Smart Elevator by Using Support Vector Machine Regression and Machine Learning Applications

1. Abstract

Waiting for elevators take our precious time. A computer algorithm is provided to solve this modern life problem to increase punctuality of elevators through machine learning applications. Support Vector Machine(SVM) regression is applied on a dataset created to demonstrate daily usage of an elevator.

2. Introduction

An elevator is the device that is designed to transport people who live in a building from the lobby floor to his destination or vice-versa. Its properties depend on its area of usage such as speed, number of serving people, number of people in the building etc.

ELEVATOR history

Governing equations of motion

3. Support vector machine

Machine learning is an application of artificial intelligence(AI) that provides systems to automatically learn and apply its knowledge without explicitly programming. It uses data and learns how to use that data properly with getting experienced by using previous data.

SVM

4. Collecting dataset

In this application, dataset is created randomly with a manipulation data to see whether SVM is working or not. A matrix created called big data:

$$\text{Big data of elevator usage} = [\ \ ]_{mxnxk}$$

Where m is time with respect to time variable, n is number of floor, k is number of days that the data is collected. In this matrix the corresponding values are named as demand values.

i.e, in the z^{th} day of data collection, at time t, x person called the elevator from i^{th} floor in a k-floors-building.

$$\text{Big data(t,y,z)}=x+\frac{k}{2}-abs(y-i) \text{ for y=1,2,3…,k}$$

This data is created randomly for each floor as in a usual day for x=1,2,3 and i is a random time in this day. Where t is defined as:

$$t = \frac{hour * 60}{time\ variable} + round\ (\frac{minute}{time\ variable})$$

However a manipulation data created for manipulation floor 7 to explain usage of SVM on this dataset. For x=3 and random value between 7:30 AM and 7:40 AM to demonstrate a family's usual usage of the elevator. MATLAB R2016a was used to apply defined random data creation algorithm. The code is presented in appendix.

5. Application

Random dataset was created to demonstrate the possible usage of an elevator with a manipulation. Dataset named "big_data.csv" was created and inserted into Python programming via a library named "pandas". However, dataset created as 3D matrix, CSV does not support 3D data. Thus dataset converted into the desired format in Python.

```python
dataset = pd.read_csv('big_data.csv')
size = dataset.shape
days=30; #days for random data
floors=int(size[1]/days); #apartment floor number
big_data=np.array([[[0]*days]*floors]*size[0])
time_var=2880/size[0]
flr=np.array([[0]*floors])
dates = list(range(1,days+1))
for i in range(1,floors+1):
    flr[:,i-1]=np.array([eval("button"+str(i))])

for i in range(days):
    big_data[:,range(floors-1),i]=(dataset.iloc[:,range((i+1)*floors-floors,(i+1)*floors-1)].values)
```

Dataset is now ready to apply machine learning as it was desired. Another library named "sklearn" is used to apply SVM. The code for this application is presented in appendix.

