# CS F320: FODS Assgnment 2

**Group Members:**

| | |
|---|---|
| Arjun Muthiah | 2019B3A70374H |
| Sanket Bhatt | 2019A7PS0147H |
| Hitaishi Desai | 2019B3A70602H |


## Question 2A

We began by preprocessing our data, standardizing it and creating a 80:20 training-testing split. We then calculated the covariances and subsequently the Pearson correlation coefficient of all of the features with the target attribute.
The correlation coefficients of all of the features are tabulated below.
Correlation Table (Descending order of magnitude of correlation of features with the target/dependent variable)

| **Feature** | **Correlation with Appliances** |
|---|---|
| **RH_out** | -0.161303 |
| **T6** | 0.115728 |
| **T2** | 0.104107 |
| **Windspeed** | 0.099430 |
| **T_out** | 0.098223 |
| **RH_8** | -0.088983 |
| **RH_6** | -0.081840 |
| **RH_1** | 0.077535 |
| **T3** | 0.073009 |
| **RH_2** | -0.064780 |
| **RH_7** | -0.059866 |
| **RH_9** | -0.046869 |

| | |
|---|---|
| **T1** | 0.038151 |
| **T4** | 0.034465 |
| **RH_3** | 0.030070 |
| **T8** | 0.027894 |
| **Press_mm_hg** | -0.020707 |
| **T7** | 0.016085 |
| **RH_4** | 0.013843 |
| **RH_5** | 0.012830 |
| **Tdewpoint** | 0.006252 |
| **T5** | 0.004402 |
| **Visibility** | -0.004089 |
| **rv1** | -0.000477 |
| **rv2** | -0.000477 |
| **T9** | -0.000158 |

Once the best features in order of their correlations were found, we built regression models with the best feauture, best 2 features and so on until we had built 26 models, the last including all of the features. The training and testing errors of those are tabulated below.

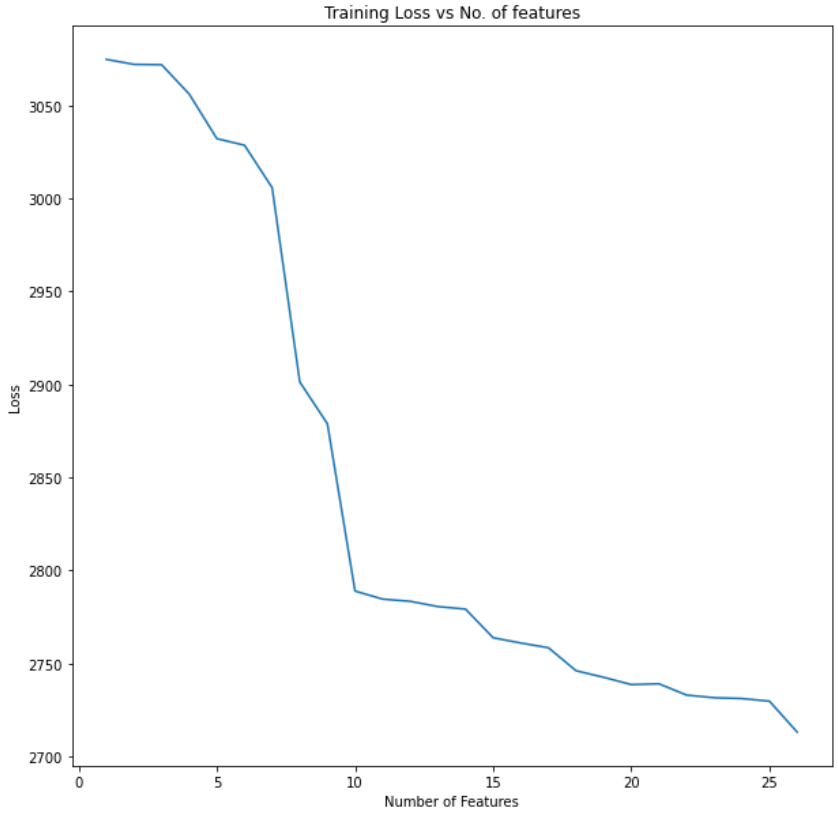
Error Table for selecting features from model based on correlation with Appliances

