PREDICTION OF HOURLY LOAD FOR NEW HAMSPHIRE

(Team 1 : Kedharnath Reddy Gali, Vivek Reddy Karra, Sai Dinesh Kondragunta)

1. Business Purpose and Background

This project's main goal is to create a reliable predictive model that can predict New Hampshire's hourly load with accuracy in January 2023. We hope to give our traders accurate insights into the energy market at a pivotal time of the year by concentrating on this particular time frame. We can record demand trends, seasonal variations, and any other special factors that may impact the New Hampshire power market in that month thanks.

Precise hourly load estimates are essential in the world of electricity trading. Since the energy market is real-time, any disparity in load projections can result in significant losses. Our hedge fund hopes to obtain a competitive edge by using sophisticated predictive models to make well-informed trading decisions. Our traders can maximize profits, adjust quickly to market fluctuations, and optimize their trading strategies thanks to our accurate and timely forecasts.

Precise hourly load forecasts are important because they can serve as a foundation for decisions in the power trading sector. They enable traders to forecast changes in demand, strategically buy and sell electricity contracts, and modify energy generation to meet fluctuating demand. Accurate load forecasts make it easier to identify profitable trading opportunities, reduce the risks associated with price swings, and improve overall portfolio management.

Reliable load estimates are necessary for efficient risk management. Energy trading carries high financial risk, and poor load forecasting can result in sizable losses. Trading professionals can reduce the risks brought on by price volatility by possessing trustworthy forecasts. Furthermore, traders can evaluate possible supply-demand imbalances and related risks by having a thorough understanding of load patterns.

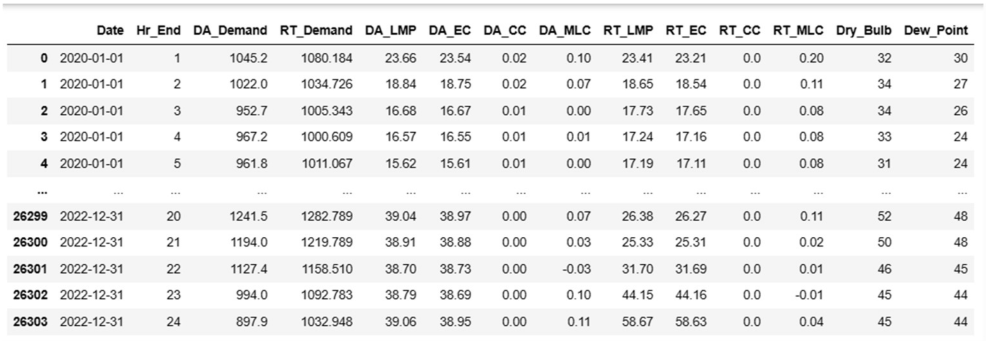
The application of an accurate forecasting model has the capacity to revolutionize trading practices. Since the traders have access to precise hourly load forecasts, they can make data-driven decisions and optimize the timing and size of trades. This accuracy lowers the possibility of losses while increasing income. It enables the hedge fund to react to market shifts, seize fresh opportunities, and cope with unanticipated circumstances. The project's success eventually translates into improved financial outcomes, elevating the standing and profitability of our hedge fund.

2. Data Exploration

The ISO-New England provided the datasets. To carry out our analysis, we make use of New Hamsphire data from 2020 to 2022. Three datasets comprise our collection:

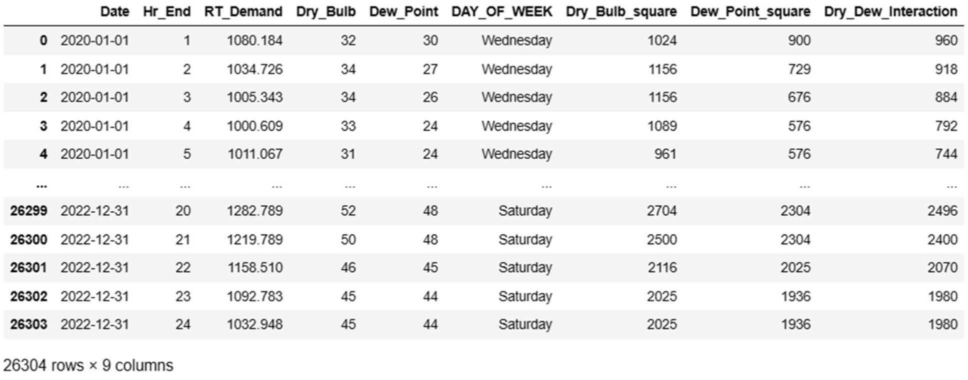
- 2020 SMD Hourly Data
- 2021 SMD Hourly Data
- 2022 SMD Hourly Data

Once the data is loaded, we first merge three datasets into a single dataframe. Once they are all in a single dataframe, we handle missing values, remove duplicates, and restructure the data using preprocessing and data cleaning.



	Date	Hr_End	DA_Demand	RT_Demand	DA_LMP	DA_EC	DA_CC	DA_MLC	RT_LMP	RT_EC	RT_CC	RT_MLC	Dry_Bulb	Dew_Point
0	2020-01-01	1	1045.2	1080.184	23.66	23.54	0.02	0.10	23.41	23.21	0.0	0.20	32	30
1	2020-01-01	2	1022.0	1034.726	18.84	18.75	0.02	0.07	18.65	18.54	0.0	0.11	34	27
2	2020-01-01	3	952.7	1005.343	16.68	16.67	0.01	0.00	17.73	17.65	0.0	0.08	34	26
3	2020-01-01	4	967.2	1000.609	16.57	16.55	0.01	0.01	17.24	17.16	0.0	0.08	33	24
4	2020-01-01	5	961.8	1011.067	15.62	15.61	0.01	0.00	17.19	17.11	0.0	0.08	31	24
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
26299	2022-12-31	20	1241.5	1282.789	39.04	38.97	0.00	0.07	26.38	26.27	0.0	0.11	52	48
26300	2022-12-31	21	1194.0	1219.789	38.91	38.88	0.00	0.03	25.33	25.31	0.0	0.02	50	48
26301	2022-12-31	22	1127.4	1158.510	38.70	38.73	0.00	-0.03	31.70	31.69	0.0	0.01	46	45
26302	2022-12-31	23	994.0	1092.783	38.79	38.69	0.00	0.10	44.15	44.16	0.0	-0.01	45	44
26303	2022-12-31	24	897.9	1032.948	39.06	38.95	0.00	0.11	58.67	58.63	0.0	0.04	45	44

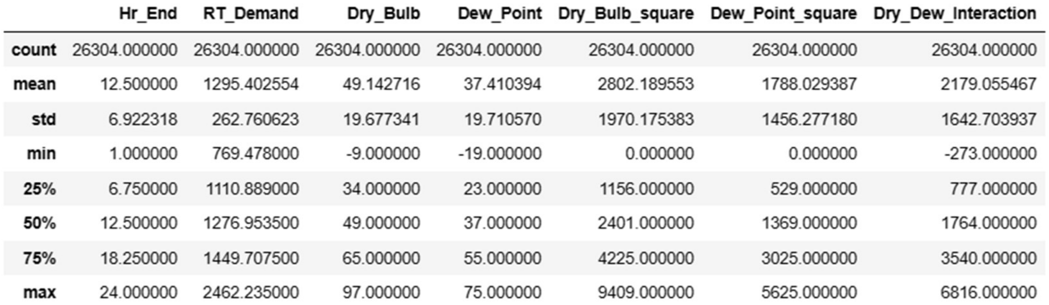
Some unnecessary variables are dropped, and other features are added to the column. Day of the week from Date, dry bulb square, dew point square, and dry bulb and dew point interaction are some of these features.