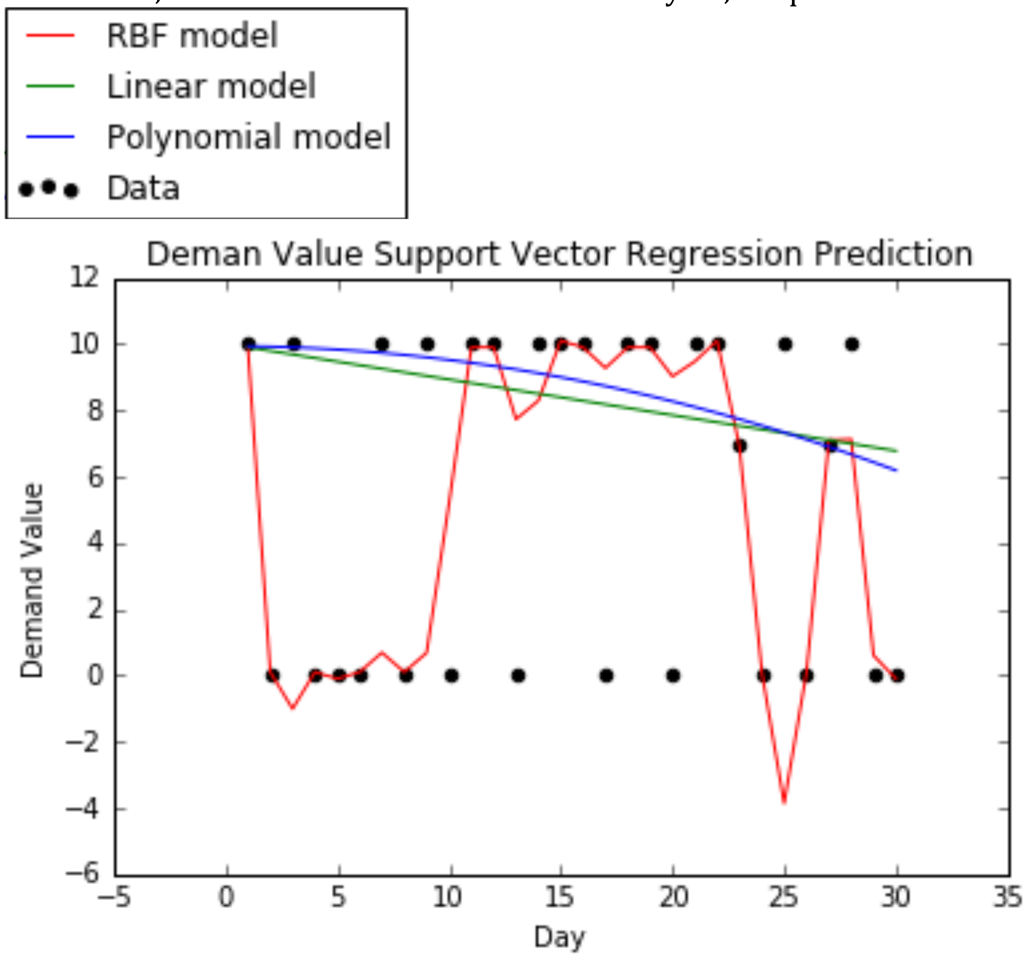
6. Results and conclusion

As it is stated in collecting dataset section, 7^{th} floor is the manipulated floor between 7:30 and 7:40. Two different random datasets are created to see the results for time variable of 5 and 10. The other conditions will be the same. 30 days of data for 13 floors apartment. "big_data5.csv" and "big_data10.csv" are represented with this report and in this part results obtained via ML on these datasets will be discussed.

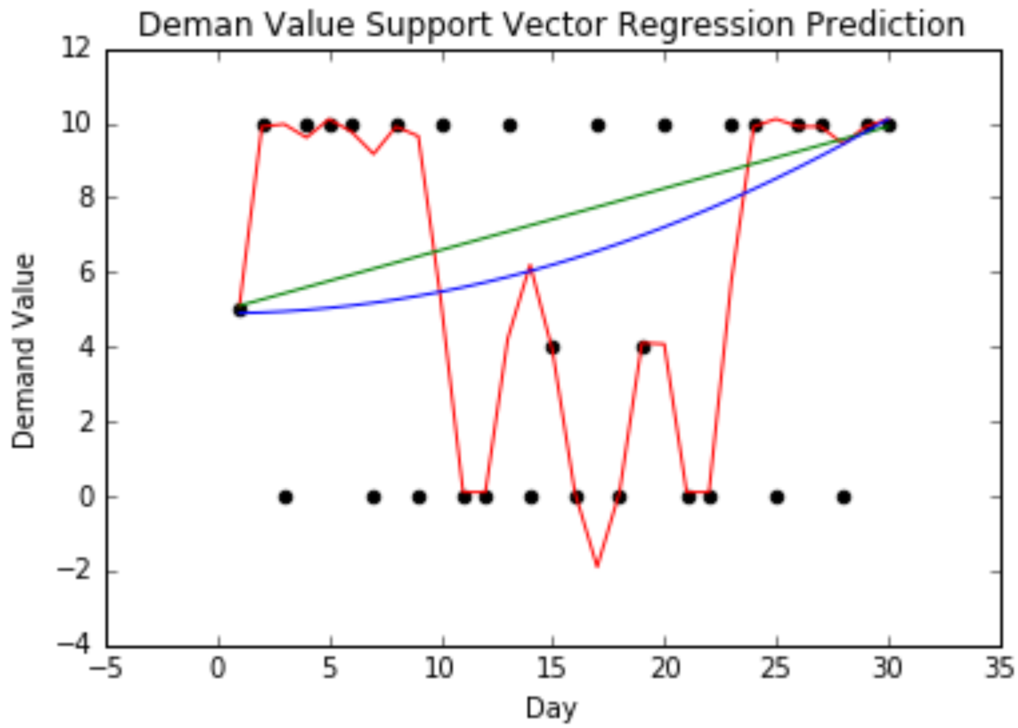
Firstly, for time variable 5, 7:30 AM is represented at t=90 and 7:40 AM is t=92. And 7:30 corresponds for t=45 and 7:40 is t=46 if time variable is 10. A MATLAB function was defined to do this conversion as:

```
function time = timeconv(hour,minute,time_var)
    time = hour*60/time_var+round(minute/time_var);
end
```

