CAPSTONE PROJECT

MACHINE LEARNING ENGINEER NANODEGREE

APPLIANCE ENERGY PREDICTION

Project Overview :

The purpose of this project is to predict the consumption of the energy by the appliances and also the need for the management and saving of the energy for future endeavours which is possible only through estimation of energy needed for every appliances and ways to optimise it. This is a case of Regression analysis which is part of Supervised Learning problem. Here the target is the appliances and while sensors and weather data are the features

Datasetsource: http://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

Problem Statement :

Here we need to develop a regression model using supervised learning with sensors and weathers data as features and appliances as target

Metrics:

Since we are using the regression algorithm, the metric used will be as *R2* (R squared) also defined as coefficient of determination which gives a measure of the variance of target variable that can be explained using the given features. It can be mathematically defined as:

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

where, SSres = Residual sum of squares , SStot = Total sum of squares

For this project, we will be using *metrics* module of scikit-learn library called as '*r2_score*()' function. the *RMSE (Root Mean Squared Error)* gives the measure of how well model fits the data i.e. how close are the predicted values to the actual values.

Formula: $\textbf{root of } 1/N*\Sigma(yi - yi')^2$

where, N = number of observations ,

yi = Actual value of target variable

yi′ = Predicted value of target variable

In this project, RMSE is calculated by using square root of *mean_squared_error()* function provided in the *metrics* module of scikit-learn library. These two metrics are helpful for this problem because of the following reasons: a)

a) It is a Regression based problem.
b) R2 score will show the statistical robustness of the model.
c) RMSE will info about how accurate the predictions are to actual values.

Analysis :

a) Data Exploration:

NAME	DESCRIPTION	UNIT
T1	Kitchen Temperature	℃
T2	Living Room Temperature	℃
T3	Laundry Room Temperature	℃
T4	Office Temperature	℃
T5	Bathroom Temperature	℃
T6	Temperature outside Building (North)	℃
T7	Ironing Room Temperature	℃
T8	Teenager Room Temperature	℃
T9	Parents Room Temperature	℃
T_out	Outside Temperature (Weather Station)	℃
T_dewpoint	Dewpoint Temperature (Weather Station)	℃
RH_1	Kitchen Humidity	%
RH_2	Living Room Humidity	%
RH_3	Laundry Room Humidity	%

RH_4	Office Humidity	%
RH_5	Bathroom Humidity	%
RH_6	Humidity outside Building (North)	%
RH_7	Ironing Room Humidity	%
RH_8	Teenager Room Humidity	%
RH_9	Parents Room Humidity	%
RH_out	Outside Humidity (Weather Station)	%
Pressure	Outside Pressure (Weather Station)	mm Hg
Wind speed	Outside Windspeed (Weather Station)	m/s
Visibility	Visibility (Weather Station)	km
Date	Timestamp of the reading	yyyy-mm-dd HH:MM:SS
rv1	Random Variable 1	-
rv2	Random Variable2	-
Lights	Energy used by lights	Wh

Target Variable

Appliances	Total energy used by Appliances	Wh

Out of all thefeatures the features relating to time are dropped and so taking all into consideration total number of features would be 24 and target variable would be 1

a) Number of instances in training data = 14,801
b) Number of instances in testing data = 4,934
c) Total number of instances = 19,735
d) Count of Null values = 0
e) All features have numerical values

b) Descriptive statistics:

Ranges of the columns:

