Assignment 3 – sxb180048 – Sai Pratheek Banda
ENERGY DATASET

Classification Problem:

The aim of the problem is to predict the classification of the energy consumption into high and low. The interesting part of this classification is that it will help us understand the patterns for energy consumption thus understanding why energy consumption might be low or high. **Class 0 (Low)**: 10357 Observations| **Class 1 (High)**: 8048 Observations

EDA:

1) Features 'T_out, RH_7, T9, RH_4, T3, rv2, RH_9, T1, T7 ' were dropped due to high correlation.
2) Date, Visibility features are insignificant due to its inability to explain the target variable.
3) Lights Feature was removed due to high null values.
4) The outliers were removed based on the Inter Quartile Range.
5) Data has been Scaled for computational productivity.

Artificial Neural Network:

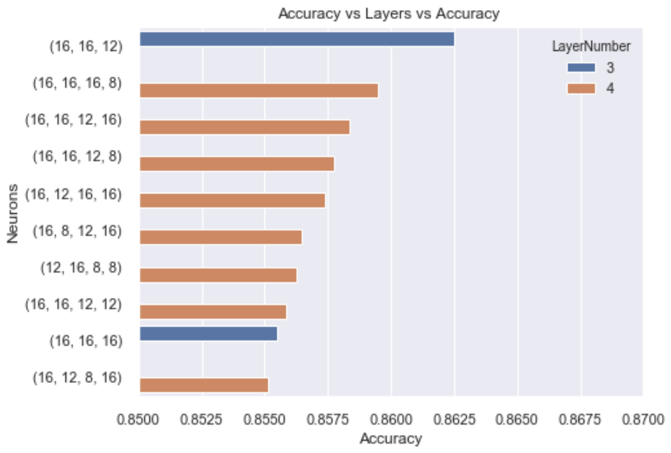
We are using a Multi-Level Perceptron for our Neural Network with the library Keras to create a Sequential Model.

Parameters:

In search for the best network of Perceptron's we experiment with 4 parameters while other parameters are at default:
i) Neurons ii) Epochs iii) Layers iv) Activation

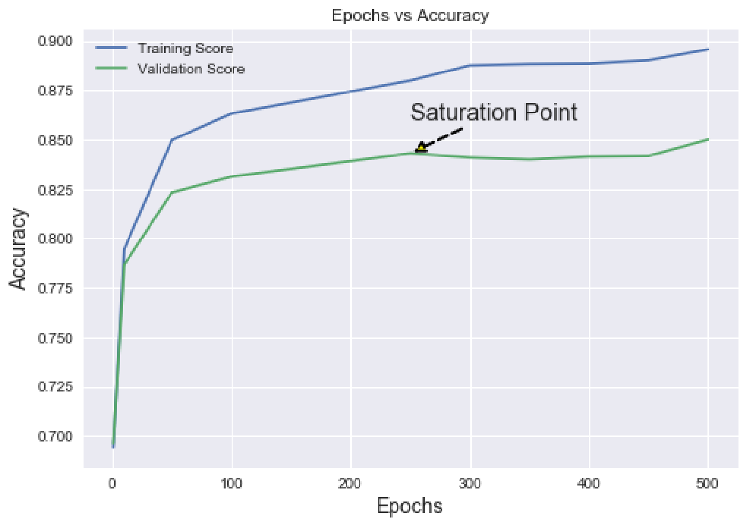
Neurons:

Experimenting with a short set of neurons for each hidden layer, number of layers together to find the best outline of the model was an expensive iterative process but the most important part of building the network. Cross validation though important was avoided due to the computational limitations.
The best network was found to be at **(16,16,2)**
First Hidden Layer: 16 Neurons
Second Hidden layer: 16 Neurons
Third Hidden Layer: 12 Neurons
which scored an Accuracy of **86.25%**



