Dataset 1: Appliances energy prediction

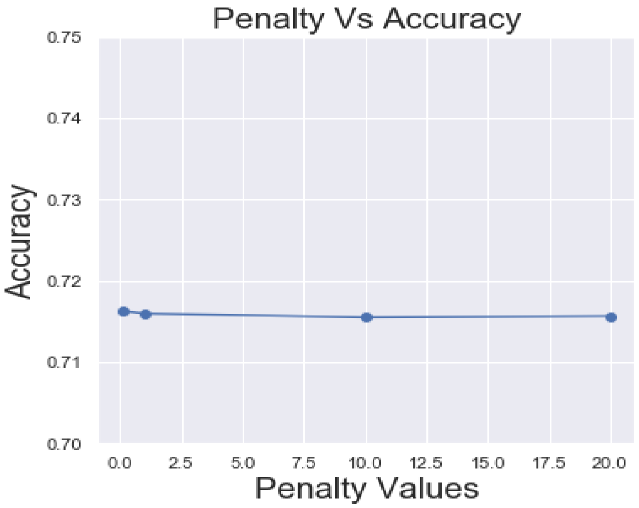
Classification Problem: Predict the energy usage (High or Low) of the appliances from the varying house temperature and humidity conditions that were monitor by a wireless sensor network. Other attributes like the weather from the nearest airport station are also included in this dataset.

Exploratory Data Analysis:

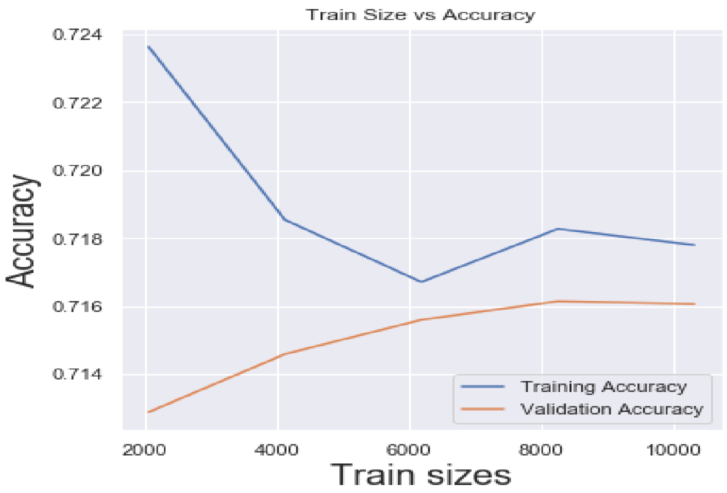
- Class 0 (Low): 10357 observations; Class 1 (High): 8048 observations
- Dropped features 'T3','RH_4','T5','T8','RH_7','T7','RH_8','T6','rv2','Visibility' from the dataset due to its high multicollinearity with the other feature variables.
- Dropped lights feature as it had almost 80% (15252 observations) of its value to be 0.
- Dropped features date, rv1, rv2 as they are just random and do no have any relation to the target variable

Support Vector Machine – Linear Kernel

Hyperparameter Tuning to find optimal Penalty (C)



Learning Curve using CV: Train size Vs Accuracy



Model implementation with cross validation to find the optimal hyperparameter:

C represents the severity of the penalty that is added to the support vectors. Accuracy is the same with varying penalities from 0.1 to 20.

Adding penalty does not have any affect on the accuracy with this dataset. So, the number of the points chosen as support vectors for varying penalities are the same.

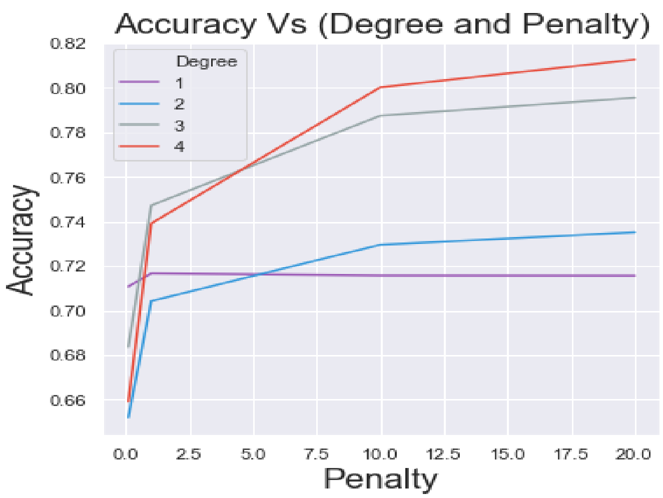
Parameters: Penalty = 0.1, kernel = linear, cv split = 3

With the increasing train size, the model is generalising well as the validation accuracy are increasing by small margins.

