Assessment 3 Report :

This report aims to predict the air conditioner status based on predictors in the dataset using Machine Learning models. We apply classification models using various hyperparameter tuning and ensemble models to check whether the accuracy score is better than the base models or not.Below are the following steps taken to reach the result.

Q-1
Conducting EDA on the Train dataset :

The dataset contains 12 variables namely load, ac, hourofday, dayofweek, dif, absdif, max, var, entropy, nonlinear and hurst with 417720 rows. We conduct a five-point summary for all the variables to determine the mean, median, quartiles and any unique values. The following table highlights the variable load, ac and hourofday. Since, dayofweek is a categorical variable, we cannot compute a five point summary for it. Furthermore, there are no missing values in this dataset, hence, we skip the step of imputing any missing values.

Variables	Mean	Median	Quartiles (25%) & (75%)	Unique Values
Load	2.184664	1.279000	0.807000 3.358000	NaN
Ac	0.242265	0.000000	0.000000 0.000000	NaN
Hourofday	11.484487	11.000000	5.000000 17.000000	NaN

We can observer that since “ac” is a binary variable, hence the median and quartiles are 0.
Load variable (also known as load monitoring which provides detailed electricity consumption and usage of individual appliances) has a mean of 2.18 with a standard deviation of 1.89. Similarly, hourofday has a mean of 11.48.

Distribution of the variables :

We implement the seaborn.distplot() function from the Seaborn module to plot the distribution. Distplot depicts the univariate distribution of data, implying a data distribution of a variable against the density distribution. For instance, the Figure 1. below shows distribution of the load variable. It has two peaks, density ranging from 0.0 to 0.8 with a short peak between 4 and 6.

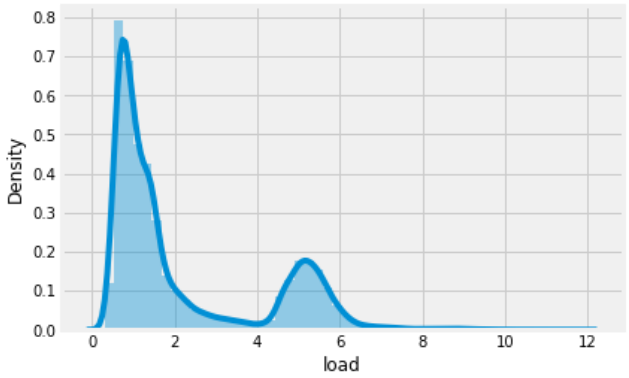


Figure 1

The hour of day is a time series variable, hence the distplot shows a constant flow with density being 0.04. This shows a uniform distplot.

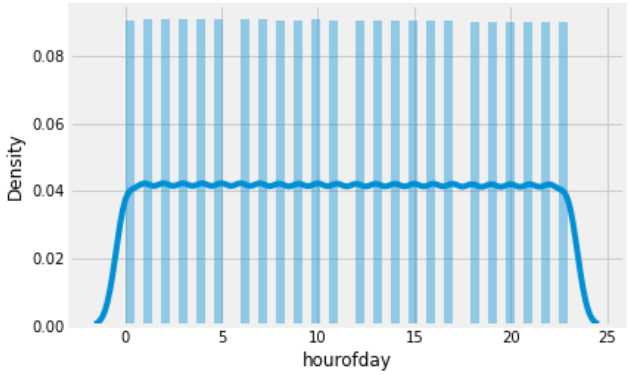


Figure 2

The entropy distplot displays a peak density of 8 between 0.6 and 1.0.

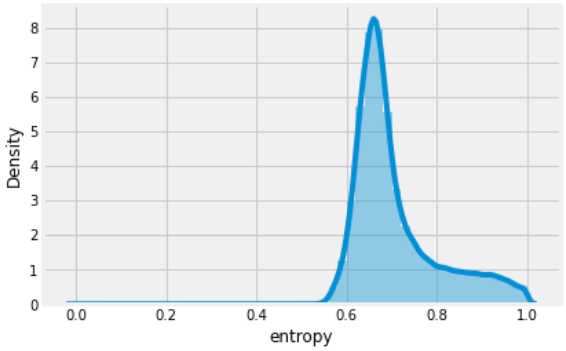


Figure 3

Distribution of the target variable :

The ac variable is a binary variable with unique values of 0 and 1. The figure below indicates when the air conditioner is on or not.

