1. **A brief on the approach, which I have used to solve the problem.**

Raw Data

Data cleaning : Created a new feature(weekend)

New Feature: weekend

Dummy variable creation for categorical variables

Hyperparameter Tuning

Tuning of the following parameters:

n\_estimators, max\_depth, max\_features, min\_sample\_leaf, min\_sample\_split

Random search of parameters:

Random search of above parameters to get the optimal parameters.

Final model:

Output is obtained as the aggregation of several decision trees in the random forest model using the optimal hyperparameters.

I have proceeded with an ensembling technique wherein I have used the random forest algorithm to get the sales of two months. Decision trees are used as base learners to get the final forecast by boostrapped aggregation method.

1. **Data-preprocessing /feature engineering ideas.**

I created a new feature called ‘**weekend’** which was extracted from the **Date** column. Almost all the features were categorical variables which were handled using dummy variables. The “#order” column was not used in the training purpose as we do not have that in the test set. The features that were were encoded with dummy variables were: "Region\_Code", "Store\_id", "Store\_Type", "Location\_Type", "Region\_Code", "Holiday" and "Discount"

After introducing dummy variables each of the categorical variables about generated n-1 features which were then concatenated to the original datasets after removing the original columns in both the train and test sets. So the total number of features went from 9 to 384 in train data and from 7 to 382 in test data after introduction of dummy variables.

Hyper parameter tuning of the following parameters were carried out:

max\_depth , max\_features, min\_samples\_leaf , min\_samples\_split and n\_estimators . After tuning each parameters a radomisedsearch was conducted and the optimum parameters were used to fit the model and predictions were made on the test data.

3. **Final model:**

The final model is a random forest model with the following hyperparameters.

**'n\_estimators': 26,**

**'min\_samples\_split': 100,**

**'min\_samples\_leaf': 100,**

**'max\_features': 70,**

**'max\_depth': 35**

It was observed that the MSE and RMSE of the model after using the optimal parameters increased.

MSE before parameter tuning was: 108922565.98

MSE after parameter tuning was: 130977458.89

RMSE before parameter tuning was: 10436.60

RMSE after parameter tuning was: 11444.53

Model score on dashboard before parameter tuning: 207.496440604687

Model score on dashboard after parameter tuning: 228.618009249067

4**. Scope for improvement of model performance:**

* + 1. Additional features like weekdays and months can be created.
    2. Boosting techniques like Catboost and Xgboost can be ensemble along with Bagging techniques like the model we have used i.e. Random Forest, which might give a better performance.