Investigating Sound Levels in Dublin City

Project Assignment for Specialist Certificate in Data Analytics Essentials

Kevin Goslin

1. GitHub URL

https://github.com/kevgos/UCDPA_KevinGoslin.git

Repository includes:

- 'Data' directory: All datasets used in Jupyter notebooks described below are contained within this directory.
- 'jupyter_notebooks' directory: Jupyter notebooks containing the code for the project which will be referenced in this report.
- 'data_collection_scripts' directory: Python scripts of custom function used for API request and the script for merging JSON arrays from the requests.

2. Abstract

This project investigated sound levels in Dublin City at different stages over approximately five years. Noise level data was collected from several different individual monitors in Dublin using requests to the SmartDublin API. After the different monitor data was merged together various analyses were carried out including an investigation in to the effect of the covid-19 lockdown response, holiday periods and different weather conditions on detectable sound levels in the city. Following this it was investigated if a supervised machine learning approach could be used to predict sound levels of an individual monitor. Regression models and a decision tree regressor model were trained using weather features, time of day and traffic volumes and the performance metrics of the models were assessed and compared.

Main findings

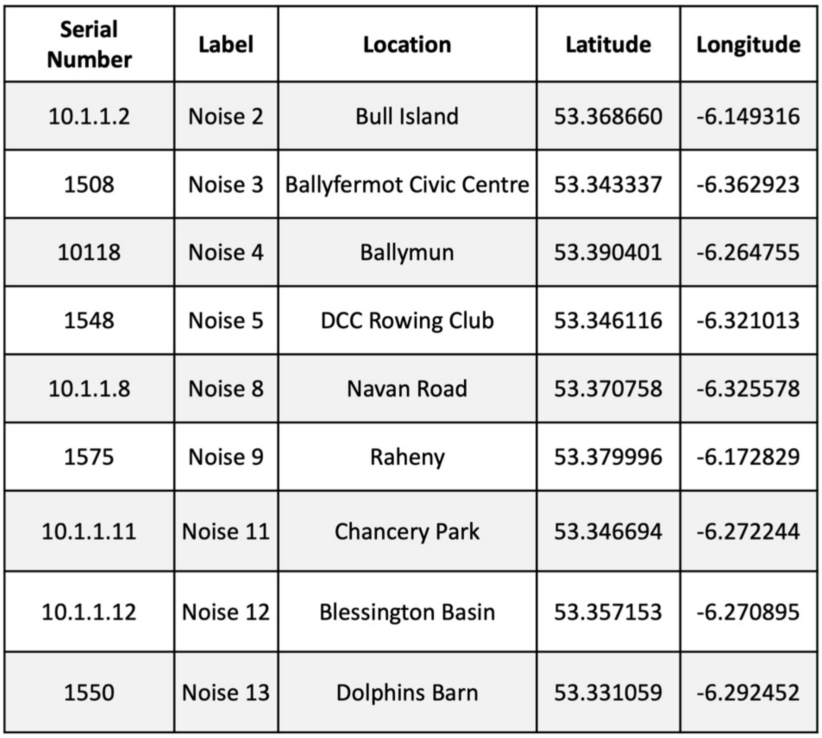
Lockdown: Noise monitor sound differences recorded before and after the covid-19 lockdown response indicate that the lockdown had a detectable impact on recorded noise across the city.

Weather: The impact of weather on noise levels may depend on the geographical location of the monitor. Urban monitors do not seem to be as effected by weather as the monitor on Bull Island.

3. Introduction

Excessive noise can have detrimental effects on human health (World Health Organization, 2018). I was interested in studying sound levels in Dublin City and the effects that human behaviour might have on recorded sound levels around the city. Approximately nine years ago the SmartDublin initiative was founded by the four local Dublin authorities. One tool which is part of this initiative is the SmartDublin application programming interface (API) which allows users to request data from various monitors around Dublin City, including multiple sound recording monitors. I decided to use nine different monitors around the city to generate a sound level dataset from the years 2015 – 2021 (see table 1 for monitors used). The output of these monitors is a weighted measurement of equivalent continuous sound pressure level (LAeq dB(A), hereafter referred to as 'LAeq') over hourly intervals. I was interested in the sensitivity of these monitors and whether human behaviour resulted in a detectable difference in recorded sound and the impact of the weather on sound. For urban planning and noise-reduction measures it is important to be able to assess how loud an area is. In order to assess if supervised machine learning might be an approach for such predictions I generated regression models with one monitor in Ballymun and several features including weather and traffic volume.

