**Dataset**

* The NYC Airbnb Data file consists of Airbnb listings and metrics in NYC, USA (2019). It includes all needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.
* The dataset consists of 100 instances and 3 features.
* The features included for computations are neighbourhood group, price and reviews per month.
* Categorical feature: neighbourhood group – Value can be Manhattan or Brooklyn
* Numerical feature: Price and reviews per month.
* Source: <https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data>

**Distance Measures**

* The three distance measures used for calculating distances are Euclidean distance, Manhattan distance and Minkowski distance.
* The distances represented in the code are distance1, distance2 and distance3 for Euclidean, Manhattan and Minkowski respectively.

**Imputation Methods**

* The three imputation methods used to compute the predicted value is 1-NN algorithm, k-NN algorithm and weighted k-NN algorithm.
* The three imputation measures are used for 5%,10%,20% values of all features by considering all three distance measures for each imputation algorithm.

**Imputation accuracy measure**

* The imputation accuracy is calculated by using relative error measure.
* Relative error = (Predicted value – Original Value / Original Value)

**Feature scaling methods**

* The two feature scaling methods used are min-max normalization and Mean normalization.

**Tools**

* The language used for implementation is R-3.6.1 with IDE as R studio.
* No additional libraries are needed to be installed before execution.

**Execution steps**

* AccuracyMeasure data frame determines the relative error for particular feature by considering all three distance measures and imputation methods.
* Feature1 (1) is termed for categorical feature, feature2 (2) and feature3 (3) for two numerical features.
* Parameter featureno determines the feature for which missing values will be created and final values will be computed.
* Load the R file and run the whole file.
* The first run will give the accuracyMeasure for 5% values of feature 3.
* For every execution, run the line determineAccuracyMeasure(featureno,percent,scaleFlag)
  + Featureno – pass the feature number (1,2,3) for which imputation should be done
  + Percent – pass the percentage (0.05,0.1,0,2) of missing values that needs to be generated
  + scaleFlag – TRUE for distance computation on scaled features otherwise FALSE
  + Example: determineAccuracyMeasure(2,0.1, FALSE) will give me the output for 10% missing value computation of 2nd feature i.e. numerical feature.
* Uncomment the line no 300 to write the accuracy measure data in csv file. (write.table(globalErrorMeasure,"Output.csv",append = FALSE))

