**DETECTION OF MANIPULATED PRICING IN SMART ENERGY CPS SCHEDULING**

**INTRODUCTION:**

In this coursework, we are given the data of a small community with 5 people and 10 smart home appliances each. Each person is also given 10 tasks to use the smart appliances and requires them to schedule their usage. We are also given 10000 labelled training data which contains the pricing for each hour of the day and its corresponding label (normal or abnormal). Using this data, a model needs to be built to effectively classify the scheduling data into normal and abnormal (i.e., effectively predicting a pricing attack). We are also given 100 pricing curves for the grid as testing data without labels that needs to be classified using the built model. On classifying a price curve as abnormal, we need to minimize the cost by building a Linear Programming solution according to the abnormal price guideline. This minimized cost schedule will give the normal scheduling for each abnormal price curve.

**THE CLASSIFICATION MODEL:**

To classify the testing data into normal and abnormal, we need to build a model that is trained based on the given training price curves. For this purpose, various classification models that are available can be used. In our case, multiple models were tried and the model with the highest accuracy was picked. The models tried were Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Support Vector Machines (SVM) and Decision trees. Of the above-mentioned models, LDA gave the maximum accuracy and most stable and consistent results.

**LINEAR DISCRIMNANT ANALYSIS (LDA):**

Linear Discriminant Analysis uses Bayes Theorem to estimate the probability of a data point being in each available class label. The class label that gets the maximum probability will be considered the winner and given as output.

Briefly Bayes’ Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:

P(Y=x|X=x) = (PIk \* fk(x)) / sum(PIl \* fl(x))

Where PIk refers to the base probability of each class (k) observed in your training data (e.g., 0.5 for a 50-50 split in a two-class problem). In Bayes’ Theorem this is called the prior probability.

PIk = nk/n

The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminate function and the class is calculated as having the largest value will be the output classification (y):

Dk(x) = x \* (muk/siga^2) – (muk^2/(2\*sigma^2)) + ln(PIk)

Dk(x) is the discriminate function for class k given input x, the muk, sigma^2 and PIk are all estimated from your data.

**IMPLEMENTATION OF LDA:**

To implement the LDA algorithm, we use the pre-existing model from the SKLEARN Library in python. The source code for the implementation is shown below:

Text

Description automatically generated

For local testing purposes, the 10000 records from the is split into 80% training data and 20% testing data. This helps with localized random test records to test the working range of the model.

The LDA model is derived using the LinearDiscrimnantAnalysis() of the sklearn library. The fit function of the lda object is used to train the model with the labelled data that is given to it. Once the training process is done, the model can be used to predict the labels in the testing data. This is done using the predict function of the lda object which is stored in y\_pred. This is the prediction of the previously split local testing data. The accuracy of these results can be measured using the score function and the corresponding training and testing accuracy is displayed in the above image.

Now, the given un-labelled testing data is used for the prediction of abnormal and normal cases. These predicted results are stored in /TestingResults.txt and the predictions alone are stored in /PredictionsOnly.txt

**LINEAR PROGRAMMING:**

For all the price curves that were predicted to be abnormal, scheduling needs to be done using a linear programming solution. Each of the 5 users are given 10 tasks making them use the smart home appliances. This data is given by reading the COMP3217CW2input.xlsx. From the “User &Task ID” sheet then data on user and task ID, start time, deadline, max energy per hour and energy demand can be extracted. Also, the AbnormalGuidelinePricing sheet gives the prices that needs to be used during scheduling.

Text

Description automatically generated

For all abnormal curves, the corresponding user tasks and the abnormal guideline price of that hour should be used to create a minimization linear programming objective function and the constraints on which the problem should work as follows.