Table 1. Noise monitors used for sound level data collection. For API data request the serial number of the monitor and label are used for the request with datetime information.



Serial Number	Label	Location	Latitude	Longitude
10.1.1.2	Noise 2	Bull Island	53.368660	-6.149316
1508	Noise 3	Ballyfermot Civic Centre	53.343337	-6.362923
10118	Noise 4	Ballymun	53.390401	-6.264755
1548	Noise 5	DCC Rowing Club	53.346116	-6.321013
10.1.1.8	Noise 8	Navan Road	53.370758	-6.325578
1575	Noise 9	Raheny	53.379996	-6.172829
10.1.1.11	Noise 11	Chancery Park	53.346694	-6.272244
10.1.1.12	Noise 12	Blessington Basin	53.357153	-6.270895
1550	Noise 13	Dolphins Barn	53.331059	-6.292452

4. Dataset

Table 2. Dataset names and their sources with justification of use. Individual datasets are described further in implementation section below.

Dataset	**Source**	**Justification**
2015to2021dublin_noise_pollution_mergeFile.csv	SmartDublin API: https://data.smartdublin.ie/sonitus-openapi.json	This API provides access to Sonitus noise and air quality monitoring readings for the Dublin area.
2015to2021_weather.csv	Met Eireann: https://www.met.ie/climate/available-data/historical-data	Met éireann provide access to historical hourly weather data. This data is from the Dublin weather station in Phoenix park which is proximal to all noise monitors used in this project
ballymun_traffic.csv	TII traffic data: https://trafficdata.tii.ie/publicmultinodemap.asp	The TII Traffic Data website presents data collected from the TII traffic counters located on the road network. Data was used from the Ballymun traffic counter (ID: TMU R108 000.0 N) which is proximal to the Ballymun noise recorder.

5. Implementation process

5.1 Generating 2015 – 2021 noise dataset

I identified nine noise monitors active in Dublin City (see Table 1) that measure equivalent continuous sound pressure level (LAeq). The SmartDublin API has a two-week duration limit on requests. To gather a longer timespan I wrote a custom Python function that takes two lists of Unix timestamps as arguments, as well as the serial number of a noise monitor, and iterates through the list of dates (see '**smartDublin_request_function.py**' in ‘data_collection’ directory). This function generates multiple JSON arrays containing the noise monitor data for each timespan. I then merged the JSON files and generated a csv file of data for each monitor using another Python script I generated (see ‘**mergingJSONfiles_output_csv.py**’). I then read each csv file in to pandas dataframes and merged the dataframes together (see ‘**mergingCSVfiles.ipynb’**) in to a dataset as **'2015to2021dublin_noise_pollution_mergeFile.csv**' which was saved to the ‘data’ directory.

5.2 Cleaning the 2015 – 2021 noise dataset
notebook: (1)cleaningData.ipynb

In order to clean the data I removed duplicate values and as the data had some chunks of missing values from some individual monitors I dropped rows of missing values (see ‘**cleaningData.ipynb**’. I also added in several columns for month/day/time of the year using the ‘datetime’ column of the dataframe and the pandas.Series.dt accessor object. To annotate whether days of the week were at the weekend or not I used a dictionary object and the map function. This cleaned and annotated DataFrame was then written to a csv file in the ‘dataset’ subdirectory of the ‘data’ directory as ‘**2015_2021noise_pollution_Cleaned.csv**’.

5.3 Exploratory data analysis
notebook: (2)EDA.ipynb

To explore the distribution of the noise monitor data I generated histograms, cumulative distribution function (CDF) plots, and probability density function (PDF) plots using noise monitor data from hourly timepoints compared to a theoretical normal distribution as well as resampled data which was averaged in to daily LAeq averages (see ‘**EDA.ipynb**’). For hourly timepoints a left-skew and high kurtosis could be seen for some of the data, indicating the noise monitor data recorded at this site could be affected by high values. For example the average LAeq recorded by the Ballymun area monitor has a left-skewed peak and also has the highest mean and median LAeq recorded for hourly timepoint data. This is possibly explained by the proximity of Ballymun to Dublin airport, which would be affected by the noise of airplanes during day and night (Basu *et al.*, 2021).