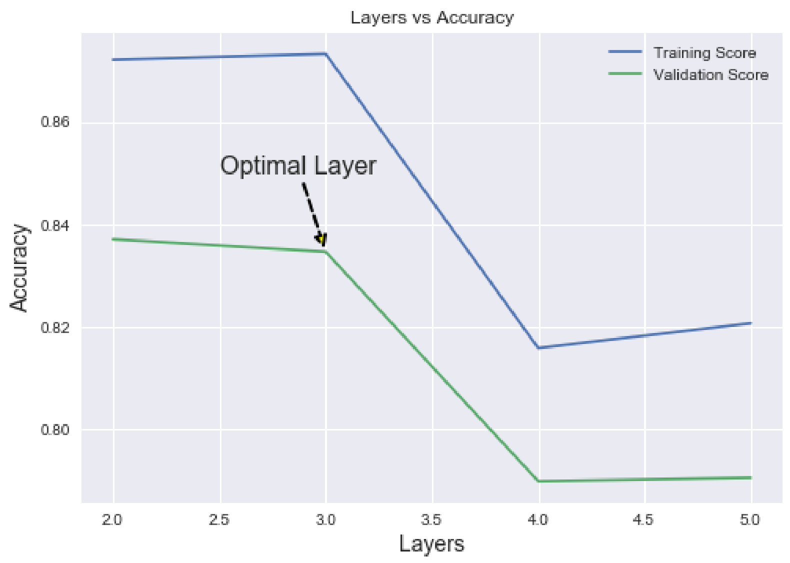
Epochs:

Experimenting with multiple Epochs with Cross Validation of 5 Sets, we redistribute weight to each neuron, resulting in a gradual increase in accuracy for both train and validation dataset, but reaches a saturation point at 250 Epochs with an accuracy of 84.3% on validation Data, which implies as to minute change in loss. Though Training sample increase with more iterations, it tends to fit the noise resulting in overfitting. At 250 Epoch the train and validation seem to model well thus is chosen as the best epoch parameter.



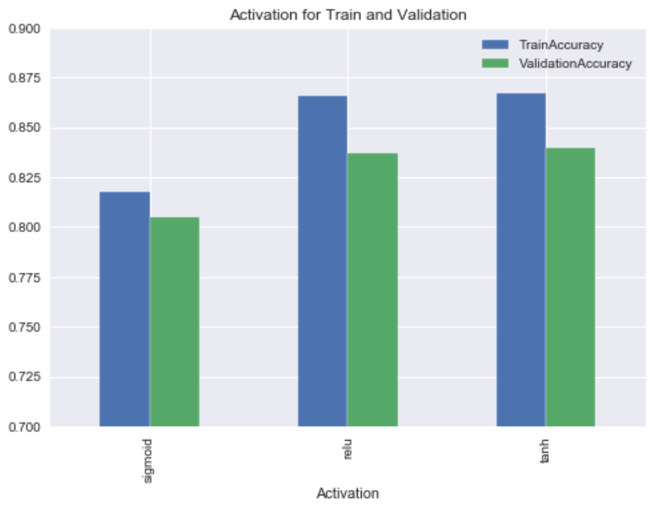
Layers:

An iterative expensive experimentation with Cross Validation of 5 sets was done to find the apt number of layers for the network with 250 epochs, which was found at 3 Hidden Layers, the train and test data take a significant dip of accuracy at 4 Layers, which implies the increase in loss after 3 Layers due memorization , This result is verified by first parameter experimentation with number of both Neurons and layers.
Thus, we choose the optimum parameter for Number of Layers as 3 which resulted in 83.46% accuracy on validation data.



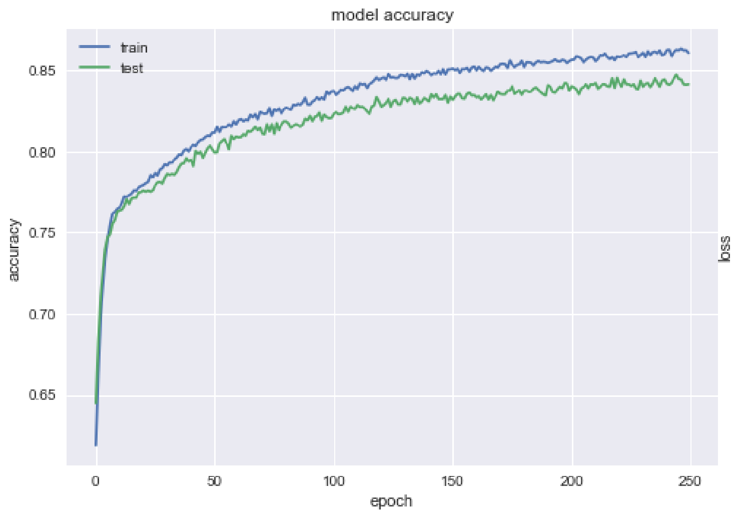
Activation:

The three activation functions that are experimented are 'Relu', 'Sigmoid', 'Tanh', But the last layer of the network is 'Sigmoid Activation', this is done as the classification is binary and sigmoid is the most apt activation as it its results are binary. In the process of experimentation 'Relu' was found to be the most accurate with 83.73% accuracy on the validation dataset, 'Relu' is less sensitive to the complexity of the problem since it is max function which return 0 or the value. Thus 'Relu' is chosen as the optimal Activation Parameter.

