**ENGR 421 HW#6**

**ANIL KUL**

**60337**

In that homework, we will develop a machine learning algorithm for real life regression problem. We have 42.958 data for training the cash withdrawals from 47 different ATMs of a bank and with our model we will predict 940 different cash withdrawals according to their given data.

*MODEL:*

For this regression problem I choose random forest classifier because:

* It is one of the most common algorithm in industry because of its high accuracy.
* It is powerful when working with large datas and high dimensionality.

*PREPROCESSING OF DATA:*

* WEEKEND
  + I added an weekend column because of the changing consumption behaviors of people in weekends.

x.insert(0,'WEEKEND',x.apply(lambda row: 0 if (pd.Timestamp(row['YEAR'],row['MONTH'],row['DAY']).weekday()<=5) else 1, axis=1))

* SPECIAL\_DATES
  + I added an special dates column and set all public holidays as special dates because of again the changing consumption behaviors of people in the holidays

x.insert(0,'SPECIAL\_DATES', x.apply(lambda row: isSpecialDate([row['YEAR'], row['MONTH'], row['DAY']]), axis=1))

* SALARY\_DATES
  + I added an salary dates column and set it first and last 5 days of month since people are tend to get money from atm more frequently.

def isSpecialDate(date):  
 specialDays=[[2018,1,1],[2018,4,23],[2018,5,1],[2018,5,19],[2018,6,14],[2018,6,15],[2018,6,16],[2018,6,17], [2018,7,15],[2018,8,20],[2018,8,21],[2018,8,22],[2018,8,23],[2018,8,24],[2018,8,30],[2018,10,29],[2019, 1, 1], [2019, 4, 23], [2019, 5, 1],[2019, 5, 19],[2019, 6, 3], [2019, 6, 4], [2019, 6, 5],[2019, 6, 6], [2019, 7, 15],[2019, 8, 10], [2019, 8, 11], [2019, 8, 12],[2019, 8, 13], [2019, 8, 14],[2019, 8, 30], [2019, 10, 29]]  
 if specialDays.\_\_contains\_\_(date): return 1  
 else: return 0

x.insert(0,'SALARY\_DAYS',x.apply(lambda row: 1 if (row['DAY']<=5 or (row['DAY']>=25 and row['DAY']<=31)) and row['WEEKEND']!=1 else 0, axis=1))

* BEFORE\_SPECIAL\_DATES
  + I added an before special dates column since people are tend to comsume more before holidays they might get money from ATMs more frequently.

def isBeforeSpecialDate(date):  
 specialDays=[[2018,12,31],[2018,12,30],[2018,4,21],[2018,4,22],[2018,4,30],[2018,4,29],[2018,5,18],  
[2018, 5, 17],[2018,6,12],[2018,6,13],[2018,7,14],[2018,7,13],[2018,8,16],[2018,8,17],[2018,8,18],[2018,8,19],[2018,8,20],[2018,8,28],[2018,8,29],[2018,10,28],[2018,10,27],[2019,12,31],[2019,12,30],[2019,4,21],[2019,4,22],[2019,4,30],[2019,4,29],[2019,5,18],[2019, 5, 17],[2019, 5, 30],[2019, 5, 31],[2019, 6, 1], [2019, 6, 2], [2019, 7, 14], [2019, 7, 13],[2019, 8, 6],[2019, 8, 7],[2019, 8, 8], [2019, 8, 9], [2019, 8, 28], [2019, 8, 29], [2019, 10, 28], [2019, 10, 29]]  
 if specialDays.\_\_contains\_\_(date): return 1 else: return 0

a

x.insert(0,'BEFORE\_SPECIAL\_DATES', x.apply(lambda row: isBeforeSpecialDate([row['YEAR'], row['MONTH'], row['DAY']]), axis=1))

* ONE HOT ENCODING FOR CATEGORICAL DATAS
  + I convert categorical data given us to keep result away from manipulative effects. (For example region 13 doesn’t better than 2 it is just a type of region)

x = pd.concat([x, pd.get\_dummies(x['IDENTITY'], prefix='IDENTITY')], axis=1)  
x = pd.concat([x, pd.get\_dummies(x['REGION'], prefix='REGION')], axis=1)  
x = pd.concat([x, pd.get\_dummies(x['TRX\_TYPE'], prefix='TRX\_TYPE')], axis=1)

* CROSS VALIDATION
  + Since we haven’t got output in test data I divede train data with the same proportion of our test/train data sizes and validate my model’s accuracy.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size= 0.025, random\_state=61)

According to that splitting of data and as a result, I found RMSE and MAE for test and training data sets.

RANDOM FOREST MODEL:

--------------------

RMSE for test: 13.589668246649868

MAE for test: 8.069646182495344

RMSE for train: 5.489972106079196

MAE for train: 3.0711025690000957