In order to identify the hour of the day in which the different monitors most frequently have their maximum LAeq value recorded for the day I wrote a custom function (peak_LAeq) that finds the modal average of the hour in which this maximum is found.

5.4 Analyses

5.4.1 Effect of lockdown on sound levels

notebook: (3)lockdown_impact.ipynb

To examine the possible effect the first lockdown implemented by the Irish government might have on sound levels I imported my noise pollution dataset using pandas and created a new subset of data made up of noise values in the week leading up to the lockdown (27th March 2020) and the week following the lockdown. To visualize the change in average noise pollution I generated a boxplot figure using matplotlib and the seaborn extension (see figure 1) and combined the reduction of means in to a table (Table 2). This code and figure is stored in the notebook '**lockdown_impact.ipynb**'.

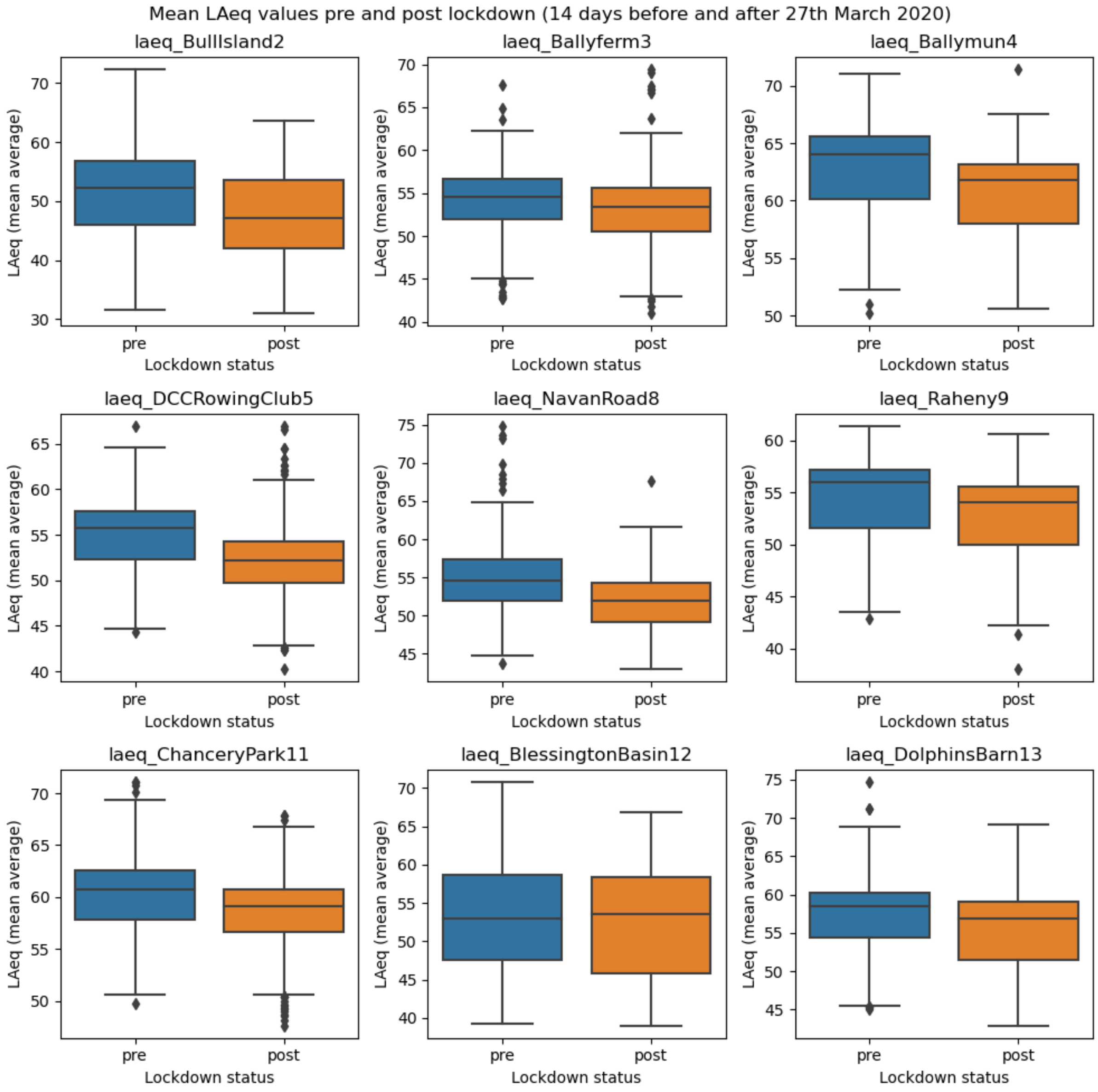
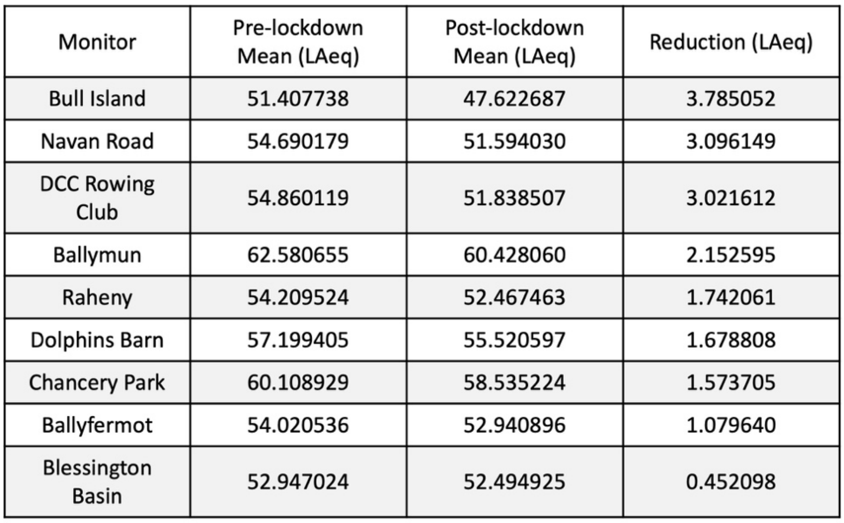