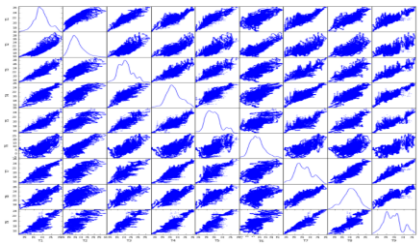
	T1	T2	T3	T4	T5	T6	T7	T8	T9
count	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000
mean	21.691343	20.344518	22.278802	20.860393	19.604773	7.923216	20.273236	22.028122	19.493479
std	1.615790	2.202481	2.012934	2.048076	1.849641	6.117495	2.118416	1.960985	2.022560
min	16.790000	16.100000	17.200000	15.100000	15.340000	-6.065000	15.390000	16.306667	14.890000
25%	20.760000	18.790000	20.790000	19.533333	18.290000	3.626667	18.700000	20.790000	18.000000
50%	21.600000	20.000000	22.100000	20.666667	19.390000	7.300000	20.075000	22.111111	19.390000
75%	22.633333	21.500000	23.340000	22.100000	20.653889	11.226667	21.600000	23.390000	20.600000
max	26.260000	29.856667	29.236000	26.200000	25.795000	28.290000	26.000000	27.230000	24.500000

	RH_1	RH_2	RH_3	RH_4	RH_5	RH_6	RH_7	RH_8	RH_9
count	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000
mean	40.267556	40.434363	39.243995	39.043799	51.014065	54.615000	35.410874	42.948244	41.556594
std	3.974692	4.052420	3.245701	4.333479	9.107390	31.160835	5.097243	5.210450	4.161295
min	27.023333	20.596667	28.766667	27.660000	29.815000	1.000000	23.260000	29.600000	29.166667
25%	37.363333	37.900000	36.900000	35.560000	45.433333	29.996667	31.500000	39.096667	38.530000
50%	39.693333	40.500000	38.560000	38.433333	49.096000	55.267500	34.900000	42.390000	40.900000
75%	43.066667	43.273453	41.730000	42.200000	53.773333	83.226667	39.000000	46.500000	44.326667
max	63.360000	54.766667	50.163333	51.090000	96.321667	99.900000	51.327778	58.780000	53.326667

	T_out	Tdewpoint	RH_out	Press_mm_hg	Windspeed	Visibility
count	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000	14801.000000
mean	7.421836	3.782509	79.824197	755.480135	4.029001	38.290284
std	5.343737	4.194994	14.901776	7.389218	2.448171	11.789650
min	-5.000000	-6.600000	24.000000	729.300000	0.000000	1.000000
25%	3.666667	0.933333	70.500000	750.900000	2.000000	29.000000
50%	6.933333	3.483333	83.833333	756.000000	3.666667	40.000000
75%	10.433333	6.600000	91.666667	760.833333	5.500000	40.000000
max	26.100000	15.316667	100.000000	772.300000	14.000000	66.000000

	Appliances
count	14801.000000
mean	97.875144
std	102.314986
min	10.000000
25%	50.000000
50%	60.000000
75%	100.000000
max	1080.000000

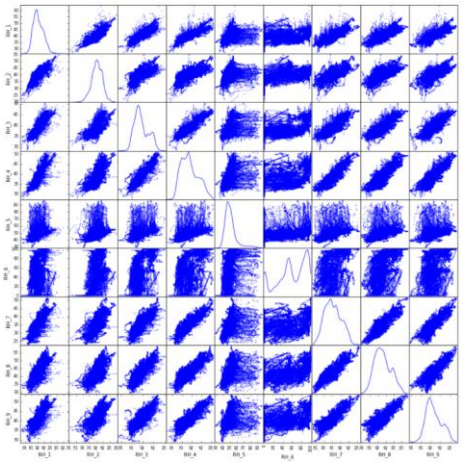
Scatter plots:



The scatter plots are found for features from T1 to T9 So we found a degree of correlation between T7 and T9 which has a value of Correlation of coefficient : 0.9460586115166221