	Date	Hr_End	RT_Demand	Dry_Bulb	Dew_Point	DAY_OF_WEEK	Dry_Bulb_square	Dew_Point_square	Dry_Dew_Interaction
0	2020-01-01	1	1080.184	32	30	Wednesday	1024	900	960
1	2020-01-01	2	1034.726	34	27	Wednesday	1156	729	918
2	2020-01-01	3	1005.343	34	26	Wednesday	1156	676	884
3	2020-01-01	4	1000.609	33	24	Wednesday	1089	576	792
4	2020-01-01	5	1011.067	31	24	Wednesday	961	576	744
...	...	...	...	...	...	...	...	...	...
26299	2022-12-31	20	1282.789	52	48	Saturday	2704	2304	2496
26300	2022-12-31	21	1219.789	50	48	Saturday	2500	2304	2400
26301	2022-12-31	22	1158.510	46	45	Saturday	2116	2025	2070
26302	2022-12-31	23	1092.783	45	44	Saturday	2025	1936	1980
26303	2022-12-31	24	1032.948	45	44	Saturday	2025	1936	1980

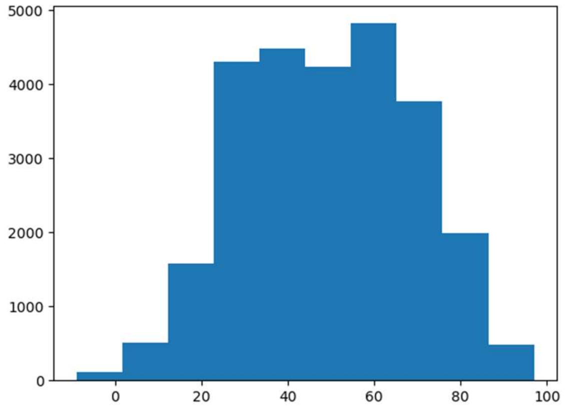
26304 rows × 9 columns

To measure the central tendencies and variability in the data, summary statistics such as mean, median, standard deviation, and upper and lower quantiles, minimum and maximum values are calculated. Correlation matrices are generated in order to investigate the relationships between load and other potential factors.

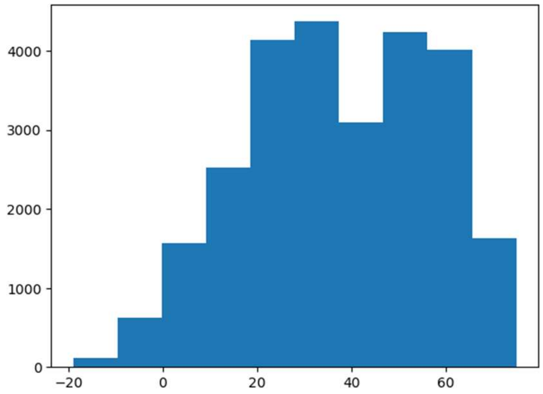


	Hr_End	RT_Demand	Dry_Bulb	Dew_Point	Dry_Bulb_square	Dew_Point_square	Dry_Dew_Interaction
count	26304.000000	26304.000000	26304.000000	26304.000000	26304.000000	26304.000000	26304.000000
mean	12.500000	1295.402554	49.142716	37.410394	2802.189553	1788.029387	2179.055467
std	6.922318	262.760623	19.677341	19.710570	1970.175383	1456.277180	1642.703937
min	1.000000	769.478000	-9.000000	-19.000000	0.000000	0.000000	-273.000000
25%	6.750000	1110.889000	34.000000	23.000000	1156.000000	529.000000	777.000000
50%	12.500000	1276.953500	49.000000	37.000000	2401.000000	1369.000000	1764.000000
75%	18.250000	1449.707500	65.000000	55.000000	4225.000000	3025.000000	3540.000000
max	24.000000	2462.235000	97.000000	75.000000	9409.000000	5625.000000	6816.000000