Figure 1. Lockdown impact on sound recording. The mean values recorded by each monitor are collected 14 days before the lockdown (27th March) and for the 14 days after the lockdown.



Monitor	Pre-lockdown Mean (LAeq)	Post-lockdown Mean (LAeq)	Reduction (LAeq)
Bull Island	51.407738	47.622687	3.785052
Navan Road	54.690179	51.594030	3.096149
DCC Rowing Club	54.860119	51.838507	3.021612
Ballymun	62.580655	60.428060	2.152595
Raheny	54.209524	52.467463	1.742061
Dolphins Barn	57.199405	55.520597	1.678808
Chancery Park	60.108929	58.535224	1.573705
Ballyfermot	54.020536	52.940896	1.079640
Blessington Basin	52.947024	52.494925	0.452098

Table 2. Mean LAeq values recorded pre and post lockdown. Numeric values as described in Figure 1 above**.**

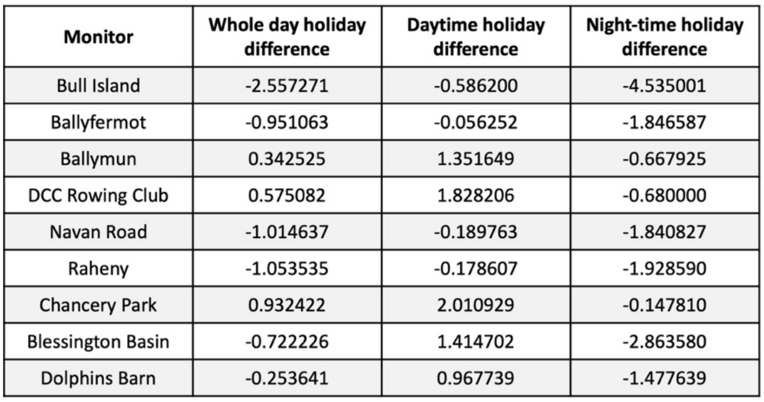
5.4.2 Effect of time of day and holidays on sound levels

notebook: (4)time_holiday_impact.ipynb

A large amount of noise recorded likely comes from human activity during ‘day time’ hours. To investigate this I specified a ‘day’ time during the hours of 6 am and 6 pm and for the other time to be labelled as ‘night’. When the data was grouped in to day and night an observable difference was seen between the two groups mean values.

