ≥ 30 is a good rule of thumb).” (ECRTD-UK, 2020).

## **Hypothesis**

The null hypothesis is that there is no significant difference in upload speeds across fixed and mobile networks.

The alternative hypothesis is that there is a significant difference in upload speeds across fixed and mobile networks.

## **Methodology**

To conduct the testing of the hypothesis on the dataset the data has to be split based on each network type into two groups. For each of the groups the mean and standard deviation were calculated with the sample size to measure the number of standard deviations between the group means. Following this calculation, a low p-value would indicate that the difference in upload speeds is statistically significant enough not to have happened by chance and would lead to rejecting the null hypothesis.

## **Testing and results**

After running the test, a z-statistic of 60.748 was concluded which immediately indicates a great difference between the group means of the data. The p-value gave a result printed as 0.0, this is interpreted as a value close to zero and far below the alpha threshold of 0.05 which is used to tell significant differences.

From the results the null hypothesis can be rejected, and the alternative can be accepted as its statistically improbable that upload speeds are not affected by the network type and that network type of fixed or mobile have a significant enough impact on upload speeds.

## **Conclusion**

The testing conducted provided with a high confidence that upload speeds are affected by network type. This outcome can help consumers when choosing a network between mobile and fixed. The use of the z-test in this hypothesis proved to be appropriate through the large sample sizes proving a high degree of confidence in the testing. This confidence is key as if other statistical method were used the results of this test could be different and affect the real-world applications of them leading consumers to think there is no significant difference between the network types of upload speeds.

# **Model implementation and analysis**

In my analysis I used the implementation of machine learning models to predict features of the dataset which may provide useful insights. For both regression and classification solutions I opted for gradient boosters after reading that they are the leading techniques for regression and classification problems (Gert-Jan-Streefland, 2023) and are favored in many spaces such as Kaggle competitions to business solutions by minimizing the bias error of a model (Saini, 2023).

For the regression solution for predicting download speeds, I opted to compare this model to a deep learning model. This will determine if traditional models or deep learning models will yield higher accuracy and speed and the implications of that on real world uses. The second prediction will be to predict the network type based on the rest of the features of the database which will be predicted using a gradient boosted classification.

## **Predicting Download speed with regression**

### **Model evaluation**

After training the XGBRegressor model on the logged data I used metrics to analyse how accurate it was on my dataset. The first metric I used was the MSE (mean squared error) which after unlogging came to 1.079 billion which when rooted comes to 32,850. This is a significant amount of difference on average of each data point and the graph below corroborates to that as there are more false predictions when navigating further along the graph. A reasoning behind this may be due to the biased distribution of the data used to train the model which leads to the “algorithm tending to systematically learn the wrong signals by not considering all the information within the data.” (Barcelos, 2022). As there were fewer high download speeds leading to lower accuracy when the model tries to generalise the data.

A diagram of a blue line

Description automatically generated

Figure shows accuracy of predicted values against the actual values of the XGBRegressor model.

Another accuracy metric used was the R2 score which came to 91.07% which shows that the model fits well and that features used to predict the download speed captured the variation of the data. The coefficients shown in the figure show which features have the strongest correlation to the download speed which align with the correlation matrix previously visualised.

A screenshot of a computer

Description automatically generated

Table showing the coefficients of each feature used to train the model.

### **Alternate solutions**

Alternate models to compare this include traditional regression libraries such as scikit-learn, and deep learning solutions provided by tensorflow. Starting with the traditional model the same data preparations as XGBRegressor were carried out before training and after training got a MSE score of 3.49billion which shows that it did not beat the score of the Gradient booster model as expected and by that of a significant amount, it also has an inferior R2 score of 70.79% showing that it had a worse fit to the data. The deep learning model achieved a MSE of 1.4billion and an R2 score of 88.2% which is an improvement over that of the traditional model however is second to that of the gradient booster. When using neural networks considering the computational cost is key as with each new entry to the database, the model will have to be retrained which makes them less practical in real world applications (Tan, 2006). However, if this is not a factor to consider they may be more accurate if left to train over enough time.

## **Predicting Network type with classification**

The next prediction model I wanted to implement was a network type predictor that can predict if the network is likely to be “fixed” or “mobile” this would use a classification model as the data is of categorical origin.

A screenshot of a graph

Description automatically generated

### **Model evaluation**

As stated, before that gradient boosters are superior, I used the XGBClassifier model to predict the network type. After training the accuracy metrics revealed a good accuracy of 86.95% which indicates the percentage of correct predictions. The classification report explores the precision, recall and f1 score of the model. The precision gave labels of mobile (0) an 85% and fixed (1) 90% which gives a further insight into which labels the model can predict better than others. The recall score for both the labels is strong particularly that of mobile which is at 90% this shows the effectiveness of capturing actual instances. The f1 scores for both labels are 0.87 which shows a balanced tradeoff between precision and recall scores. The model overall has the ability to make accurate predictions and performs well on the dataset despite a slight imbalance of accuracy between labels.

A red and blue squares with white text

Description automatically generated

Confusion matrix showing the predicted results vs actual results instances. (my confusion matrix when ran in an environment with tensorflow did not display all values)

### **Alternate solutions**

Alternative models such as Logistic regression or Random Forrest could be used to compare the accuracy. When Comparing between the Random Forest and Gradient booster I decided to compare by building a random forest model. This came to an 86.45% accuracy showing that it can compete with the “leading technique” in terms of accuracy, however, does use more computational power which Is a key factor when deploying the model to real world operations. To improve the current models using dimension reduction suggested by (Deng, 2021) which will “increase the models efficiency while maintaining accuracy” which could be a solution to the computation cost required.

# **Bibliography**

Barcelos, G., 2022. Understanding Bias in Machine Learning Models. *Understanding Bias in Machine Learning Models*, 15th March.

Changyong FENG, 2014. National library of medicine. *Log-transformation and its implications for data analysis,* p. 1. introduction.

Deng, F., 2021. Performance and efficiency of machine learning algorithms for analyzing rectangular biomedical data. *Performance and efficiency of machine learning algorithms for analyzing rectangular biomedical data,* p. 1.

ECRTD-UK, 2020. European Journal of Statistics and Probability. *European Journal of Statistics and Probability,* p. 4.

Emmanuel, T., 2021. A survey on missing data in machine learning. *A survey on missing data in machine learning,* p. 10.

Gert-Jan-Streefland, 2023. A Gradient Boosting model to predict the milk production. *A Gradient Boosting model to predict the milk production,* 6(100302), pp. 2-3.

Saini, A., 2023. Gradient Boosting Algorithm: A Complete Guide for Beginners. 2nd august .

Tan, C. O., 2006. Predictive models in ecology: Comparison of performances and assessment of applicability. *Predictive models in ecology: Comparison of performances and assessment of applicability,* p. Introduction.