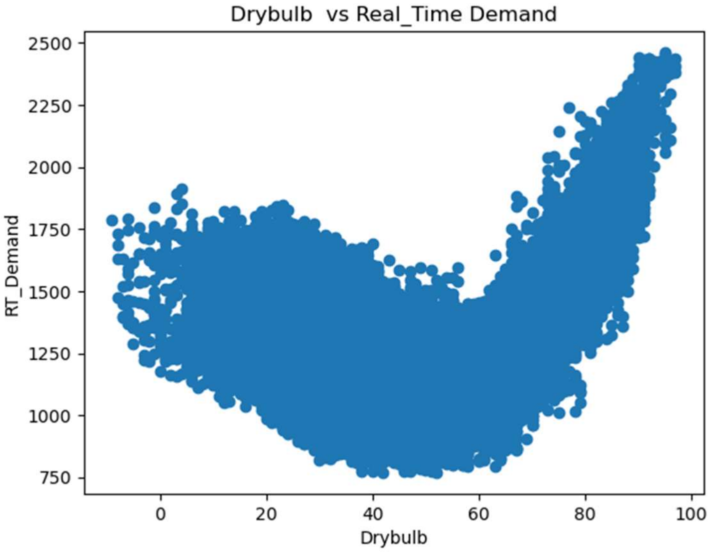
Later, in order to comprehend hourly data more thoroughly, we visualize the data using tools like scatter plots, histograms, and boxplots. Furthermore, we include dummy variables for Hr_End and DAY_OF_WEEK in the data frame to facilitate the model development processing.