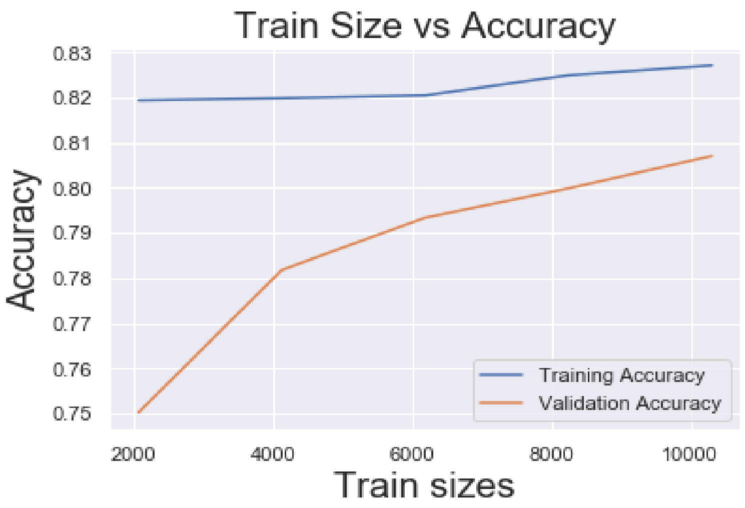
Model implementation with cross validation on increasing training sizes ensure the model does not overfit as the training and validation accuracies tries to converge. Saturation reached when the train size was 8000.

Support Vector Machine – Polynomial Kernel

Hyperparameter tuning to find optimal Penalty (C) and Degree



Learning Curve with Cross Validation: Train Size Vs Accuracy using the Best Parameters



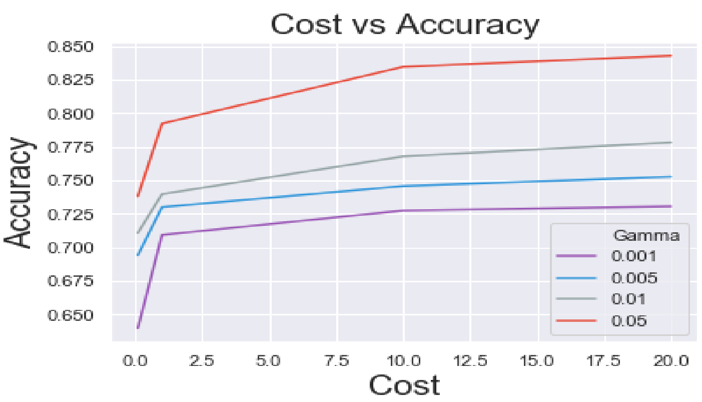
Model implementation with cross validation to find the optimal hyperparameters:
Degree allows for a more flexible decision boundary. ***Optimal cost and degree are 10 and 4***. After reaching this point, there is almost a saturation in the accuracy.

Parameters: Cost = 10, kernel = poly, degree = 4, cv = 3

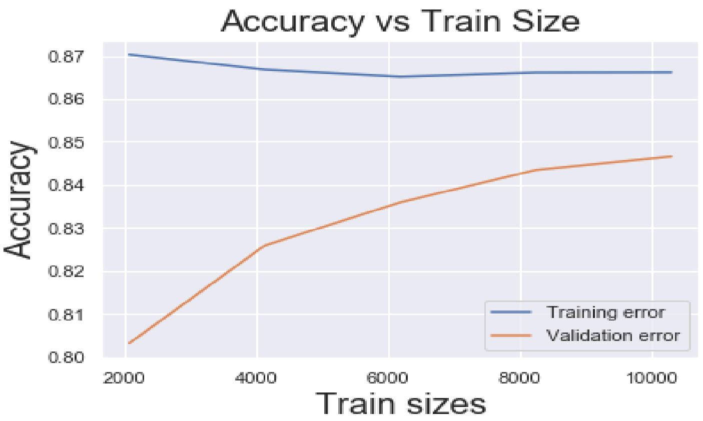
With increasing train sizes, performances in the training set is not increasing much but we have increasing accuracies in the validation data. So, the model is generalising well but it not learning much with increaasing data points.