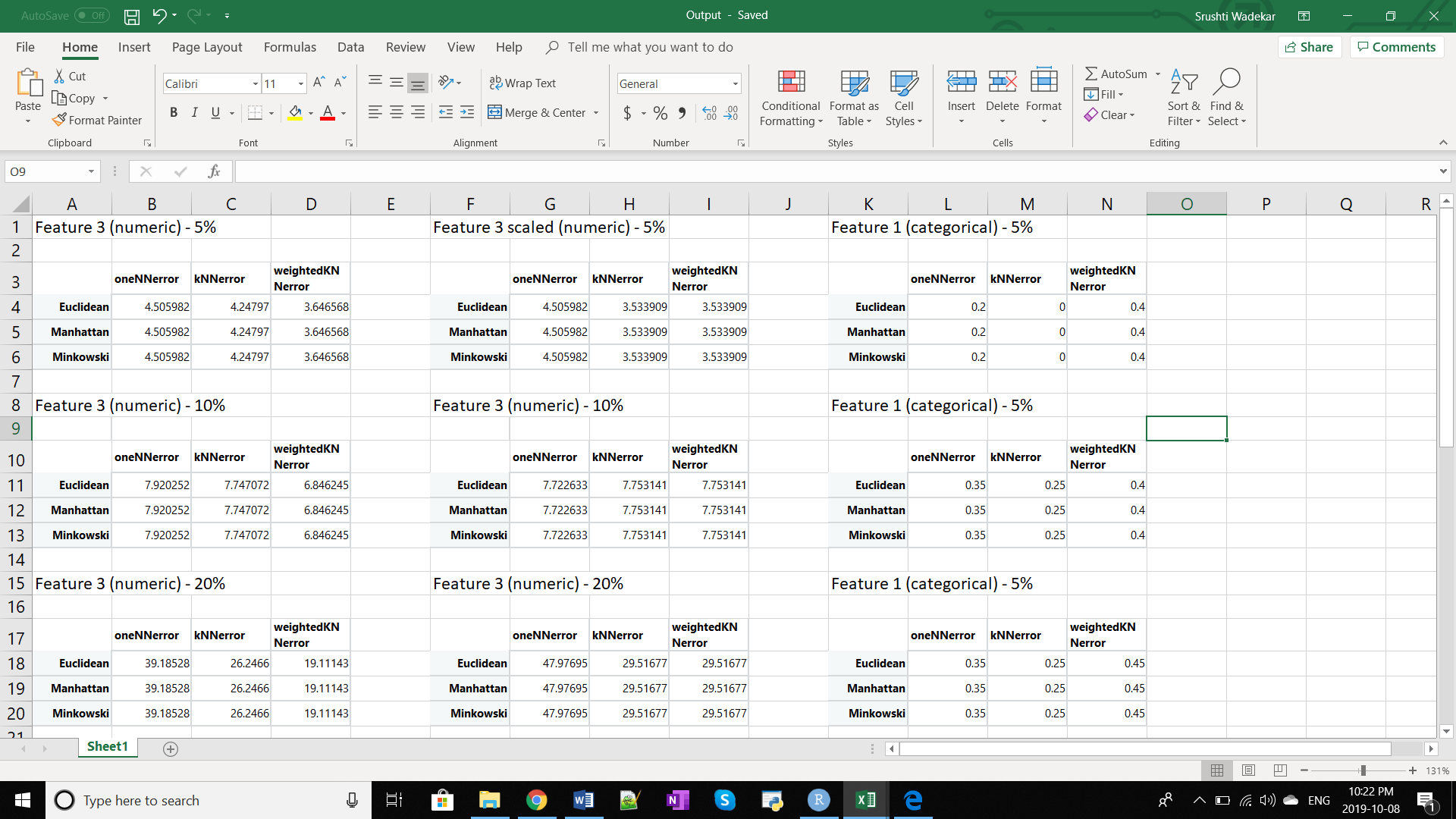
**Results**

* Output based on accuracy measure i.e. relative error is represented below in graphical and tabular format.