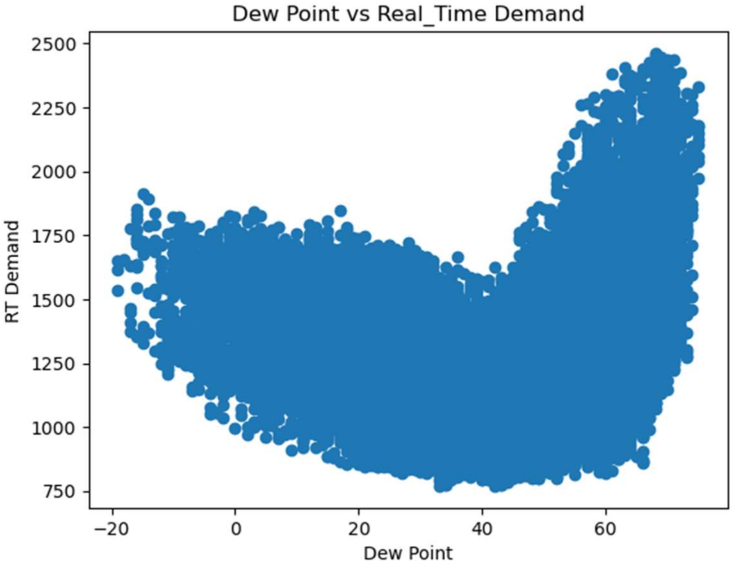
Histogram plot of Dry Bulb



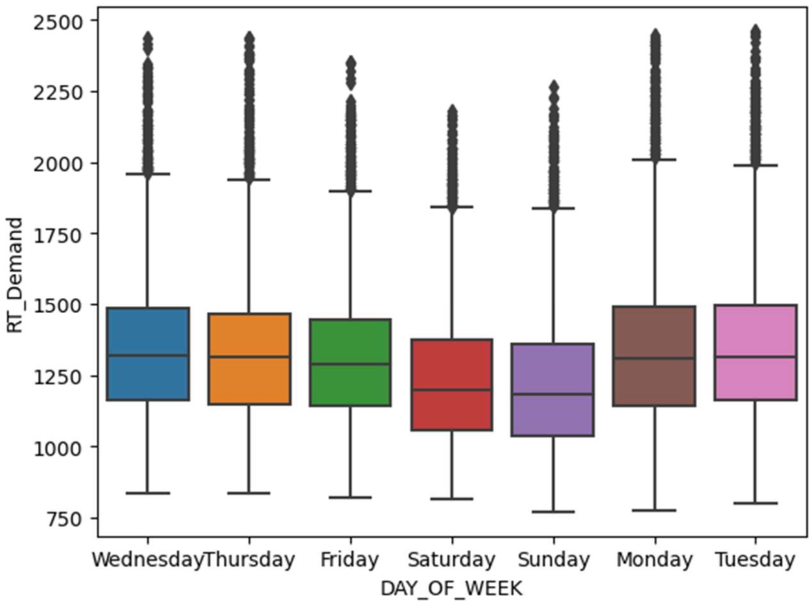
Histogram plot of Dew Point



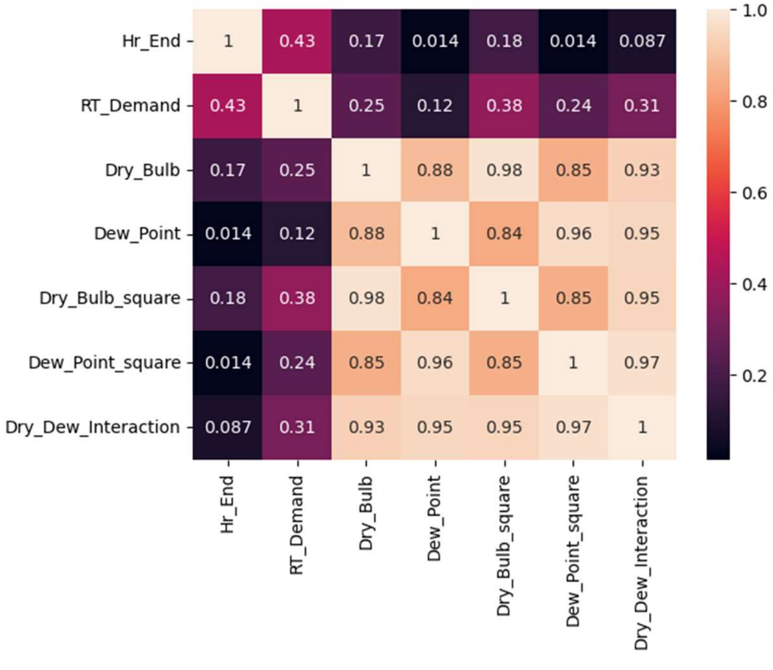
Scatter plot between RT_Demand and Drybulb



Scatter plot between RT_Demand and DewPoint



Boxplot graph of DAY_of_WEEK and RT_Demand



Heatmap of correlation matrix

3. Model Development

We use RT_Demand as the response variable (y) for model development. Variables such as Date, Dry Bulb, Dew Point, Squared Dry Bulb, Squared Dew Point, Interaction of Dry Bulb and Dew Point, and Dummy values of DAY_OF_WEEK and Hr_End are considered for input features (x).

This data is split into testing and training sets. Using the train_test_split function, we split the training data into 75% and the test data into 25% for training and testing. To determine which machine learning algorithm performs best, this split data is used to evaluate the model using a variety of algorithms. The K-Nearbors, Decision Tree, Random Forest, and Linear algorithms are the ones that are used. The sklearn library is used to import these algorithms.

In the same way, we further split the data into sets for cross-validation—training and validation. A more accurate evaluation of a model's performance can be obtained through cross-validation, particularly in situations involving sparse data or time-varying patterns. It helps spot possible problems like overfitting and offers more accurate predictions for how well the model will generalize to fresh, untested data. We use training data from January 2020 to November 2022 in this, and data from the previous month, December 2022, is used for validation. In a similar vein, different ML algorithms are employed to evaluate the model using this split data.