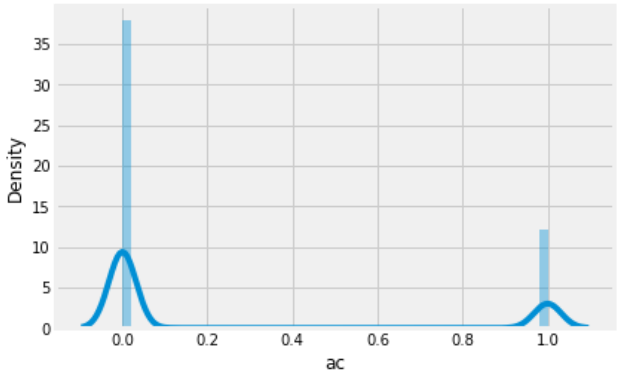


Figure 4

Correlation between the variables : The correlation matrix of the Energy data computes the Pearson's correlation coefficient between the variables. The correlation matrix is shown below:

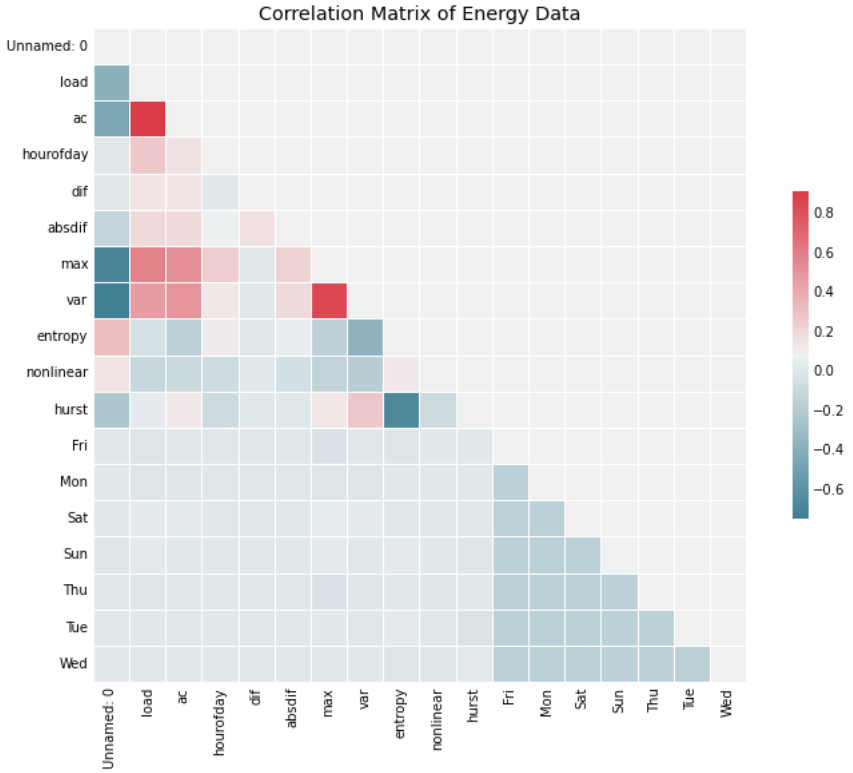


Figure 5

Barplots between the variables :

1) The first barplot indicates the relationship between a numeric dayofweek (categorical variable) and load (numerical) variable. We can observe that Saturday and Sunday have the highest load of more than 2.0, followed by Tuesday and Wednesday.

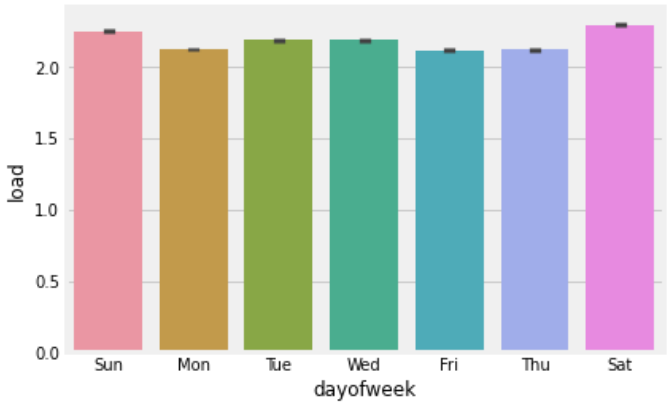


Figure 6

2) The barplot displays the relationship between hourofday and load variable. Since it is a time-series data, it shows a significantly higher amount of load during the evening time. Consumers mostly use electricity during evening and night time since there is no natural daylight.