Support Vector Machine – Radial Kernel

Hyperparameter tuning to find optimal gamma and penalty



Learning Curve with Cross Validation: Training size Vs Accuracy using the Best Parameters

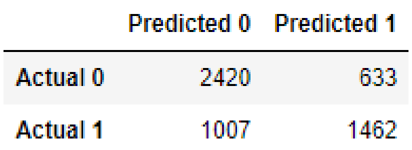


Model implementation with cross validation to find the optimal hyperparameters:
With the increase in Gamma, data points closer to the seperation line are used as support vectors and vice versa. **Optimal cost and gamma are 20 and 0.05**. After reaching this point, the accuracy reaches a saturation.

Parameters: Cost = 20, kernel = radial, gamma = 0.05, cv split = 3
Training accuracies are almost constant with increasing training size. There is a gradual increase in the validation accuracies with increase in the train sizes. More data will help the validation curve to converge to the training curve. Hence, the model generalises well but does not learn much.

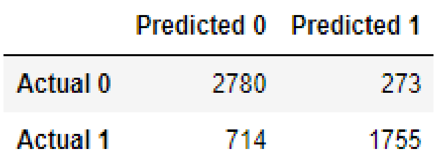
Test Result Predictions and evaluating the metrics of all the 3 kernels

Linear SVM



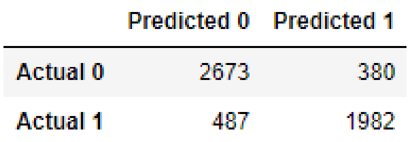
	Predicted 0	Predicted 1
Actual 0	2420	633
Actual 1	1007	1462

Polynomial SVM



	Predicted 0	Predicted 1
Actual 0	2780	273
Actual 1	714	1755

Radial SVM



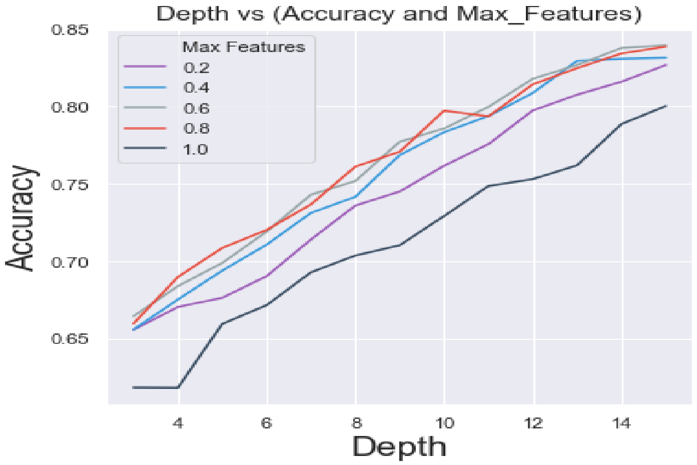
	Predicted 0	Predicted 1
Actual 0	2673	380
Actual 1	487	1982

Kernel	Test Accuracy
Linear	0.70
Polynomial	0.82
Radial	0.84

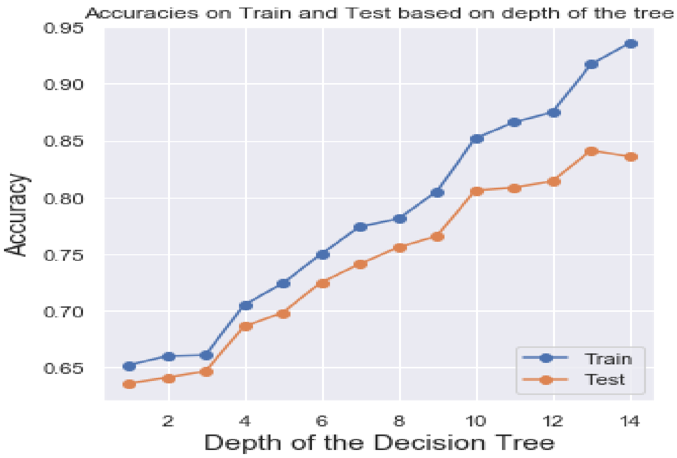
Radial Kernel SVM performs the best based on the test accuracy. From this, we can conclude that data points are mostly distributed across the center. It also makes sense as all the features are in units of Temperature and Humidity. For the very reason, the Linear kernel is not performing that well.

Decision Trees

Hyperparamter tuning to find the optimal depth of the tree and Maximum Features



Training and Test Accuracies with varying depth



Model implementation with cross validation to find the optimal hyperparameters:
As the depth of the increases, the validation accuracy also increases. This happens until a certain point where a saturation has been reached. Although cross validation does not overfit the model, we can experiment with depths ranging from 10 to 15 to come up with optimal depth. Optimal max_features parameter 10 is 0.8.