| **Number of Features** | **Training Error** | **Testing Error** |
|---|---|---|
| 1 | 3074.859095 | 773.067820 |
| 2 | 3072.167489 | 771.862965 |
| 3 | 3071.987684 | 771.219975 |
| 4 | 3056.168895 | 771.142216 |
| 5 | 3032.216043 | 767.365689 |
| 6 | 3028.668054 | 764.739219 |
| 7 | 3005.855997 | 756.296494 |
| 8 | 2901.353860 | 733.283750 |

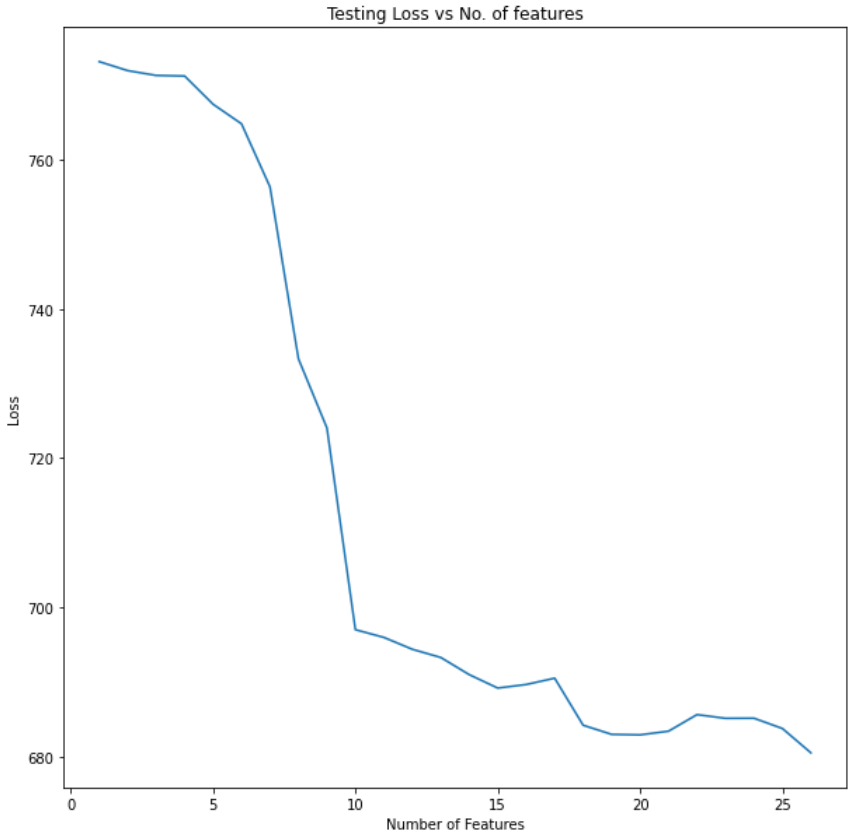
| | | |
|---|---|---|
| 9 | 2878.851167 | 724.034315 |
| 10 | 2788.876470 | 697.030222 |
| 11 | 2784.534018 | 695.997843 |
| 12 | 2783.334807 | 694.408392 |
| 13 | 2780.509641 | 693.308572 |
| 14 | 2779.146938 | 691.012884 |
| 15 | 2763.723163 | 689.215512 |
| 16 | 2760.911179 | 689.692525 |
| 17 | 2758.368081 | 690.540145 |
| 18 | 2746.002855 | 684.244519 |
| 19 | 2742.456468 | 683.019264 |
| 20 | 2738.546067 | 682.953432 |
| 21 | 2738.976209 | 683.445370 |
| 22 | 2732.894148 | 685.661840 |
| 23 | 2731.482650 | 685.167724 |
| 24 | 2731.045416 | 685.181467 |
| 25 | 2729.622624 | 683.804988 |
| ***26*** | ***2713.040922*** | ***680.551286*** |