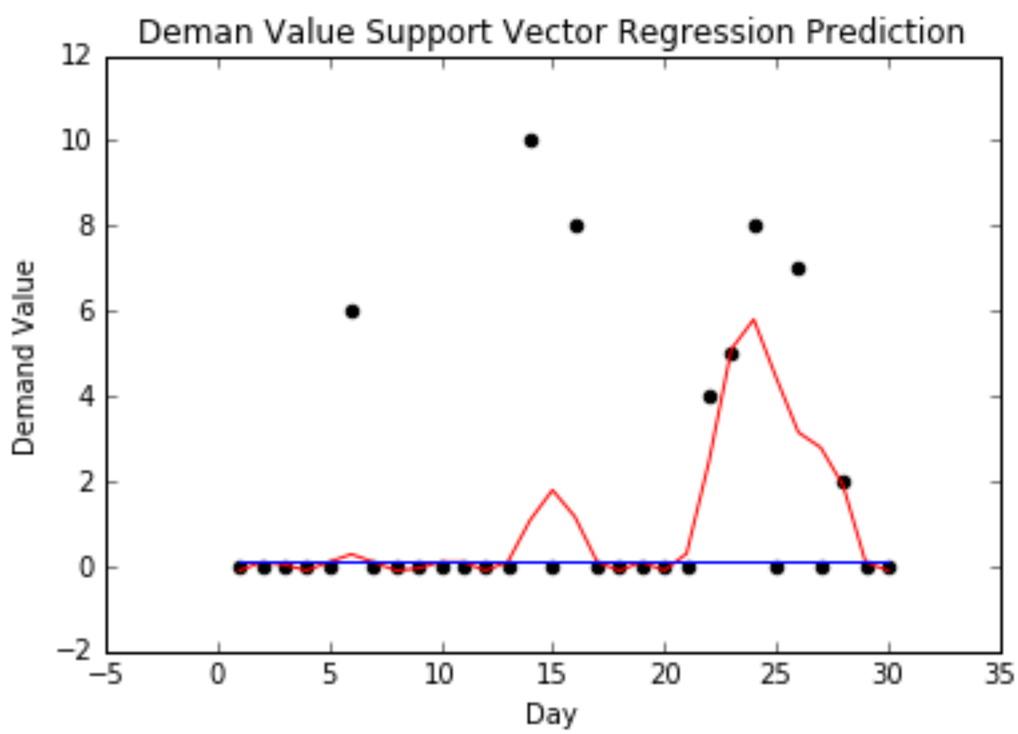
So then; for time variable of 10 minutes at day 31, the predicted data will be



The predicted demand value for 7th floor at 7:30 AM RBF kernel: 11.2899675127 Linear kernel: 6.66923076892 Polynomial kernel: 5.93333317009