To investigate whether events such as holiday periods might also have an effect on detectable noise pollution I generated a subset of data made up of Christmas holiday dates. To do this I changed the ‘datetime’ column to type ‘string’ object and then used a regular expression for the dates to filter the dataset. I created a new dataset, ‘df_christmas’ made up of data from the 25th and 26th December and the 1st January. In order to detect if a change in holiday noise might be more pronounced with only daytime values I carried out this analysis using the daytime only values and the night-time only values also. The differences in the mean LAeq between the whole data and the holiday times are described in Table 3 below.

Table 3. Differences in mean sound levels recorded over the Christmas holiday period.



Monitor	Whole day holiday difference	Daytime holiday difference	Night-time holiday difference
Bull Island	-2.557271	-0.586200	-4.535001
Ballyfermot	-0.951063	-0.056252	-1.846587
Ballymun	0.342525	1.351649	-0.667925
DCC Rowing Club	0.575082	1.828206	-0.680000
Navan Road	-1.014637	-0.189763	-1.840827
Raheny	-1.053535	-0.178607	-1.928590
Chancery Park	0.932422	2.010929	-0.147810
Blessington Basin	-0.722226	1.414702	-2.863580
Dolphins Barn	-0.253641	0.967739	-1.477639

5.4.3 Effect of weather on sound levels

notebook: (5)weather_impact.ipynb

To investigate if weather had any impact on noise levels recorded I downloaded hourly weather data for Dublin using Met éireann's historical weather data (see Table 2). I merged the hourly weather data with the hourly sound level data using the 'datetime' column and pandas merge function. I generated a correlation coefficient matrix and plotted this as a heatmap. The correlation values were mostly close to 0 indicating that weather does not have a strong effect on the monitor noise data. Interestingly, the highest correlation coefficient value was found between 'windspeed' recorded and noise values on the Bull Island monitor. The correlation coefficient was weakly positive (r = 0.36) with windspeed possibly increasing noise recorded (see figure 2 below).

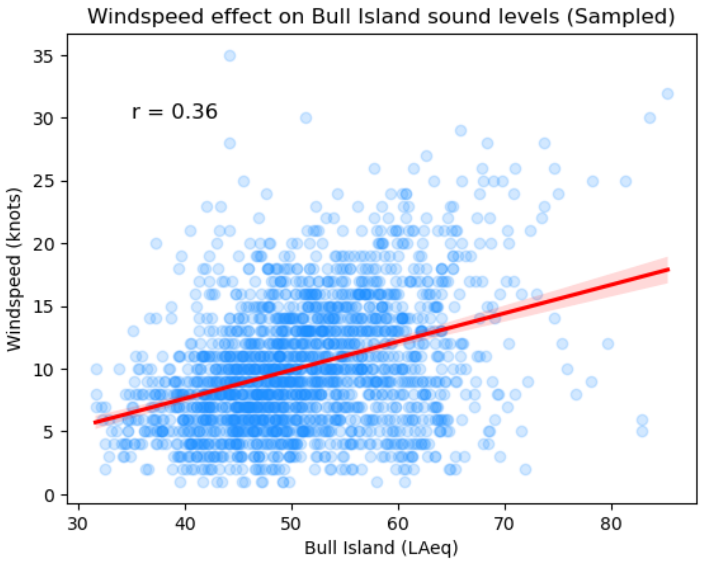


Figure 2. Windspeed effect on Bull Island sound levels. 2000 data points were randomly sampled from a dataset made up of noise monitor data and weather temperature and windspeed was plotted against noise recorded on Bull Island.

From the 27th February to the 4th March 2018 Storm Emma, or 'the beast from the East', resulted in heavy snowfall across Ireland. I was interested in how this storm would have effected noise monitor levels. I calculated the mean of the monitor noise level between the 28th February to the 3rd March 2018 and calculated the difference between this mean and the monitor's data mean for the whole dataset (see figure 3). Interestingly, monitors in urban areas either go down in volume or do not change significantly, whereas the Bull Island monitor has an increase in mean LAeq recorded.