The features used in the model with 26 features:
**['RH_out', 'T6', 'T2', 'Windspeed', 'T_out', 'RH_8', 'RH_6', 'RH_1', 'T3', 'RH_2', 'RH_7', 'RH_9', 'T1', 'T4', 'RH_3', 'T8', 'Press_mm_hg', 'T7', 'RH_4', 'RH_5', 'Tdewpoint', 'T5', 'Visibility', 'rv1', 'rv2', 'T9']**

Training Loss vs No. of features graph for model based on correlation with appliances




Testing Loss vs No. of features graph for model based on correlation with appliances



## Principal Component Analysis

When dealing with larger datasets, we are faced with the challenge of reducing the number of dimensions to reduce our computation time, while also preserving all of the variability and information the dataset gives us.
To remedy this, we use principal component analysis. We try to find a vector to project all of our features onto such that maximum variance is preserved.
We begin by computing the variance-covariance matrix of all of the features, and we then find its eigenvalues and eigenvectors. This allows us to reduce dimensionality without losing much information, i.e. with the largest amount of variance.
The eigenvectors represent the direction of the axes where there is the most variance, and the eigenvalues represent the amount of variance carried in each principal component.

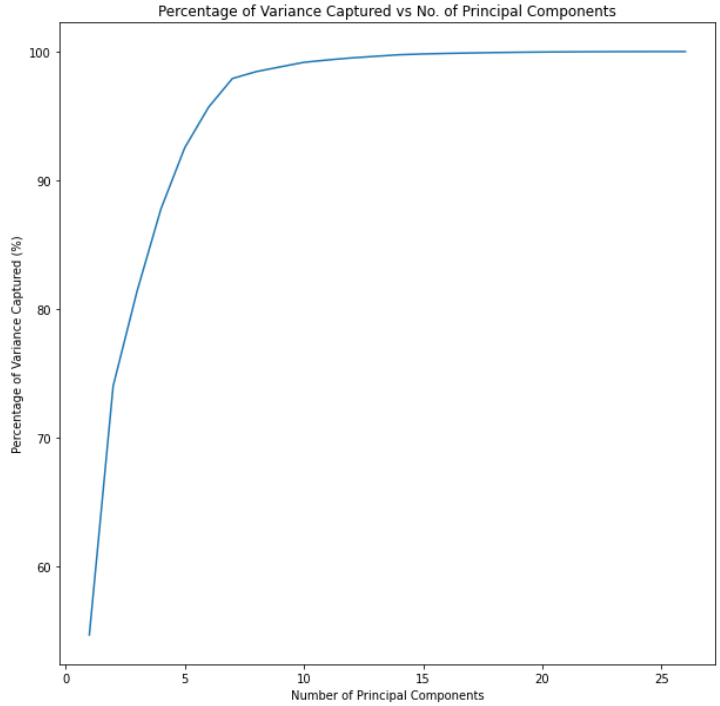
We started off by taking the eigenvector with the highest eigenvalue, and calculated the first principal component by taking the dot product of that eigenvector with each of the training examples. We trained this model on the training dataset.
The second principal component was calculated by taking the dot product of the eigenvector with the second highest eigenvalue, and trained this on the training data. We continued this process until all 26 of the principal components were included in the model.