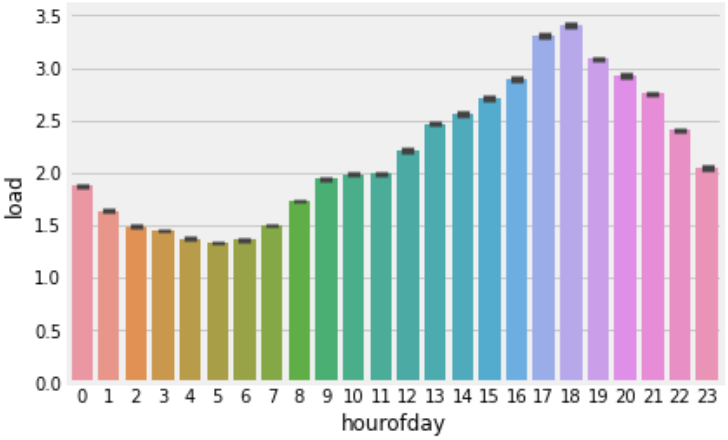


Figure 7

3) This displays the relationship between dayofweek and ac variable. Similar to the one above, as well as the relationship between hourofday and ac variable. The shape of the barplot is similar to the barplots above.

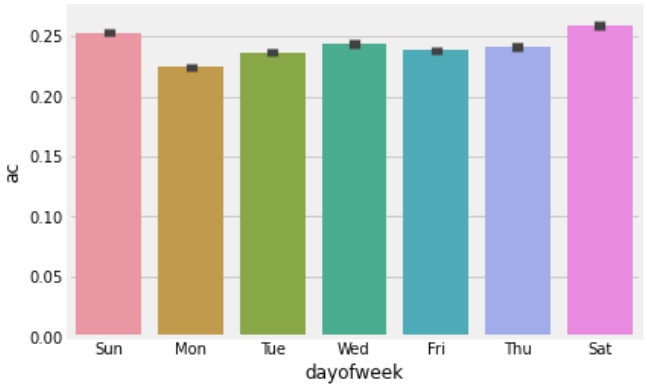


Figure 8

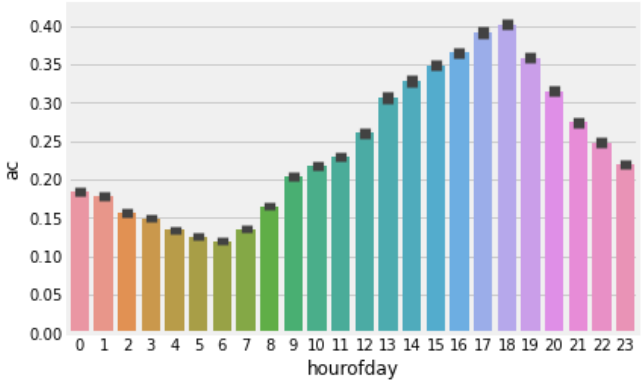


Figure 9

Q-2

Feature Importance

For Feature selection and importance, we first create dummy variables for the categorical variable "dayofweek". This is required since we need a numerical output for when we conduct our classification process.

1) **Chi-square approach :** This is used for categorical features in the data. We calculate Chi-square between each feature and the target and select the desired number of features with best Chi-square scores. It determines if the association between two categorical variables of the sample would reflect their real association in the population.

Therefore, we get the best features with chi-square score:

Load	3.99093010e+04
Hourofday	1.73963331e+05
Max	4.13790067e+00

Var	1.07432226e+02
Entropy	8.84745145e+01
Nonlinear	3.85898220e+01
Hurst	6.10375228e-01
Mon	1.02410883e+01
Sat	1.02742709e+00
Sun	2.55393169e+01

2) **Univariate analysis by using the SelectKbest function :** We compare each feature to the target variable, to see whether there is any statistically significant relationship between them, also known as the p-value. We get the same features as above.