Linear Regression

Linear Regression is a statistical technique used to model the relationship between a response variable and one or more independent variables. It is a simple and widely used method for predicting a quantitative response based on one or more predictor variables. The basic idea behind linear regression is to fit a straight line to a set of data points such that the line minimizes the distance between predicted values and actual values.

Linear Regression can be used for both simple and multiple linear regression. Simple linear regression involves a single independent variable while multiple linear regression involves multiple independent variables.

Linear Regression has many application in the field of finance, engineering, economics, health and many more.

KNeighbors Regressor

KNeighbors Regressor is a machine learning algorithm that can be used for regression tasks. It is a type of instance-based learning, where the algorithm does not create a model during training but instead stores the training dataset and uses it during prediction.

The KNeighbors Regressor works by finding the k-nearest neighbor in the training dataset to a new input data point, where ‘K’ is a hyperparameter that is set by the user. The algorithm then predicts the output value of the new data point by taking the average of the output values of its k-nearest neighbors.

KNeighbors Regressor is a non-parametric algorithm which means it does not make any assumptions about the underlying distribution of the data. It can handle both linear and non-linear relationships between the input and output variables making it a versatile algorithm for regression tasks.

One of the advantages of this regressor is that it can handle noisy data and outliers since it is based on the local structure of the data. It is also relatively easy to uunderstand and implement, making it a good choice for simple regression problems. However, one of the main disadvantages of is that it can be sensitive to the choice of ‘K’, which can effect the performance of algorithms. It can also be computationally expensive for large datasets, since it requires computing the distances between the new data point and all data training points.

Decision Tree Regressor

Decision Tree Regressor is a machine learning algorithm that can be used for regression tasks. It is a type of supervised learning algorithm that builds a decision tree from the training data to predict the output value of the new input data point.

The Decision Tree Regressor works by recursively splitting the training data into subsets based on the values of input features. The algorithm selects the best feature to split at each node of the tree based on the criteria such as information gain. The process continues until the tree reaches a stopping criterion such as maximum depth or a minimum number of samples per leaf node.

Decision tree is simple and interpretable algorithm that can handle both linear and non-linear relationships between the input and output variables. It can also handle noisy data and outliers since it is based on the local structure of the data. One of the advantages of this regressor is that it can handle both categorical and continuous input features.

Random Forest Regressor

Random Forest Regressor is a machine learning algorithm used for regression problems. It is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the average prediction of the individual trees.

The basic idea behind Random Forest is to create a large number of decision trees, each trained on a randomly selected subset of the training data and a random subset of input features. Each decision tree in random forest independently predicts the target variable and finally predicts the target variable and the final prediction is obtained by averaging the predictions of all trees in the forest. This helps to reduce the overfitting and improve the generalization performance of the model.

The random forest algorithm is widely used in machine learning for regression tasks, especially when dealing with high-dimensional datasets.
Some of its advantages include:

- The ability to handle a large number of input features and large number of training samples.
- The ability to estimate the importance of each input feature in predicting the target variable.

4. Model Evaluation

We use the Mean Absolute Error (MAE) and Mean Squared Error (MSE) metrics to assess the accuracy of the machine learning models we use. The differences between the actual load values and the model's predictions are measured by these metrics. Better accuracy and a closer match between predictions and actual observations are indicated by lower MAE and MSE values. In the same way, different ML algorithms are used to evaluate the model using this split data.

The average absolute difference between the values that were predicted and those that were observed is called the mean absolute error, or MAE. Because MAE displays the average prediction error in the same unit as the target variable, it is simple to interpret.

The average squared discrepancy between the expected and actual values is measured by the Mean Squared Error, or MSE. Greater errors are penalized more severely by MSE than by MAE. MSE is helpful for comprehending the model's performance regarding outliers since it sheds light on the magnitude of errors.