The eigenvalues and the percentage of variance of the original dataset captured by the corresponding principal components has been given in the table below.
Eigen Value Table

| **No. of features** | **Eigenvalue** | **Percentage of Variance Captured** |
|---|---|---|
| 1 | 1160.813811 | 54.686415 |
| 2 | 410.555786 | 74.027866 |
| 3 | 155.871522 | 81.371038 |
| 4 | 135.544179 | 87.756580 |
| 5 | 101.110456 | 92.519935 |
| 6 | 66.905370 | 95.671874 |
| 7 | 47.403072 | 97.905052 |
| 8 | 11.386269 | 98.441464 |
| 9 | 7.743336 | 98.806256 |
| 10 | 7.575849 | 99.163157 |
| 11 | 3.922708 | 99.347958 |
| 12 | 3.347663 | 99.505667 |

| | | |
|---|---|---|
| 13 | 2.653064 | 99.630654 |
| 14 | 2.585955 | 99.752480 |
| 15 | 1.298401 | 99.813648 |
| 16 | 0.914461 | 99.856728 |
| 17 | 0.716957 | 99.890505 |
| 18 | 0.627985 | 99.920089 |
| 19 | 0.482533 | 99.942822 |
| 20 | 0.403027 | 99.961808 |
| 21 | 0.292856 | 99.975605 |
| 22 | 0.217012 | 99.985828 |
| 23 | 0.133460 | 99.992116 |
| 24 | 0.097064 | 99.996689 |
| 25 | 0.070292 | 100.000000 |
| 26 | 0.000000 | 100.000000 |