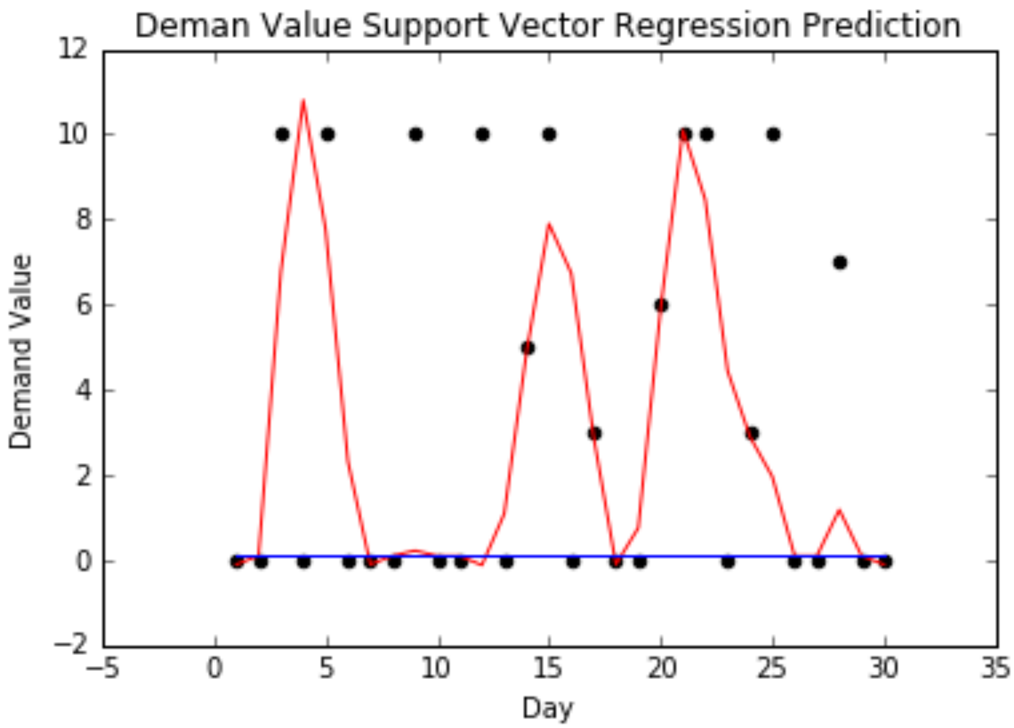
The predicted demand value for 7th floor at 7:40 AM : RBF kernel: 6.65262856087 Linear kernel: 10.0655172412 Polynomial kernel: 10.4528364756



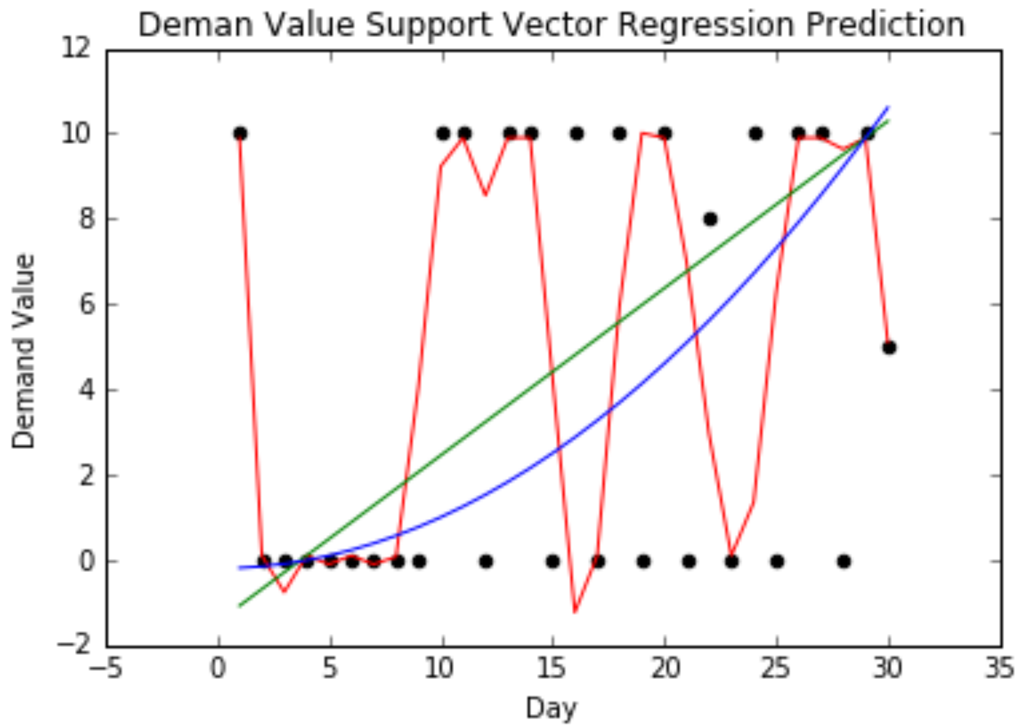
The predicted demand value for 7th floor at 7:50 AM: RBF kernel: 2.98264908699 Linear kernel: 0.0999999999889 Polynomial kernel: 0.10000014618