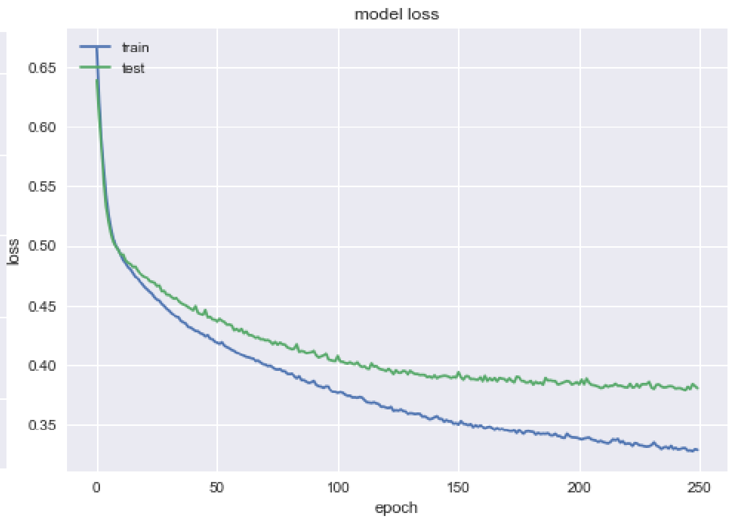


Learning Curve:

Accuracy:



Loss:



The above Learning Curve observes the change in metrics as with an increase in the epochs/ iterations for training data.

For the above model we plot a training curve which results saturation of accuracy at 86.25% at 250 epochs thus implying that further epochs will start to overfit the data which is shown by the validation dataset which converges and models as same the train dataset. The loss also gradually decreases over time and saturates at 250 on the validation dataset, though the training loss keeps decreasing which would fit to the noise.

As learning curve implements cross validation, we can conclude that the decrease in accuracy in training is the model trying not to overfit thus generalizing well.

Final Train and Test Model

Choosing the best parameters from all the above experiments, we fit the model to the train.

The Model performs well with an accuracy of 86.86%. Model performs well on test data with an accuracy of 84.88% which suggests a good model as it generalized well with unseen data as well.

```
Test Measures
       Loss  Accuracy  F1 Score  Precision    Recall
0  0.371832  0.848864  0.793554   0.838594  0.765888
```

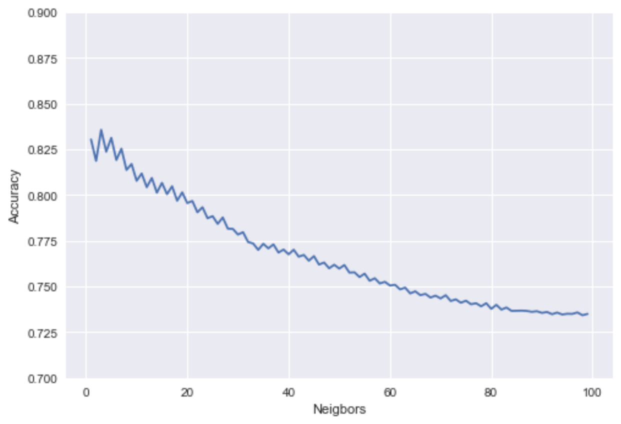
K Nearest Neighbors

Parameters:

In search for the best classification we experiment with 2 parameter i) Neighbors ii) Distance Metric

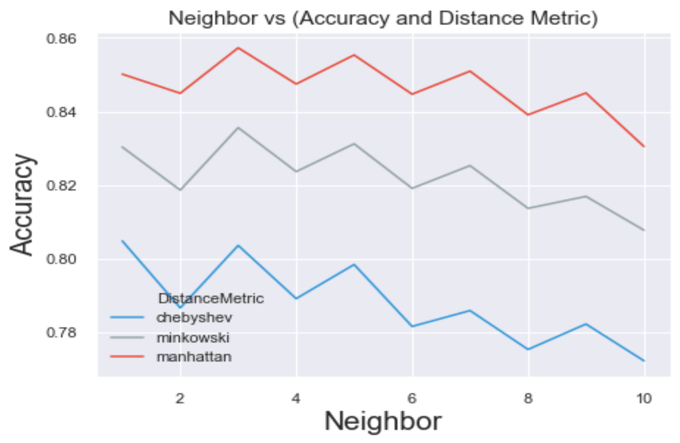
Neighbours

Experimenting with the number of neighbors from 1 to 100, we find that the model is most accurate at 3 Neighbors, as we increase the number of neighbors it tends to fit the noise, which also suits the narrative of choosing an Odd number of neighbors for best Practice. K can also be chosen as SQRT(N) but from the graph we can interpret the case to be the same, thus we choose the K value as 3 as the Optimum Neighbors parameter. Cross Validation with 3 splits also makes sure there is minimizes the bias in our results.