Graph of Percentage of Variance captured vs No. of Principal Components



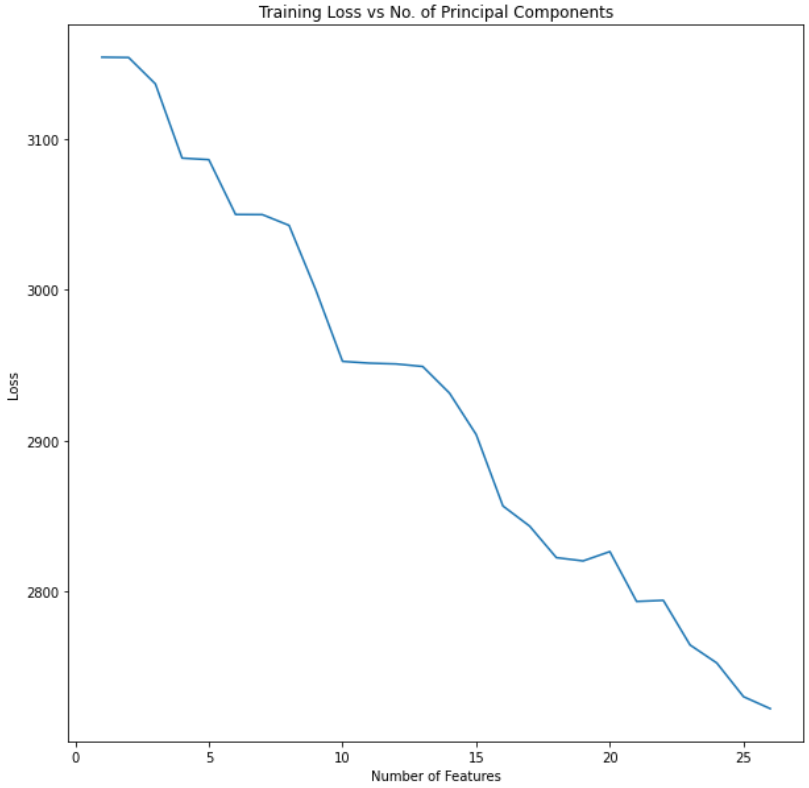
Error Table for PCA

| **Number of Principal Components** | **Percentage of Variance Captured** | **Training Error** | **Testing Error** |
|---|---|---|---|
| 1 | 54.686415 | 3154.033739 | 790.126675 |
| 2 | 74.027866 | 3153.825464 | 790.156444 |
| 3 | 81.371038 | 3136.331051 | 788.208147 |
| 4 | 87.756580 | 3087.147012 | 778.826058 |
| 5 | 92.519935 | 3086.129502 | 777.569683 |
| **6** | **95.671874** | **3049.788724** | **772.813163** |
| 7 | 97.905052 | 3049.718016 | 772.737215 |
| 8 | 98.441464 | 3042.571841 | 769.762495 |
| 9 | 98.806256 | 2999.781651 | 767.127146 |
| **10** | **99.163157** | **2952.446909** | **746.508038** |
| 11 | 99.347958 | 2951.242702 | 745.777358 |
| 12 | 99.505667 | 2950.734840 | 745.898530 |
| 13 | 99.630654 | 2949.042075 | 744.658691 |
| 14 | 99.752480 | 2931.423533 | 743.275388 |
| 15 | 99.813648 | 2903.919835 | 735.794105 |
| 16 | 99.856728 | 2856.588835 | 722.604916 |
| 17 | 99.890505 | 2843.231617 | 717.448181 |
| 18 | 99.920089 | 2822.347015 | 711.330997 |
| 19 | 99.942822 | 2820.165690 | 710.995003 |
| 20 | 99.961808 | 2826.361538 | 710.804504 |
| 21 | 99.975605 | 2793.308777 | 701.845155 |
| 22 | 99.985828 | 2794.039176 | 703.246394 |
| 23 | 99.992116 | 2764.474291 | 694.545445 |

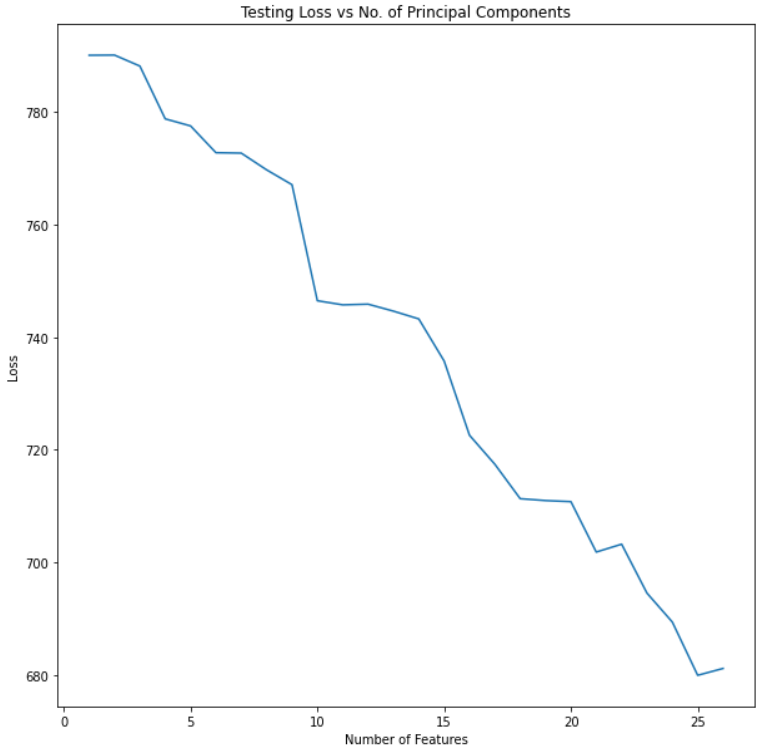
| | | | |
|---|---|---|---|
| 24 | 99.996689 | 2752.525506 | 689.406464 |
| ***25*** | ***100.000000*** | ***2730.183749*** | ***679.944818*** |
| 26 | 100.000000 | 2722.254961 | 681.162635 |

We find that the model with 6 principal components explains 95% of the variation in data, one with 10 explains 99% of the variance in the data, and that with 25 explains 100% of the variation in data.

Training Loss vs No. of Principal Components



Testing Loss vs No. of Principal Components




# Question 2B

We are now posed with the problem of selecting the features out of the 26 we have that explain the data best. Ideally, we would take every subset of the feature set and compute the training and testing errors to find which best fit our data, but due to the computational complexity of that, we employ a heuristic approach instead.
We use greedy feature forward and backward algorithms.