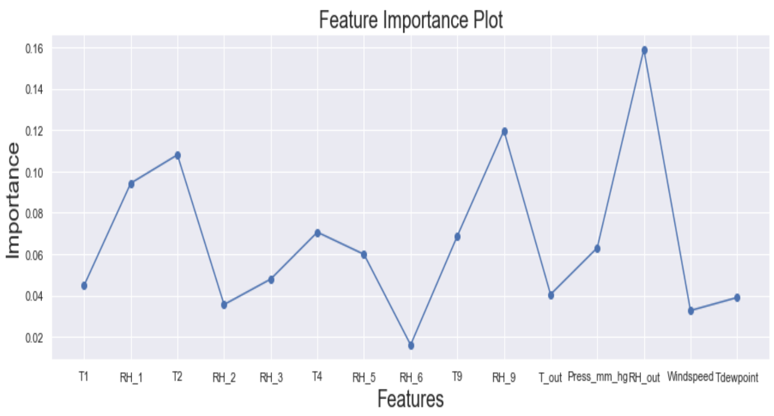
It is suprising to see that the accuracies are not good enough when maz_features = 1 (i.e all the features are selected)

Parameters: Max_features: 0.8, criteria: entropy

Training accuracies keeps increasing with increase in the depth of the tree. We need to choose an optimal depth based on the test accuracy. Depth = 10 seems to be accurate as the model is neither high bias or high variance.

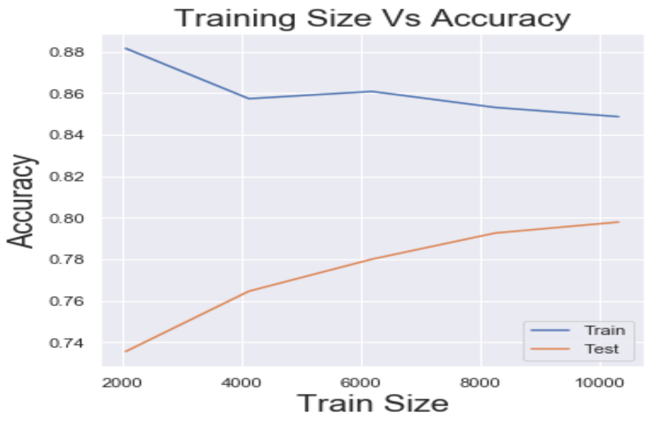
Although depths (13,14) shows better results than depth 10, it is performing too well on the training data and might be overfitting and may not perform well on other test datasets.

Feature Importances



Features RH_1, T2, RH_9, RH_out has atleast 10% of importance with RH_out contributing the most with importance close to 16%.

Learning Curve with Cross Validation

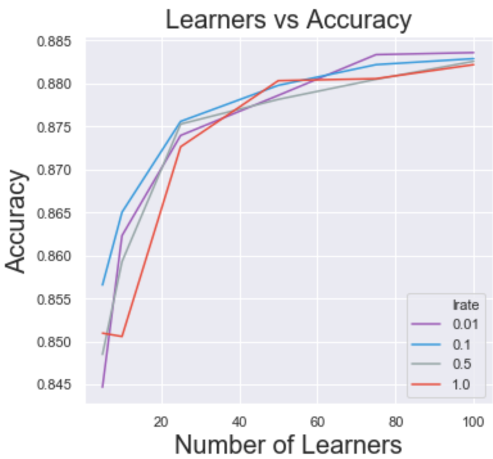


Parameters: Depth = 10, Max_features = 0.8

The model is generalising well as the number of obervations increases, the model can also attain a validation accuracy of 90% with more data points. It also shows that the model is a good fit.

Boosting the Decision Tree with AdaBoosting Algorithm

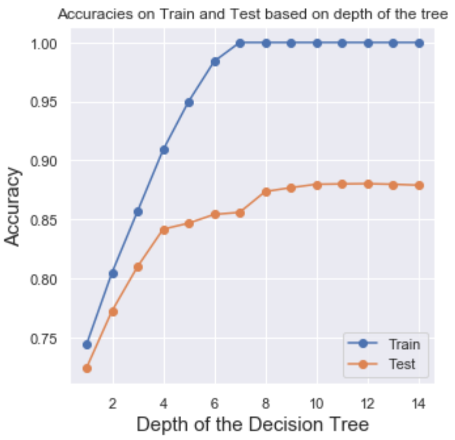
Finding the optimal learning rate and learners