Low values of MAE and MSE among the models indicate better model accuracy in hourly load forecasting. These metrics provide a quantitative evaluation of the model's effectiveness in comparison to actual load data. By comparing these metrics across the models for the historical performance, the traders can ascertain the forecasts' accuracy. The quality and precision of the input data greatly influence the forecasts' accuracy. Historical load that is erroneous or incomplete can skew predictions.

5. Result and Conclusion

The results obtained by the test models after evaluating with metrics are tabulated below:

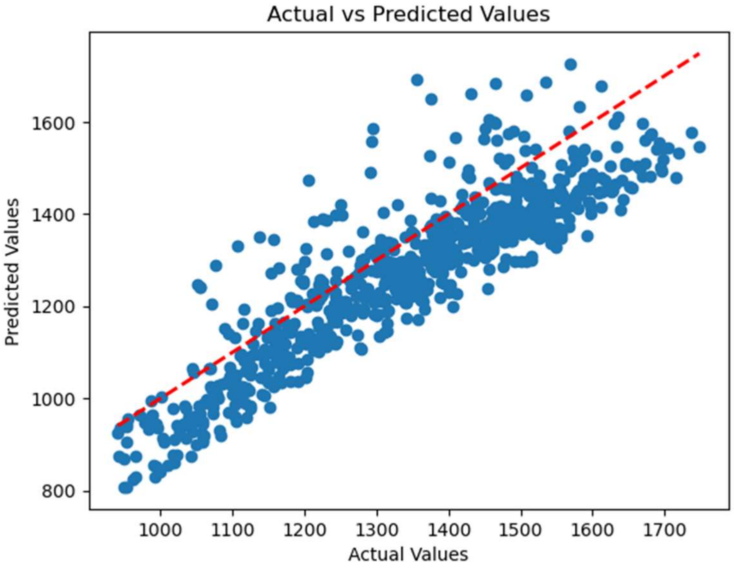
Model	MAE	MSE
Linear Regressor	84.731	11958.9
KNeighbors Regressor	136.665	29374.3
Decision Tree Regressor	78.787	11526.5
Random Forest Regressor	60.775	6662.18

By evaluating performance metrics and contrasting the model predictions with the actual data, it was possible to assess the models' dependability and accuracy. It is crucial to acknowledge the constraints related to data limitations and uncertainties arising from external factors. Validation and regular model updates can address these problems.

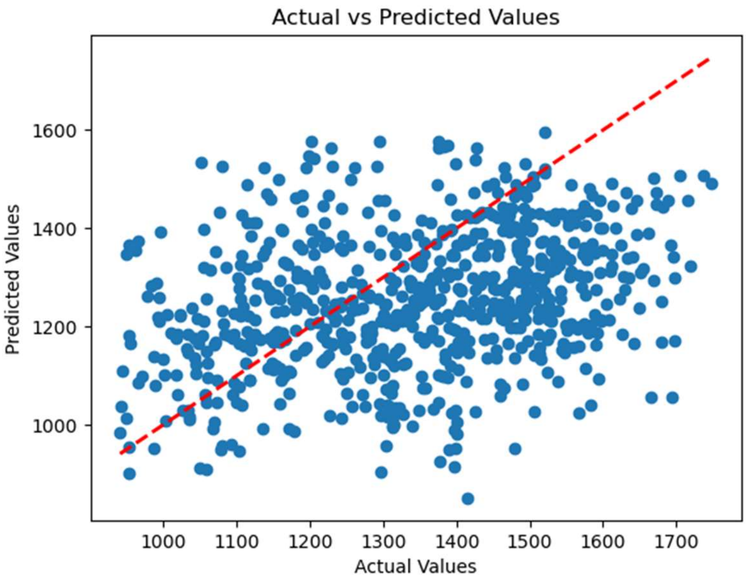
The results obtained by the validation model after evaluating with the metrics are tabulated below :