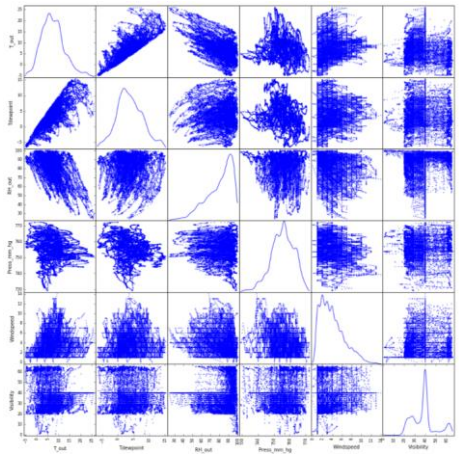
p-value : 0.0.(This can be seen in the ipython notebook code no 51)

Scatter plots for RH_1 to RH_9:

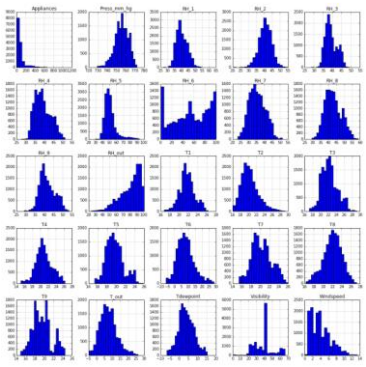


There is no significant correlation exists among different humidity values and weather among weather parameters like Pressure, Windspeed, Temperature, etc. which can be confirmed from the plots b. and c**.**

Weather data:



Distribution of all the columns depicted using histogram graph:

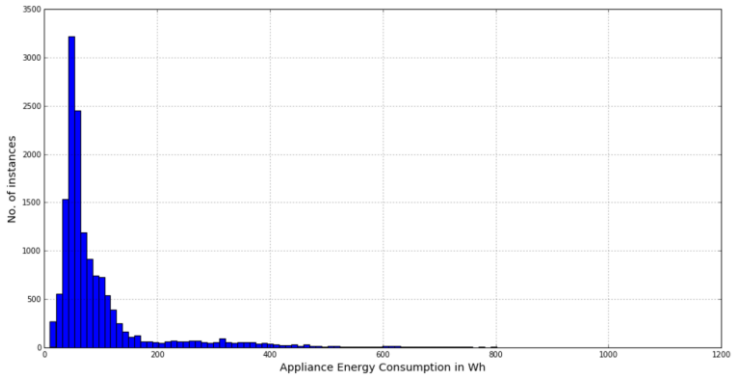


From the above plot, it is clear that no columns have a distribution like the **Appliances** column, which is our target variable. Therefore, there is no linear relationship of any single feature with the target variable independently.

Exploratory Visualization

Here in the above visualized plots especially in case of the histograms it is clear that most of them have normal distribution curves while some have left/right skewed curves like the T_2 and RH_2. Also some don't have a normal distribution curves like the Rh_6 and visibility.

Distribution of the target variable:



Observations:-

i. Most features are normally distributed.
ii. The target variable has a highly skewed distribution and it doesn't have linear relation with any other features.
iii. The feature ***T9*** is highly correlated with features ***T3***, ***T5*** and ***T7***.
iv. The feature ***T6*** is highly correlated with feature ***T_out***.

Algorithms and Techniques:

The most important algorithm used in this project is the is Linear Regression as it can explain the data well, there is no need for further complexity. Future as a modification to original Least Squares Regression, we can use the Regularization techniques to penalize the coefficient values of the features, since there is a tendency for the higher values being over fitting and loss of generalization. Regularization techniques enhance performance of linear models in a great way. Also linear data being fitted without reguralization perfectly persists to very few cases. Here the problem of Linear Regression is transformed into Lasso or Ridge Regression respectively.

The next algorithms used is the tree basedregression models which can robust to outliers than linear regressors. Tree based models

1. Random Forests

2. Gradient Boosting Machines

3. Extremely Randomized Trees

Finally, neural network can also be considered in case of non linear primary algorithm. They work great when there is a complex nonlinear relationship between the inputs and the output. Neural Networks are Multi-layer Perceptron

Benchmark:

The benchmark model is Linear Regression on unscaled data using all the features.