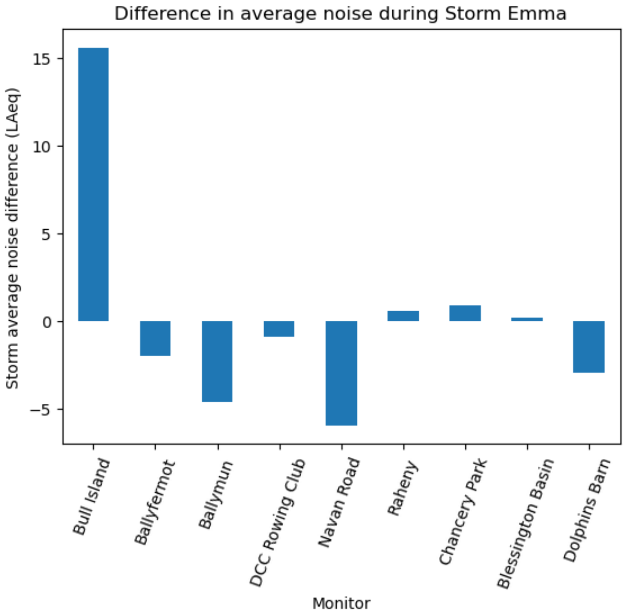


Figure 3. Effect of Storm Emma on noise monitor sound levels.

5.4.4 Supervised machine learning for prediction of sound levels

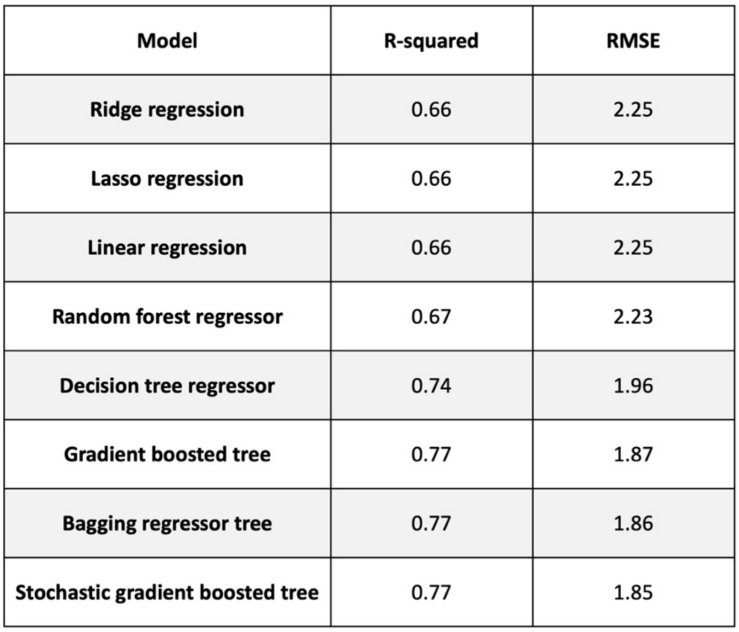
notebook: (6)supervised_learning.ipynb

I was interested in using a supervised machine learning approach to predict the values of an individual noise monitor. The weather data discussed above did not provide any likely variables that would be strong predictors for a model by themselves. Two variables that would likely have an impact on noise are the time of day and the volume of traffic in the area. In order to investigate if these would be useful variables I downloaded traffic volume data from a monitor in Ballymun (see table 2) between January and June in 2019 (‘**ballymun_traffic.csv**’ in ‘Data’ directory) . I then merged this data with noise recording data from the Ballymun monitor by subsetting the Ballymun data in the same time range, and weather data using the ‘datetime’ column in all datasets.

I generated a dataset (‘df_learning’) made up of hourly noise monitor values, time of the day (‘hour’), traffic volume and rainfall data. I then instantiated a number of machine learning models from the sklearn module. I trained a linear regression model and then trained regularized ridge and lasso regression models. I performed hyperparameter tuning on the ridge and lasso models by tuning the alpha hyperparameter of the models. To gauge performance I used the R-squared value of the model’s predictions, which describes how much variation the model can explain, and the root mean squared error (RMSE) which measures the average difference between the model’s predicted values and the training data values. I then used K-fold cross validation to calculate the mean R-squared value and RMSE for the tuned models.

Following this I instantiated a decision tree regressor model. I calculated model performance for a random forest regressor model and then used gradient boosting and stochastic gradient boosting to see if this would improve a decision tree’s performance. I found that a stochastic gradient boosted tree gave the best performance for R-squared and RMSE values (see table 4).