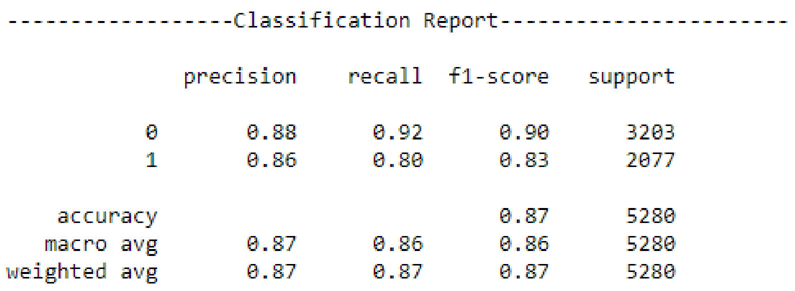
Distance Metric

Experimenting with three different Distance Metrics, we find that 'Manhattan' is the most accurate Distance metric for the classification with 3 Neighbors. Manhattan distance is the sum of horizontal and vertical distance between two points which suits best for our scaled data. We can also verify the number of neighbors here as it shows the highest accuracy with Manhattan Distance and 3 Neighbors. Cross Validation with 3 splits reduces the bias at the optimum.



Model with best Patamers on Test

K Nearest neighbors is used to classify the data into High and Low Energy categories with the best parameters found in the above experiments which resulted in an Accuracy of 87%. Though a lazy learner the KNN does a great job with the classification.

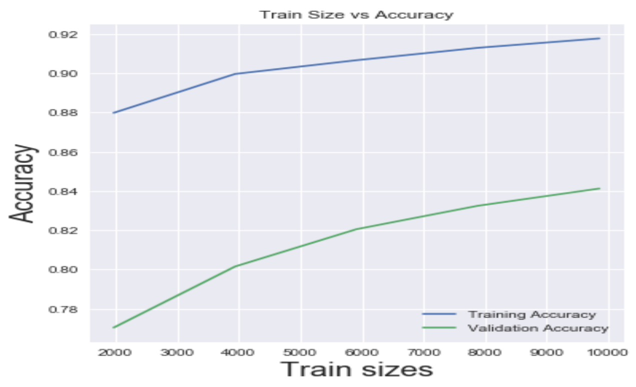


```
-----------------Classification Report----------------------

              precision    recall  f1-score   support

           0       0.88      0.92      0.90      3203
           1       0.86      0.80      0.83      2077

    accuracy                           0.87      5280
   macro avg       0.87      0.86      0.86      5280
weighted avg       0.87      0.87      0.87      5280
```

Learning Curve

For this model we plot a training curve which results in increase in accuracy up to 91% of train data, the validation data accuracy increases as it understands the data better with increase of influx training data.

As learning curve implements cross validation, we can conclude that model trying not to overfit thus generalizing well as both train and test move up gradually.



How All the Models Fare

Model	Accuracy	Comparison
Support Vector Machines	86.4%	Support Vectors are robust due to the margin gap and separating hyperplanes and is computationally efficient. The accuracy though not the highest it is the easiest one to implement with 3 different kernels.
Decision Tree	83.2%	The Decision tree is quick and does not need feature selection as it does it within itself. In the case of the dataset, the classification which contains both categorical and numerical values is not effective even with multiple leaf nodes, depth.
Ensemble Method (Boosting)	87.7%	The Ensemble method of Boosting is the Best measure when compared to all the model run till date. It improves on the Decision trees accuracy by 5.5%, which is highly significant. Boosting works on the estimators, the weak learners to reiterate itself to correctly classify them thus achieving a high accuracy.
Artificial Neural Network	84.88%	Artificial Neural Networks is a design which is like a black box. It is highly effective as it does not assume but works with itself to assign the importance of perceptron's i.e. the weights, the right network is all you need to have the most effective model but finding the right model needs a lot of computational power/GPU. ANN on this dataset does moderately well but less than imagined.
K Nearest Neighbor	87%	The simplest implementation of all the models which can separate data which is nonlinear. It is an unsupervised model which implies no training period, the model memorizes to predict the test data. It has only one hyperparameter and is a lazy learner which predicts with an accuracy almost equal to boosting which makes it highly competitive.