Observations:

i) R2 score on training data: 14.687%
ii) R2 score on test data: 14.258%
iii) RMSE on test data = 0.926

Methodology

Data Preprocessing :

Ranges of features irrespective of units

Temperature	-6 to 30
Humidity	1 to 100
Windspeed	0 to 14
Visibility	1 to 66
Pressure	729 to 772
Appliance Energy Usage	10 to 1080

Due to difference in the ranges of features, there is a possibility that some features can dominate the regression algorithm leading to over fitting so to avoid this all features need to be scaled which is done using a function called standard_scaler imported from the sklearn preprocessing module.

Here a row of data is shown before and after scaling.

Before scaling:

T1	20.200000
RH_1	37.500000
T2	17.823333
RH_2	39.300000
T3	20.290000
RH_3	36.560000
T4	18.200000
RH_4	37.290000
T5	17.926667
RH_5	47.633333
RH_6	67.666667
T7	18.463333
RH_7	29.390000
T8	21.390000
RH_8	35.663333
RH_9	35.500000
T_out	2.800000
Press_mm_hg	744.000000
RH_out	86.666667
Windspeed	2.666667
Visibility	28.000000
Tdewpoint	0.766667
Appliances	70.000000

After scaling:

T1	-0.923012
RH_1	-0.696318
T2	-1.144741
RH_2	-0.279932
T3	-0.988045
RH_3	-0.826966
T4	-1.299016
RH_4	-0.404723
T5	-0.907291
RH_5	-0.371220
RH_6	0.418863
T7	-0.854395
RH_7	-1.181242
T8	-0.325420
RH_8	-1.398182
RH_9	-1.455508
T_out	-0.864936
Press_mm_hg	-1.553686
RH_out	0.459187
Windspeed	-0.556489
Visibility	-0.872853
Tdewpoint	-0.718939
Appliances	-0.272454

We also removed the columns ***T6*** and ***T9*** which had a significant correlation with columns ***T_out*** so there are 22 features in the training data.

Implementation:

A function is created called the ***pipeline_1()*** function to execute each Regressor and record the metrics. Then each Regressor is passed to above defined function from ***execute_pipeline_1() function which is also defined later.***. The above obtained metrics are consolidated into a DataFrame with the help of the function ***get_properties_inf()*** and all these metrics are plotted on a graph.

List of Algorithms tested:

i) sklearn.linear_model.Ridge
ii) sklearn.linear_model.Lasso
iii) sklearn.ensemble.RandomForestRegressor
iv) sklearn.ensemble.GradientBoostingRegressor
v) sklearn.ensemble.ExtraTreesRegressor
vi) sklearn.neural_network.MLPRegressor

Results:

	RMSE	Testing scores	Training scores	Training times
Ridge	0.936121	0.123677	0.137409	0.0260139
Lasso	1	0	0	0.0315361
RandomForestRegressor	0.728899	0.468707	0.91342	12.097
GradientBoostingRegressor	0.86821	0.246212	0.331539	6.69755
ExtraTreesRegressor	0.664811	0.558027	1	3.38832
MLPRegressor	0.844788	0.286334	0.353317	17.8957
Linear Regression (Benchmark)	0.926026	0.142476	0.146873	0.0375407

From results shown above , we can see that ***ExtraTreesRegressor*** performs better than all other regressors in terms of all metrics except for Training time where Linear models outperform it.

Refinement:

For refining the model, the following properties of ExtraTreesRegressor are consideredwhile analysing:

i) n_estimators: The number of trees to be used.
ii) max_features: The number of features to be considered at each split.
iii) max_depth: The maximum depth of the tree.

On considering above parameters we can see that Before Tuning, the R2 score on test set was 0.558. After tuning, it rose to 0.610, a performance gain of **5.2%**.