Base: DecisionTreeClassifier with depth 10 and max_features 0.8

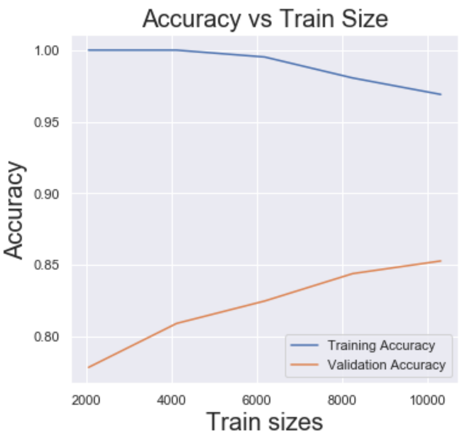
Very small accuracy changes with varying learning rates. Accuracy increases with number of learners and reaches the saturation point when the number of learners is 50.

Accuracies Vs Tree Depth



Training and test accuracies increase with increase in the depth of the tree. However, we prune the tree at **Depth = 5 or 6**. *50 DecisionTree weak learners with depth 5 is performing better than the single DecisionTree with depth 10*. With boosting there is a 5% increase in test accuracy. Although depths (13,14) shows better results than depth 10, it is performing too well on the training data and might be overfitting and may not perform well on other test datasets.

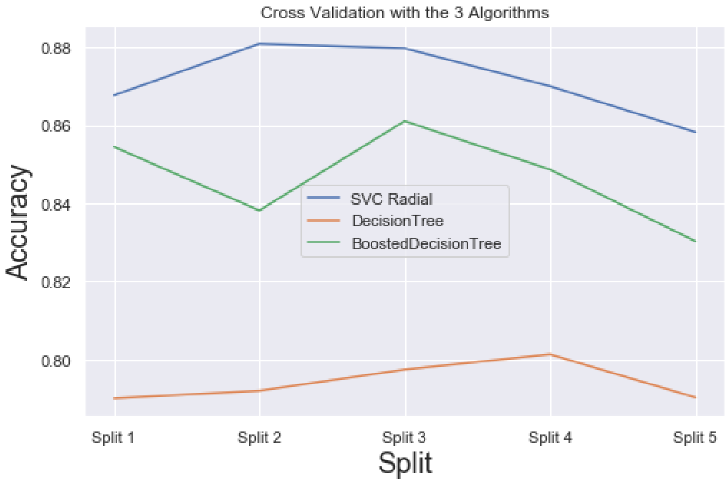
Learning Curve with CV on Train Size



Parameters: Number of weak learners = 50. DecisionTree Depth = 5, Max_features = 0.8

The model is generalising well as the number of observations increases. Validation Accuracy is 5% higher than the single Decision Tree

Cross Validation with SVM Radial, Decision Tree and BoostedDecision Tree



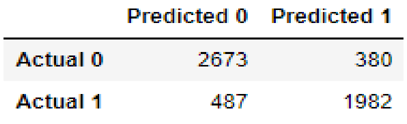
Although Boosted DecisionTree may have shown the best accuracy when compared to the other two algorithms, they are prone to overfitting.

From the cross-validation plot, we see that the SVM with Radial kernel outperforms the other two algorithms and it is not prone to overfitting.

Test Result Predictions and evaluating the metrics of all the 3 Algorithms

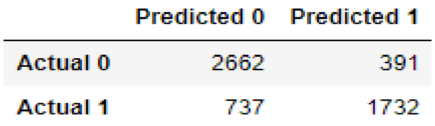
Confusion Matrix

Radial SVM



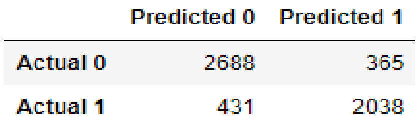
	Predicted 0	Predicted 1
Actual 0	2673	380
Actual 1	487	1982

DecisionTreeClassifier



	Predicted 0	Predicted 1
Actual 0	2662	391
Actual 1	737	1732

Boosted DecisionTree



	Predicted 0	Predicted 1
Actual 0	2688	365
Actual 1	431	2038

Classifier	**Test Accuracy**
Radial SVM	0.84
Decision Tree	0.80
Boosted DecisionTree	0.86

All the three algorithms are giving a test accuracy of atleast 80%. Boosted DecisionTree performs the best with the accuracy of 86%. But from the cross-validation plot, we can see that the Boosted DecisionTree may be prone to overfitting.

The Radial SVM is generalising better than the BoostedDecisionTree and its test accuracy (84%) is almost close to the Boosted Decision Tree. Hence, the Radial SVM can be considered to the best algorithm for this dataset.