This dataset proves SVM on this dataset can learn regular elevator usage of 3 persons. Even though the manipulation data randomly created more usage between days 25-30 at 7:40, it predicts more demand at 7:30 for 31st day of data. RBF Kernel provide better results because for this case, this kernel also considers previous demand values with more efficient way than linear and polynomial kernels.

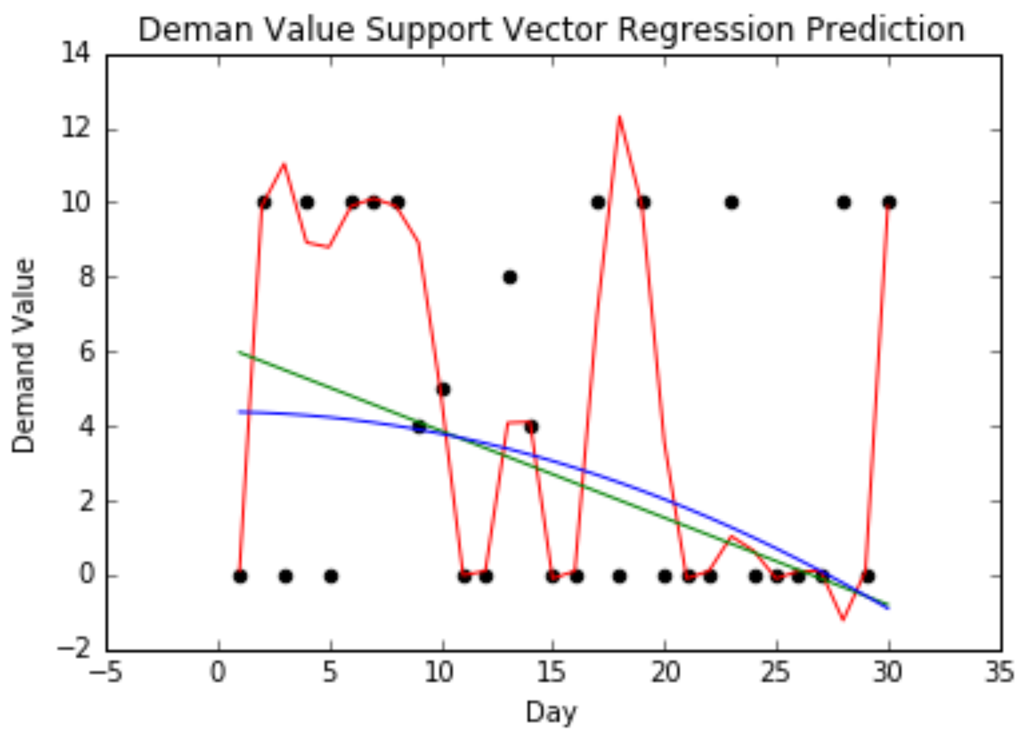
If we use dataset created for time variable of 5 minutes;



The predicted demand value for 7th floor at 7:30 AM: RBF kernel: 5.55510065927 Linear kernel: 0.0999999997999 Polynomial kernel: 0.099999895084



The predicted demand value for 7th floor at 7:35 AM: RBF kernel: -6.52852336045 Linear kernel: 10.684 Polynomial kernel: 11.341176452



The predicted demand value for 7th floor at 7:40 AM: RBF kernel: 25.7905944536 Linear kernel: -1.03333333339 Polynomial kernel: -1.26049376366

The data above shows that RBF Kernel needs sustainability to provide better results. In long term of usage, this method can provide better results. Randomization in data generating causes some errors in calculation. However, using long-term data can eliminate these errors.

7. Appendix