Challenges faced:

i) There is need for Feature scaling for Regression to avoid overfitting.
ii) Seed generator is used at the beginning for the reproducible results.
iii) For the purpose of maintain separate copies of DataFrames with scaled data, it is better to create dummies using original DataFrame's index and columns and then filling it with scaled data
iv) The pipelines considered above should b modular.
v) Cross validation is very useful for finding out the best model.

Results for model validation and evaluation:

i) R2 score on test data = **0.499**.
ii) R2 score of untuned model = 0.558.
iii) Difference = 0.059 or 5.9%.
iv) RMSE on test data = 0.708
v) RMSE of untuned model = **0.665**
vi) Difference = 0.343

Therefore, it is evident that even though the feature space is reduced drastically (by more than 75%), the relative loss in performance on test data is less.

Justification:

Parameters/Models	Final Model	Benchmark Model	Difference
Training R2 score	1.0	0.147	0.853
Testing R2 score	0.61	0.142	0.468
RMSE on test data	0.624	0.926	0.302

Based on the results above which seems to be improved , the final tuned model can be considered as a satisfactory solution.

Conclusion:

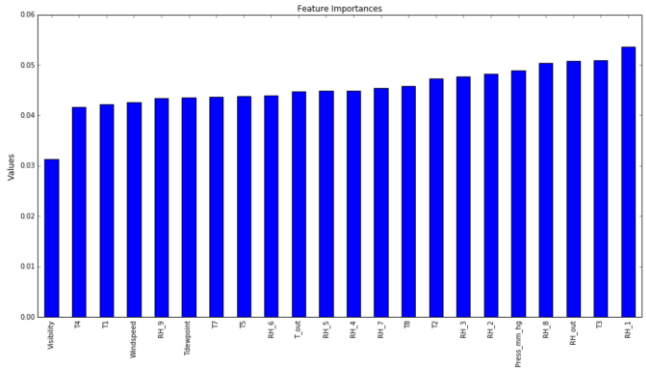
Free-form visualization: For a the best fitted model to be used in future , the feature importance can be considered is as follows:

```
Most important feature = RH_1
Least important feature = Visibility

Top 5 most important features:-
RH_1
T3
RH_out
RH_8
Press_mm_hg

Top 5 least important features:-
Visibility
T4
T1
Windspeed
RH_9
```

Visual representation of all features:



From the above graph it can be observed that , humidity affects power consumption on an average more than temperature. So it is evident the number of humidity reading are on the higher end than the temperature readings. Also, from observing the weather parameters, Humidity and Atmospheric pressure affect power consumption more nthan others. So this forms a general assumptional conclusion that factors like Windspeed and Visibility shouldn't affect the power consumption inside the home. An important conclusion drawn from this visualization controlling humidity inside the home can lead to energy savings even though they cannot be controlled outside.

Reflection:

This project can be summarized as the sequence of following steps:

1. Searching for a problem by looking at datasets deciding between Classification and Regression problems.
2. Visualizing various aspects of dataset.
3. Preprocessing the data and feature selection.
4. Deciding the algorithms to be used to solve the problem.
5. Creating a benchmark model.
6. Applying selected algorithms and visualizing the results.
7. Hyper parameter tuning for the best algorithm and reporting the test score of best model.
8. Discuss importance of selected features and check the robustness of model.

Out of the above steps, we found that, 1, 2 and 5 seems to be very interesting because deciding between Classification and Regression was an important challenge and in my project after several approaches between the classification and regression which i have done before and considering its effects, I decided that it would be more better and challenging to solve a Regression based problem. Also, visualizing a dataset from the point of view of solving a Regression problem where your output isn't defined among a few classes was particularly challenging.

Improvement :

A few of the ways the solution can be improved are:

1) Discarding unimportant features from the data set, which in case of this project i have taken into consideration the wind speed and visibility.
2) Performing a more aggressive feature engineering
3) Using of grid search for searching of parameters and to determine the better or best solution
4) Also more number of parameters can also be added if needed.