Q-3

The three classification models used here are : Logistic Regression, Decision Tree, K-NN Classifier model. We will be using different performance metrics to gauge the best performing model :

a) **Performance score for the ML models :**

Model	Accuracy Score	Precision	Recall	f1 score
Logistic Regression	0.98377	0.93	0.89	0.91
Decision Tree	0.98310	0.89	0.91	0.90
K-NN	0.97762	0.85	0.89	0.87

Since, there is a problem of class imbalance, we also evaluate the Cohen kappa score for Logistic Regression which is **0.90124.**

The above table therefore displays how Logistic Regression has the highest accuracy score of 0.98377. The classification report function further displays :

- **Precision score** – the % of predictions that were accounted true, which is 0.93 (when ac is on), higher than Decision Tree and K-NN.
- **Recall score** – the % of all positive instances, which is 0.89 (when ac is on).
- **f-1 score** – weighted harmonic mean of recall and precision scores. Logistic regression has the highest f-1 score of 0.91, which is closer to 1.0.

The above models indicate overfitting since, after a certain point the accuracy starts to decrease. Decision trees, in general, have common drawbacks of overfitting, requiring

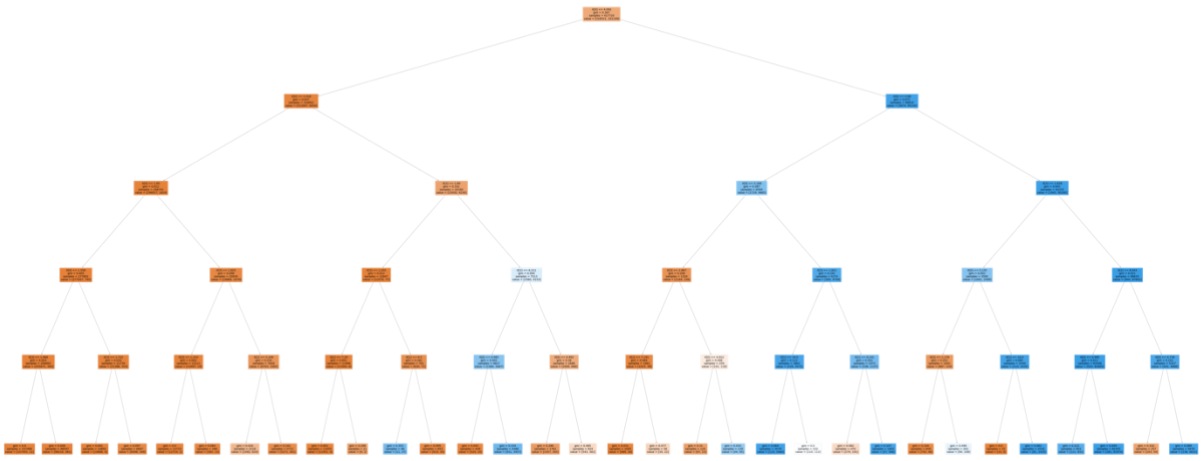
pruning and optimisation of hyperparameters. They are also prone to bias. Thus, it is imperative to take steps to prevent overfitting, for instance, using ensemble methods.

b) Different designs for classification model :

For this dataset, we use the binary Logistic Regression, since our response variable, "ac" is a binary variable implying that it can only belong to one of two categories.

With the Decision Tree, the design of the model used is CART (Classification and Regression Tree) which uses Gini method to create split points including Gini Index and Gini Gain.
The figure below proves that :

Gini Index – CART



The K-NN on the other hand is just a K-NN classifier which assumes that similar datapoints exist in proximity.

c) Hyper parameter tuning – Different types of accuracy scores

Hyperparameters are the settings that can be tuned before running the training data to control the behaviour of the ML model. They can have a huge impact on model training as it relates to training time, infrastructure resource requirements (cost), model convergence and model accuracy.

Model parameters are learnt as part of training process, whereas the values of hyperparameters are set before running the training job and they do not change during the training process.

Hyperparameter types –

- **Penalty, Random state, Max iter and Inverse of regularization in Logistic Regression.**
- **Maximum depth, min. samples required at lead node in DTs.**
- **k in K-NN model.**

Generalization (test) error of learning algorithms has two main components :
Bias and variance. The trade-off between these components is determined by the complexity of the model and the amount of training data. The optimal hyperparameters thus, helps to avoid under-fitting and over-fitting (training error is low but test error is high).