## Greedy Feature Forward Selection

We begin by building 26 models, each with one feature, and finding the best feature amongst all. We then build 25 models with the best feature and each of the others, continuing to add the best set of features in each iteration until we find that our model's performance is not boosted by adding a new feature.

In our feature forward selection, we get the following features selected as optimal for our predictive model.

| Feature Index | Feature Label |
|---|---|
| 20 | RH_out |

After selecting this feature, the greedy model bottomed out and did not appreciably change the testing and training errors when taking more features. Hence, it can be said that the optimal model is the one which includes only RH_out - if considering greedy feature forward selection algorithm.

Following this, we ran our gradient descent algorithm on this revised Data Frame - the RH_out column and the Appliances target variable.

No bias was included in this model (i.e w0 = 0)

The results of that are given below

| Feature Label | Coefficient |
|---|---|
| RH_out | -15.81841883 |

Hence, to distill this model into an equation

Y = Appliances
X1 = RH_out

$$Y = \ -15.81841883 \ * \ X1$$

Now, note that a necessary deviation from 2-A is a change in the loss function values from Sum of Squares to R.M.S.E to ensure ranged values. The training and testing error of this model are given below:

| **Feature Set** | **Training Error** | **Testing Error** |
|---|---|---|
| ['RH_out'] | 141.88400113694817 | 144.04237988159022 |

## Greedy Feature Backward Elimination

In this approach, we begin with a model that contains all 26 features, and iteratively eliminate the least significant feature until we observe no improvement in our model's errors on removing a feature.

In our feature backward selection, we get the following features selected as optimal for our predictive model.

| Feature Index | Feature Label |
|---|---|
| 3 | RH_2 |

| | |
|---|---|
| 11 | RH_6 |
| 13 | RH_7 |
| 15 | RH_8 |
| 16 | T9 |
| 17 | RH_9 |
| 19 | Press_mm_hg |
| 20 | RH_out |
| 22 | Visibility |

After selecting thesefeature, the greedy model bottomed out and did not appreciably change the testing and training errors when taking more features. Hence, it can be said that the optimal model is the one which includes only these 9 features - if considering greedy feature backwardselection algorithm.

Following this, we ran our gradient descent algorithm on this revised Data Frame.

No bias was included in this model (i.e w0 = 0)

The results of that are given below

| **Feature Label** | **Coefficient** |
|---|---|
| RH_2 | 2.41794018e-112 |
| RH_6 | 2.41794018e-112 |
| RH_7 | 2.41794018e-112 |
| RH_8 | 2.41794018e-112 |
| T9 | 2.41794018e-112 |
| RH_9 | 2.41794018e-112 |
| Press_mm_hg | 2.41794018e-112 |
| RH_out | -16.473459 |
| Visibility | 2.41794018e-112 |

Now, note that a necessary deviation from 2-A is a change in the loss function values from Sum of Squares to R.M.S.E to ensure ranged values. The training and testing error of this model are given below:

| **Feature Set** | **Training Error** | **Testing Error** |
|---|---|---|
| ['RH_2', 'RH_6', 'RH_7', 'RH_8', 'T9', 'RH_9', 'Press_mm_hg', 'RH_out', 'Visibility', 'Appliances'] | 114.5974826 | 114.79562959 |

# Question 2C

In 2A, we found the best model to be one that included all 26 features, that gave a sum of squares error of 2783 for the trainings set and 680 for the testing data.
In 2B, we find our greedy feature forward algorithm to give us the best model with one feature, RH_out, with training and testing errors (root mean square) of 141 and 144 respectively.
Our feature backward elimination algorithm gave us a model with 9 features, with training and testing errors of 114.5 and 114.7 respectively.

In conclusion, we find the best model to be the one obtained using the feature forward selection technique, as that has the lowest error, the least number of features, and the most generality. However, this is as far as empirical analysis can take us, and we would need our results to be supported by extensive domain knowledge to correctly ascertain the features that most influence our target variable.