Text

Description automatically generated

Now, bar charts can be plotted based on the solution to depict the corrected schedule of usage according to the abnormal price guideline. These bar charts have hour as x-axis and Energy usage as y-axis. It shows the day on which abnormality has occurred and the minimized cost of that day as per the solution. The plotting of charts can be done as follows:

Text

Description automatically generated

The following are a few bar charts, and the rest are saved in /plots directory.

|  |  |
| --- | --- |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |

**PREDICTIONS:**

|  |  |
| --- | --- |
| DAY => 1 Prediction => 1  DAY => 2 Prediction => 0  DAY => 3 Prediction => 0  DAY => 4 Prediction => 0  DAY => 5 Prediction => 1  DAY => 6 Prediction => 1  DAY => 7 Prediction => 0  DAY => 8 Prediction => 1  DAY => 9 Prediction => 1  DAY => 10 Prediction => 0  DAY => 11 Prediction => 1  DAY => 12 Prediction => 0  DAY => 13 Prediction => 1  DAY => 14 Prediction => 0  DAY => 15 Prediction => 0  DAY => 16 Prediction => 1  DAY => 17 Prediction => 1  DAY => 18 Prediction => 1  DAY => 19 Prediction => 1  DAY => 20 Prediction => 1  DAY => 21 Prediction => 0  DAY => 22 Prediction => 1  DAY => 23 Prediction => 0  DAY => 24 Prediction => 0  DAY => 25 Prediction => 0  DAY => 26 Prediction => 1  DAY => 27 Prediction => 0  DAY => 28 Prediction => 1  DAY => 29 Prediction => 0  DAY => 30 Prediction => 1  DAY => 31 Prediction => 0  DAY => 32 Prediction => 1  DAY => 33 Prediction => 1  DAY => 34 Prediction => 0  DAY => 35 Prediction => 1  DAY => 36 Prediction => 0  DAY => 37 Prediction => 1  DAY => 38 Prediction => 1  DAY => 39 Prediction => 1  DAY => 40 Prediction => 1  DAY => 41 Prediction => 1  DAY => 42 Prediction => 0  DAY => 43 Prediction => 0  DAY => 44 Prediction => 1  DAY => 45 Prediction => 1  DAY => 46 Prediction => 1  DAY => 47 Prediction => 0  DAY => 48 Prediction => 1  DAY => 49 Prediction => 1  DAY => 50 Prediction => 0 | DAY => 51 Prediction => 1  DAY => 52 Prediction => 0  DAY => 53 Prediction => 1  DAY => 54 Prediction => 0  DAY => 55 Prediction => 1  DAY => 56 Prediction => 1  DAY => 57 Prediction => 1  DAY => 58 Prediction => 0  DAY => 59 Prediction => 1  DAY => 60 Prediction => 0  DAY => 61 Prediction => 0  DAY => 62 Prediction => 0  DAY => 63 Prediction => 0  DAY => 64 Prediction => 1  DAY => 65 Prediction => 1  DAY => 66 Prediction => 0  DAY => 67 Prediction => 0  DAY => 68 Prediction => 1  DAY => 69 Prediction => 1  DAY => 70 Prediction => 1  DAY => 71 Prediction => 0  DAY => 72 Prediction => 0  DAY => 73 Prediction => 0  DAY => 74 Prediction => 1  DAY => 75 Prediction => 0  DAY => 76 Prediction => 0  DAY => 77 Prediction => 1  DAY => 78 Prediction => 1  DAY => 79 Prediction => 1  DAY => 80 Prediction => 1  DAY => 81 Prediction => 0  DAY => 82 Prediction => 1  DAY => 83 Prediction => 0  DAY => 84 Prediction => 1  DAY => 85 Prediction => 1  DAY => 86 Prediction => 1  DAY => 87 Prediction => 1  DAY => 88 Prediction => 1  DAY => 89 Prediction => 1  DAY => 90 Prediction => 1  DAY => 91 Prediction => 0  DAY => 92 Prediction => 0  DAY => 93 Prediction => 0  DAY => 94 Prediction => 1  DAY => 95 Prediction => 1  DAY => 96 Prediction => 1  DAY => 97 Prediction => 0  DAY => 98 Prediction => 0  DAY => 99 Prediction => 1  DAY => 100 Prediction => 0 |