Testing accuracy after Hyperparameter tuning :

Models	Hyperparameters	Accuracy Score pre-tuning	Accuracy Score post-tuning
Logistic Regression	lambda = 15	0.98377	0.98375
Decision Tree	max_depth = 10	0.98310	0.98310
KNN	N_neighbours = 7	0.97762	0.97824

We can see that for logistic regression and decision tree, not much has changed, however there has been a slight increase in the accuracy score of K-NN when the n_neighbours are 7.

d)
Henceforth, I would recommend choosing the Logistic Regression model even with the hyperparameter tuning since it has the highest f-1 score, as well as the precision score.

Q-4

a)

Ensembles are the combination of different machine learning models, where by averaging the results of all models trained together on the same dataset, the result can be away good than any individual model. These types of models can be used for all kinds of tasks, including classification, regression and detecting anomalies. Ensemble methods are employed when you want to improve the performance of the machine learning models. For instance, to increase the accuracy of classification models or to reduce the mean absolute error of the regression models. Ensembling generally provides a more stable model, hence, overcoming the overfitting and underfitting issues of the models.

b)

The ensemble methods used in this case are : AdaBoost, Gradient Boosting and Random Forest Classifier.

Similarities :

The boosting techniques used which are AdaBoost and Gradient Boost have the ability to incorporate automated variable selection and model choice in the fitting process.

Moreover, the random forest classifier is built up on the Decision Tree, by constructing the trees independently and determining the output in any order.

Differences :

AdaBoost method automatically adjusts its parameters to the based on its actual performance in its current iteration. Implying that both the weights for re-weighting the data and the weights in the final aggregation and re-computed iteratively.

The difference lies in : Loss Function.
Adaptive Boosting minimised the exponential loss function which can make the algorithm sensitive to the outliers. However, Gradient Boosting can utilise any differentiable loss function. This model is robust to outliers as compared to AdaBoost.

c) We can use Random Forest Classifier can work wonders on large datasets and interpretability is also not a major concern. It constructs decision trees on different samples and takes the majority vote for classification and average in case of regression. Important features of Random Forest are :

1) **Diversity – Each tree is independent, and not all attributes are considered.**
2) **Parallelization– there is no higher computing power required to construct the trees.**
3) **No curse of dimensionality – the feature space in RF is reduced.**
4) **Stability – The result is based on majority voting / averaging.**

d)

Models	**Accuracy Score**
Logistic Regression	0.98375
Decision Tree	0.98310
KNN	0.97824
AdaBoost Classifier	0.98341
Gradient Boosting Classifier	0.98519
Random Forest Classifier	0.97801

Hence, from the table above, we can confidently say that Gradient Boosting Classifier has the highest testing accuracy score of 0.98519. Judging from the model complexity and the performance metric, Gradient boosting often provides output that cannot be trumped. Additionally, it has lots of flexibility and can optimize on different loss functions ; works great with categorical and numerical attributes.

e) As far as I know, it is not possible to build any other ensemble method on ML classifiers other than random forest.

REFERENCES :

Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K., 2003, November. KNN model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 986-996). Springer, Berlin, Heidelberg.

Hooker, S., Erhan, D., Kindermans, P.J. and Kim, B., 2018. Evaluating feature importance estimates.

Rajagopalan, G., 2021. Data visualization with python libraries. In *A Python Data Analyst's Toolkit* (pp. 243-278). Apress, Berkeley, CA.

Retrieved from https://botbark.com/2019/12/25/top-5-hyper-parameters-for-logistic-regression/

Retrieved from https://medium.com/analytics-vidhya/why-hyper-parameter-tuning-is-important-for-your-model-1ff4c8f145d3

Retrieved from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Retrieved from https://www.baeldung.com/cs/gradient-boosting-trees-vs-random-forests

Retrieved from https://www.vebuso.com/2020/01/decision-tree-intuition-from-concept-to-application/

Zajac, N., 2021. Supplementary Material to:" The genome of the trematode parasite Atriophallophorus winterbourni, Blasco-Costa et al., 2019: A macro-and microevolutionary perspective".