CS 688 Final Project

IIoT
12/10/18
Jimmy Goddard

BOSTON
UNIVERSITY

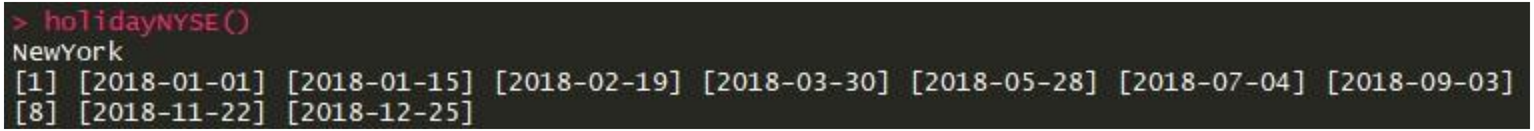
Calculate Holidays

- Very useful library, RcppBDT, which wraps the Boost C++ library’s “isHoliday” function
- Use the “getNthDayOfWeek” function from the timeDate library

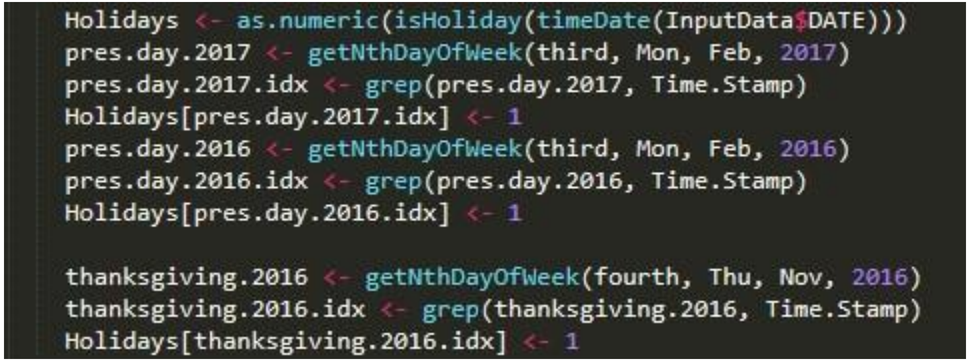
Calculate Holidays

- Encoded holidays as a 1 and non holidays as a zero
- Tried using the reverse encoding but it resulted in less accurate models
- Added President’s Day for 2016 and 2017 as well as Thanskgiving for 2017

Calculate Holidays Example



```
> holidayNYSE()
NewYork
[1] [2018-01-01] [2018-01-15] [2018-02-19] [2018-03-30] [2018-05-28] [2018-07-04] [2018-09-03]
[8] [2018-11-22] [2018-12-25]
```

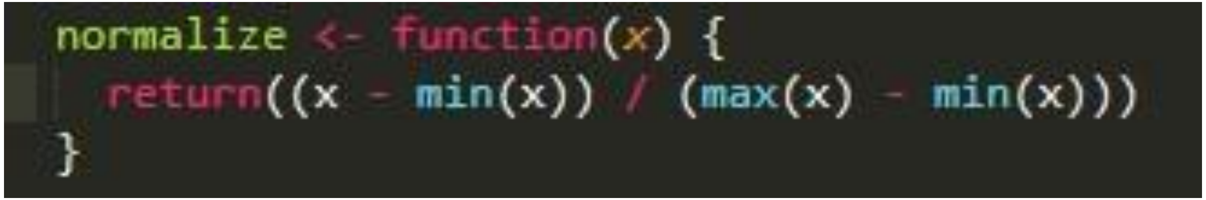


```
Holidays <- as.numeric(isHoliday(timeDate(InputData$DATE)))
pres.day.2017 <- getNthDayOfWeek(third, Mon, Feb, 2017)
pres.day.2017.idx <- grep(pres.day.2017, Time.Stamp)
Holidays[pres.day.2017.idx] <- 1
pres.day.2016 <- getNthDayOfWeek(third, Mon, Feb, 2016)
pres.day.2016.idx <- grep(pres.day.2016, Time.Stamp)
Holidays[pres.day.2016.idx] <- 1

thanksgiving.2016 <- getNthDayOfWeek(fourth, Thu, Nov, 2016)
thanksgiving.2016.idx <- grep(thanksgiving.2016, Time.Stamp)
Holidays[thanksgiving.2016.idx] <- 1
```

Unitizing inputs

- Converting each input into a number between 0 and 1:



```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}
```

- Actually reduced model accuracy

Scale Inputs

- Built-in “scale” function: scale is generic function whose default method centers and/or scales the columns of a numeric matrix.
- Gave a big boost in model accuracy
- Scaling both inputs and expected output
- This is in addition to the normalizing which was already there

Normalize Inputs

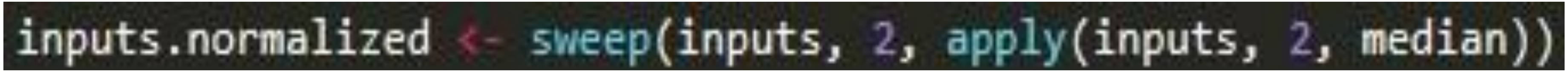
- Originally, it was dividing every number in a column by the max value of that column
- Switched it to subtract from every number in a column the mean of the column
    - This reduced model accuracy
- Applying the same transformation but with median rather than mean increased the accuracy

Normalize Targets

- Attempted similar transformations on the target values but all attempts lowered the model accuracy
- Original target scaling is to divide the target values by the max target value
  - This led to the most accurate models