Table 4. Performance metrics for different supervised machine learning approaches used to predict noise values in Ballymun. Data is sorted by RMSE from high to low.



Model	R-squared	RMSE
Ridge regression	0.66	2.25
Lasso regression	0.66	2.25
Linear regression	0.66	2.25
Random forest regressor	0.67	2.23
Decision tree regressor	0.74	1.96
Gradient boosted tree	0.77	1.87
Bagging regressor tree	0.77	1.86
Stochastic gradient boosted tree	0.77	1.85

In order to visualize the difference between the average prediction error of the worst performing model (a ridge regression model) and the best performing model (stochastic gradient boosted decision tree) I used the matplotlib pyplot module to plot the difference between the model's prediction and the training data value (residual error) for all predicted values. In order to visualize a residual error of 0, which would be a perfect prediction, I plotted a line at y=0 across the plot. It can be observed from the plot that the stochastic gradient boosted decision tree model results in residual errors that are closer to 0 than the ridge regression model.

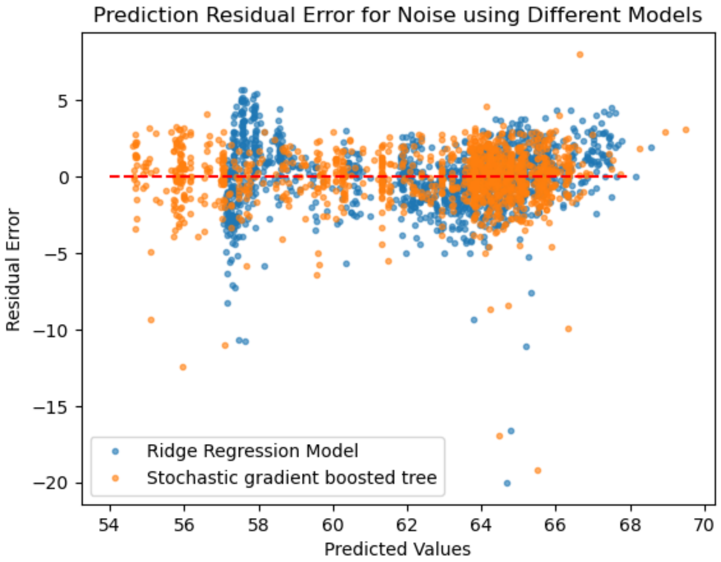


Figure 4. Difference in predicted value residual error in ridge model and boosted decision tree. Residual error was collected by calculating the difference between the predicted value of the model and the testing data used for prediction generation.

Insights from data:

- The lockdown response for covid-19 had an impact on sound levels across Dublin noise monitors with all monitors assessed having a reduction in noise (see section 5.4.1). This is indicative of the effect of human behaviours on noise in the areas.
- The Christmas holiday period had a mixed effect on noise data with no clear pattern. It does not seem likely holiday periods result in a large difference in noise levels across the city.

- Weather does not have a large detectable impact on the sound levels, except for possibly in the case of Bull Island where the 'windspeed' variable has a low positive correlation with sound levels recorded. During a storm Bull Island was also the only monitor to see a large increase in noise. This indicates that urban areas are more protected from having noise increases caused by weather patterns.
- There is a high positive correlation of the traffic volume with noise recorded at Ballymun monitor. This indicates that traffic volume is a significant contributor to noise levels in Ballymun
- A model using traffic volume, rainfall and time of day can be used to predict noise values in an area with reasonable accuracy metrics. It may be useful to consider other variables in the future to achieve better metrics.

References

Basu, B., Murphy, E., Molter, A., Basu, A.S., Sannigrahi, S., Belmonte, M. and Pilla, F. Investigating changes in noise pollution due to the COVID-19 lockdown: The case of Dublin. Ireland, Sustainable Cities and Society, Volume 65, 2021,102597, ISSN 2210-6707. Available at https://doi.org/10.1016/j.scs.2020.102597.

World Health Organization. 2018. Environmental noise guidelines for the European Region. Available at https://www.who.int/publications/i/item/9789289053563

Appendix

All notebooks found in 'jupyter_notebooks' subdirectory of Project folder.

- (1)cleaning_data.ipynb
- (2)EDA.ipynb
- (3)lockdown_impact.ipynb
- (4)time_holiday_impact.ipynb
- (5)weather_impact.ipynb
- (6)supervised_machine_learning.ipynb