Model	MAE	MSE
Linear Regressor	91.96	11723.3
KNeighbors Regressor	174.243	45372.8
Decision Tree Regressor	76.767	10175.4
Random Forest Regressor	63.21	6244.52

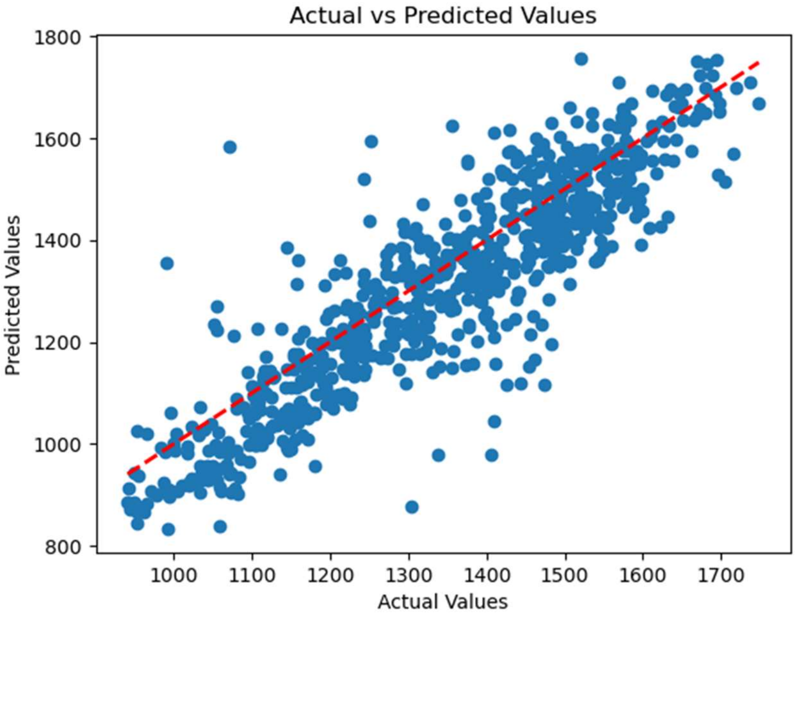
Regression plot of Linear Regression



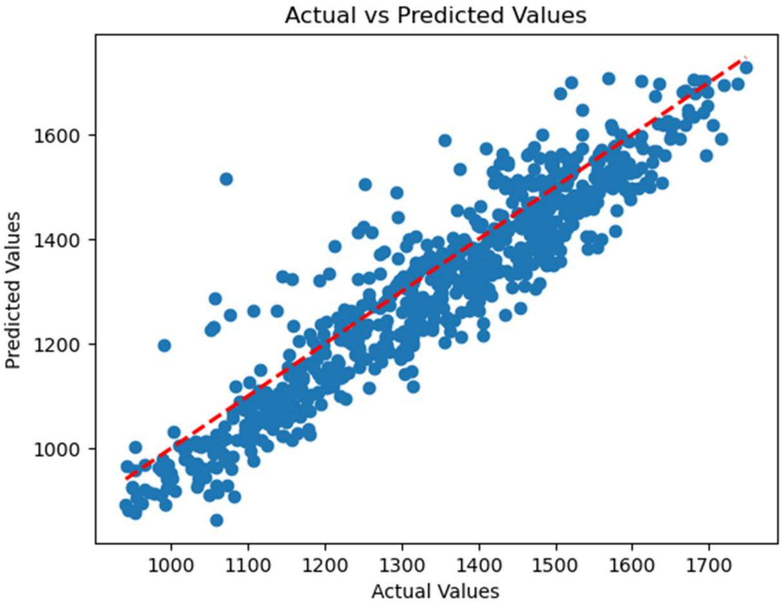
Regression plot of KNeighbors Regressor



Regression plot of Decision Tree Regressor



Regression plot of Random Forest Regressor



Conclusion

By comparing both MAE and MSE values among the models it is concluded that Random Forest Regressor is considered as the best algorithm for predicting hourly load prices.