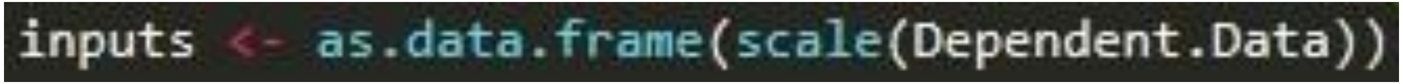
Normalize Examples

Subtract median:



```
inputs.normalized <- sweep(inputs, 2, apply(inputs, 2, median))
```

Scale input:



```
inputs <- as.data.frame(scale(Dependent.Data))
```

Neural Net Parameters

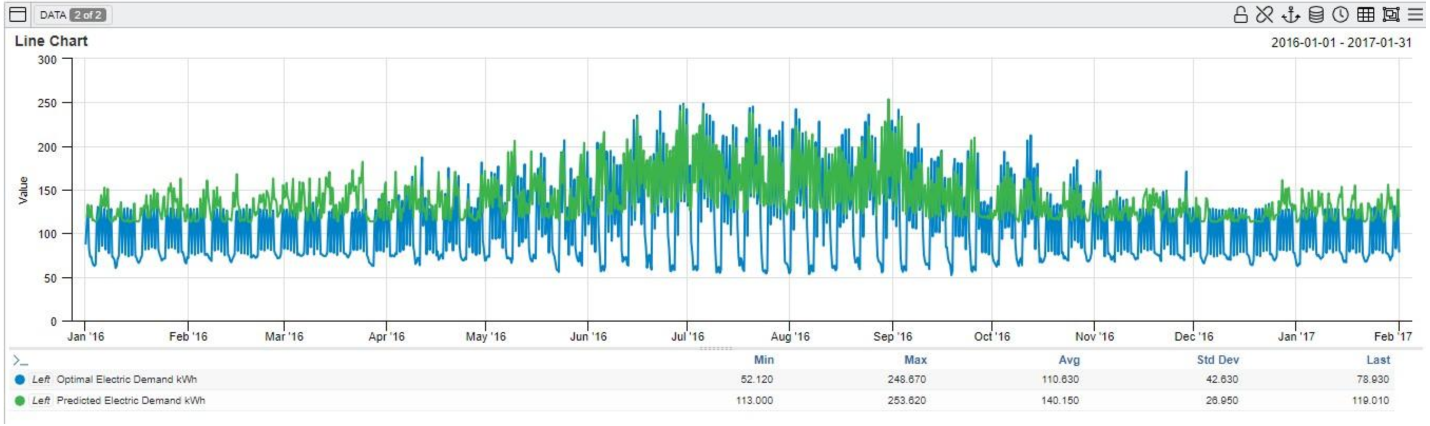
- Got better accuracy out of a NN with 6x6 hidden layer rather than original 8x8. I also tried 4x4 and 10x10, both of which lowered the accuracy
- Adjusting the learning function parameters and the maximum iterations for the model both led to less accurate models. So I kept the original values, 0.1 and 500 respectively

FacilityConneX

- Had a hard time running my code on FacilityConneX
- It would commonly show errors in the console, but it was hard to troubleshoot what was causing them
- It would finish quickly and seemed to be generating incomplete results

FacilityConneX

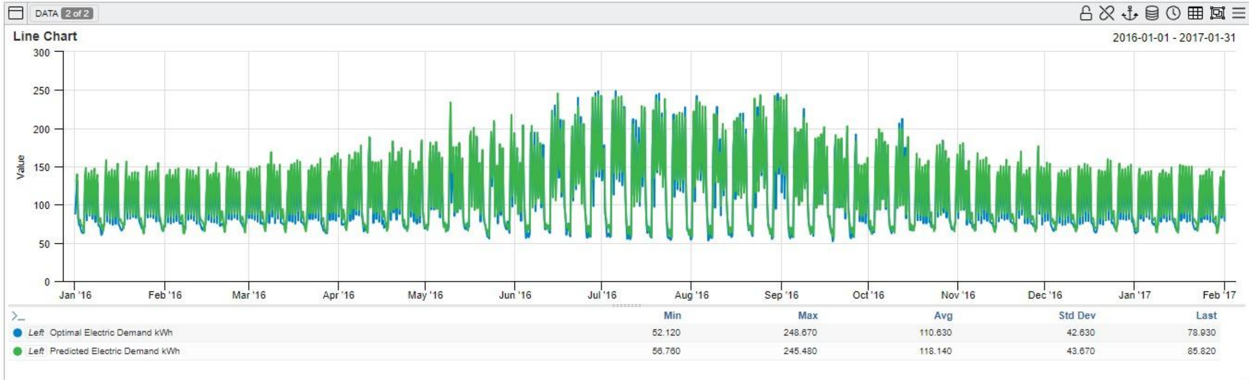
- No modifications from downloaded zip



	Min	Max	Avg	Std Dev	Last
● *Left* Optimal Electric Demand kWh	52.120	248.670	110.630	42.630	78.930
● *Left* Predicted Electric Demand kWh	113.000	253.620	140.150	26.950	119.010

FacilityConneX

- I don’t trust this because I’d get different results but this is the best looking one



	Min	Max	Avg	Std Dev	Last
● *Left* Optimal Electric Demand kWh	52.120	248.670	110.630	42.630	78.930
● *Left* Predicted Electric Demand kWh	56.760	245.480	118.140	43.670	85.820

Coefficient of Determination

- Start at: -0.05918289
- Holidays encoded as 0: 0.6543806
- Holidays encoded as 1: 0.7018767
- Subtracting mean to normalize inputs: 0.7003103
- Subtracting median: 0.7151841
- Added Thanksgiving and 2nd Pres day: 0.7237962
- 6x6 hidden layer: 0.7338013
- 40% of data set for testing: 0.7372318