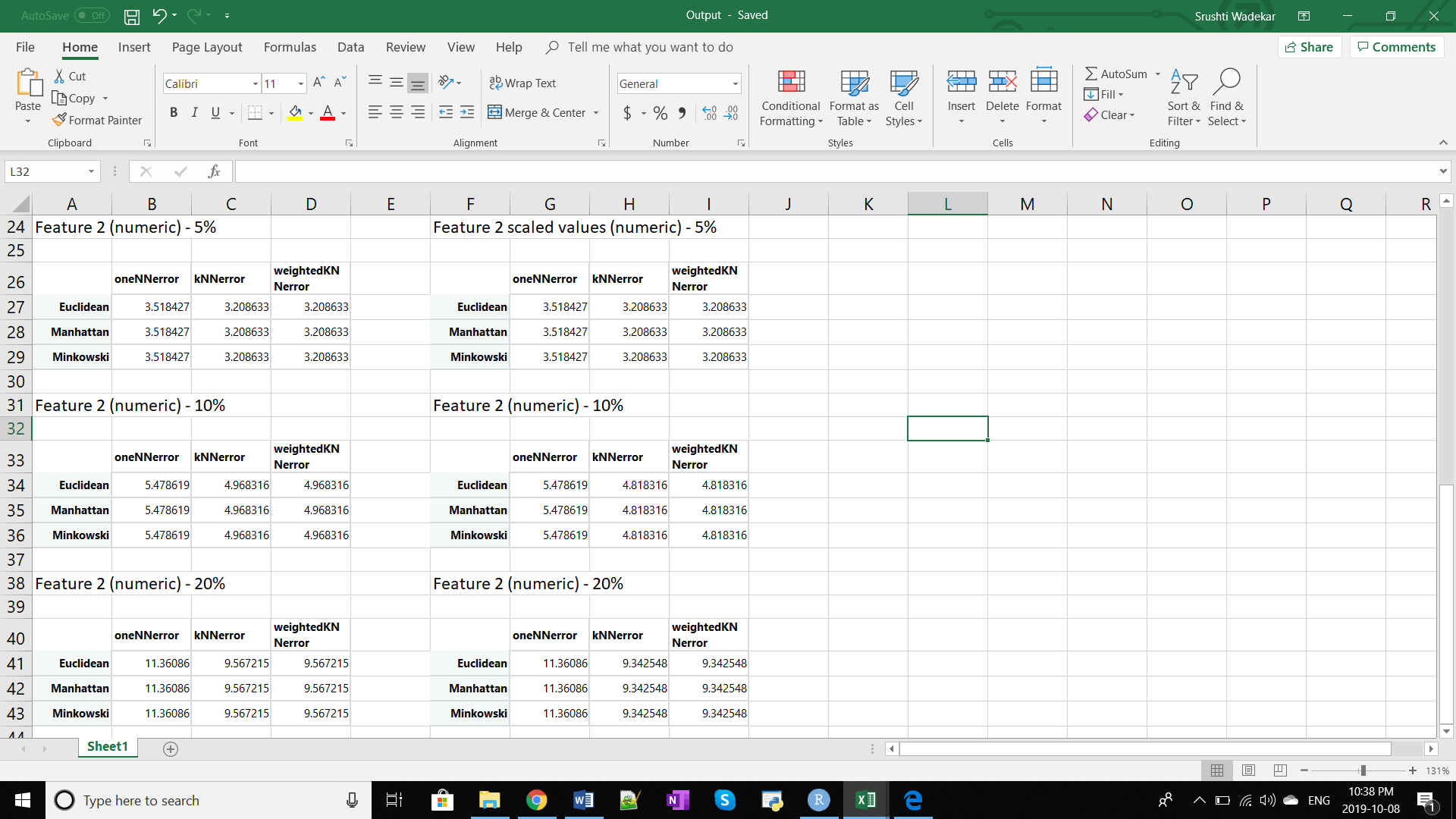
Feature 1 – Categorical

|  |  |  |  |
| --- | --- | --- | --- |
| Feature 1 (categorical) - 5% | | |  |
|  |  |  |  |
|  | **oneNNerror** | **kNNerror** | **weightedKNNerror** |
| **Euclidean** | 0.2 | 0 | 0.4 |
| **Manhattan** | 0.2 | 0 | 0.4 |
| **Minkowski** | 0.2 | 0 | 0.4 |
|  |  |  |  |
| Feature 1 (categorical) - 5% | | |  |
|  |  |  |  |
|  | **oneNNerror** | **kNNerror** | **weightedKNNerror** |
| **Euclidean** | 0.35 | 0.25 | 0.4 |
| **Manhattan** | 0.35 | 0.25 | 0.4 |
| **Minkowski** | 0.35 | 0.25 | 0.4 |
|  |  |  |  |
| Feature 1 (categorical) - 5% | | |  |
|  |  |  |  |
|  | **oneNNerror** | **kNNerror** | **weightedKNNerror** |
| **Euclidean** | 0.35 | 0.25 | 0.45 |
| **Manhattan** | 0.35 | 0.25 | 0.45 |
| **Minkowski** | 0.35 | 0.25 | 0.45 |

Feature 3 - Numerical



**Feature 2 – Numerical**



**Conclusion**

* For the categorical feature, kNN algorithm showed the best accuracy with 0% error for 5% values.
* For the numerical feature 2, kNN and weighted kNN algorithm performed in similar way and better than 1NN algorithm. After scaling feature 2, the accuracy of kNN and weighted kNN increased. Whereas, there was no change observed in 1NN after scaling.
* For the numerical feature 3, weighted kNN performed the best followed by kNN and 1NN.
* After scaling feature 3, kNN and weighted kNN computed same accuracy.
* Thus, for the Airbnb dataset the kNN algorithm computed the best results.