Approach of Solving the Problem of Demand Forecasting:

1. First imported modules
2. Imported dataframes.
3. There are no null values, so no need to worry about that.
4. Train data has 2 independent features and one target feature.
5. After so much trial of the target data I calculated a ‘z’ which is like;
   1. Z = (count of target feature – mean of target feature) / standard deviation of target.
   2. And I subtracted target feature with z
6. Then time for preprocessing of date column. I converted the date column to datetime format, then I created month, day, day\_of\_week, is\_weekend and then I dropped the main date column.
7. From hour column I created a column whether it is Early morning, Morning, Afternoon, Evening, Night or Late Night.
8. Then from month column I created whether it was Spring, Summer, Autumn or Winter.
9. In type\_of\_day and Season column there are categorical variables, So I used Ordinal Encoder to replace those categorical values to numerical.
10. Then I divided the train data into X & y training & splitting purpose.
11. Hour, month & day\_of\_week data are cyclical data. So I used cosine transformation on them & kept the rest as it is.
12. In statistics, the variance inflation factor(VIF) is the ratio of the variance of estimating some parameter in a model that includes multiple other terms by the variance of a model constructed using by only one term.
    1. So I used VIF method to calculate the collinearity of the columns and found ‘year’ & ‘is\_quarter\_date’ are the ones with high collinearity.
    2. So, I removed them.
13. Now we are ready for splitting the data using train\_test\_split and ready for training.
14. I used Ridge, MLP regressor from neural network, random forest regressor, catboost regressor, xg boost regressor, lasso, linear regression, Light gradient boosting regressor.
15. Amongst these entire models only light gradient boosting regressor & catboost regressor are the ones with good results.
16. I tried using hyperparameters through gridsearch cv but base model was working better.
17. Then same preprocessing which was done for train data frame has to be done with test dataframe, then predict using the model which gave the better score better i.e Light Gradient Boosting Machine Regressor which is a fast, distributed, high – performance gradient boosting framework based on decision tree algorithm.
18. After predicting the test set I saved that on a data frame and added with ‘z’ which I subtracted before and replaced with sample submission ‘demand’ column and that’s it.