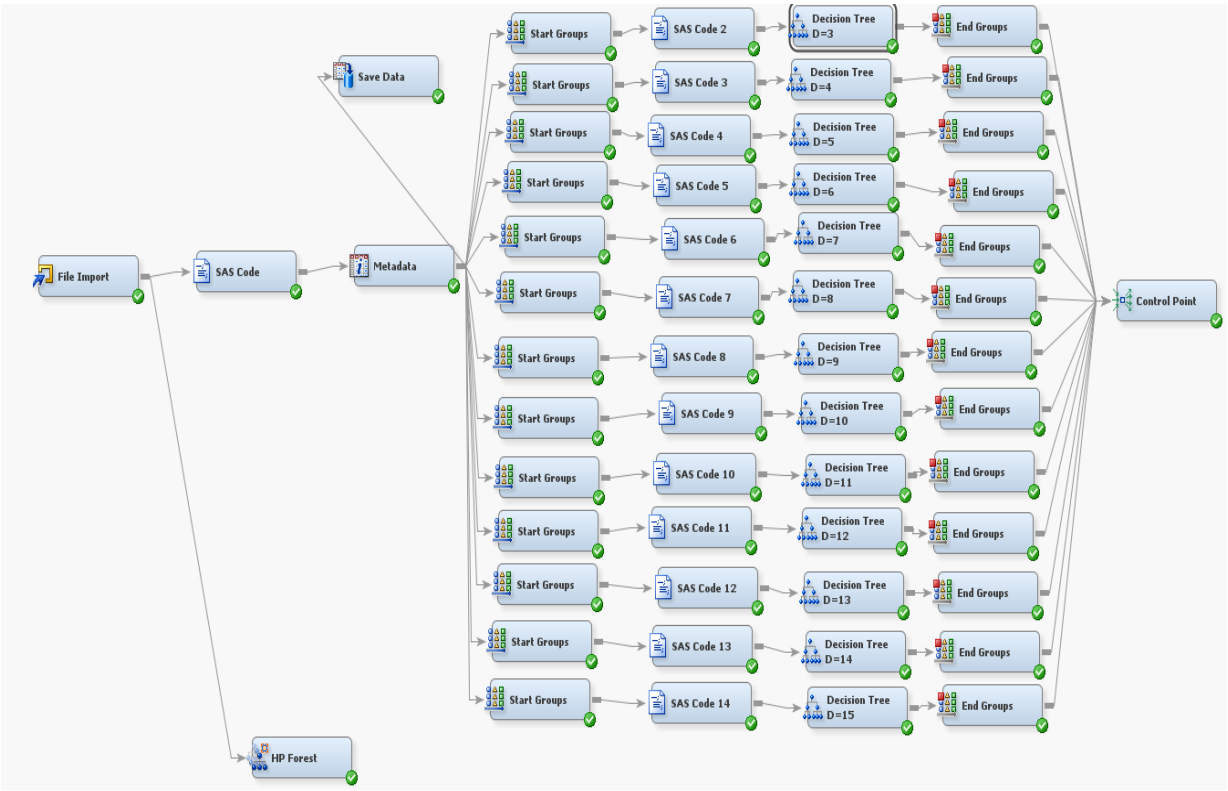
STAT 656
Homework 6

Name-Mayank Jaggi

UIN-526005299

PART 1 SAS EM

1) Project Diagram



2)

SAS Code 1

Training Code

```
data mylib.selection;
call streaminit(12345);
set &em_import_data;
urand = rand('uniform');
proc sort data=mylib.selection;
by urand;
data &em_export_train;
drop fold_size urand;
set mylib.selection NOBS=nobs_;
fold_size = round(nobs_ /4.0);
if _N_ <= fold_size then fold='A';
if _N_ > fold_size and _N_ <=2*fold_size then fold='B';
if _N_ > 2*fold_size and _N_ <=3*fold_size then fold='C';
if _N_ > 3*fold_size then fold='D';
proc means data=&em_export_train;
by fold;
var Log_Cum_Production;
run;
```

SAS Code 2

Training Code

```
data mylib.templ;
retain c1 c2 c3 c4 0;
keep c1 c2 c3 c4;
set &em_import_data end=eof;
if fold='A' then c1 = c1 + 1;
if fold='B' then c2 = c2 + 1;
if fold='C' then c3 = c3 + 1;
if fold='D' then c4 = c4 + 1;
if eof then output;
data &em_export_validate;
drop c1 c2 c3 c4 rfold;
retain rfold '0';
set mylib.AllData_Train;
if rfold ='0' then do;
set mylib.templ;
if c1=0 then rfold='A';
if c2=0 then rfold='B';
if c3=0 then rfold='C';
if c4=0 then rfold='D';
end;
if fold= rfold then output;
run;
```

Note: Couldn't complete remaining sections of Part 1

PART 2 PYTHON

##1) PYTHON CODE

```
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 27 12:11:59 2019

@author: mayank
"""

from AdvancedAnalytics import ReplaceImputeEncode
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_validate
import pandas as pd
import numpy as np
import math


df2 = pd.read_excel("OilProduction.xlsx")     #data file name

# Nominal Data to string
df2['Operator'] = df2['Operator'].astype(str)
df2['County'] = df2['County'].astype(str)



attribute_map = {
        'Log_Cum_Production':['I',(8,15)],
        'Log_Proppant_LB':['I',(6,18)],
        'Log_Carbonate':['I',(-4,4)],
        'Log_Frac_Fluid_GL':['I',(7,18)],
        'Log_GrossPerforatedInterval':['I',(4,9)],
        'Log_LowerPerforation_xy':['I',(8,10)],
        'Log_UpperPerforation_xy':['I',(8,10)],
        'Log_TotalDepth':['I',(8,10)],
        'N_Stages':['I',(2,14)],
        'X_Well':['I',(-100,-95)],
        'Y_Well':['I',(30,35)],

'Operator':['N',('1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','
16','17','18','19','20','21','22','23','24','25','26','27','28')],

'County':['N',('1','2','3','4','5','6','7','8','9','10','11','12','13','14')]

 }
```

```
# Data Preprocessing, Replace outlier, impute missing values and encode
rie = ReplaceImputeEncode(data_map=attribute_map,
nominal_encoding='one-hot',interval_scale=None, drop=False, display=True)
# Now request replace-impute-encode for your dataframe
encoded_df = rie.fit_transform(df2)
print("\nData after replacing outliers, impute missing and encoding:")
print(encoded_df.head())


# Defining target and input variables

y = encoded_df['Log_Cum_Production']   #target
x = encoded_df.drop('Log_Cum_Production',axis=1)  #input
np_y=np.ravel(y)     # convertig  y into flat array


# 4 fold Cross validation, Random Forest
score_names = ['MSE']
max_depth = [3,4,5,6,7,8,9,10,11,12,13,14,15]
score_list = ['neg_mean_squared_error']
min_mse = 1e64
print("\n***************Random Forest************")
for d in max_depth:
    print("\nDepth = ",d)
    rf = RandomForestRegressor(n_estimators = 10, criterion='mse', max_depth= d,
max_features='auto', random_state=12345)
    scores = cross_validate(rf, x, np_y,
scoring=score_list,return_train_score=False, cv=4)
    print("{:.<13s}{:>6s}{:>13s}".format("Metric", "Mean", "Std. Dev."))
    i=0
    for s in score_list:
        var = "test_"+s
        mean = math.fabs(scores[var].mean())
        std = scores[var].std()
        label = score_names[i]
        i += 1
        print("{:.<13s}{:>7.4f}{:>10.4f}".format(label, mean, std))
        if label == 'MSE' and mean < min_mse:
            min_mse = mean
            best_depth_rf = d

print("Best Depth using Random Forest = ", best_depth_rf)



# 4 fold Cross validation, Decision Tree

print("\n***************Decision Tree************")
```

```
best_depth_dt=0
min_mse_dt = 1e64
for d in max_depth:
    print("\nDepth = ",d)
    dtr = DecisionTreeRegressor(max_depth= d, max_features='auto',
random_state=12345)
    scores_dt = cross_validate(dtr, x, np_y,
scoring=score_list,return_train_score=False, cv=4)
    print("{:.<13s}{:>6s}{:>13s}".format("Metric", "Mean", "Std. Dev."))
    i=0
    for s in score_list:
        var = "test_"+s
        mean_dt = math.fabs(scores_dt[var].mean())
        std_dt = scores_dt[var].std()
        label_dt = score_names[i]
        i += 1
        print("{:.<13s}{:>7.4f}{:>10.4f}".format(label_dt, mean_dt, std_dt))
        if label_dt == 'MSE' and mean_dt < min_mse_dt:
            min_mse_dt = mean_dt
            best_depth_dt = d


print("Best Depth using Decision Tree = ", best_depth_dt)
```

```
## 2)
METRICS FOR RANDOM FOREST

****************Random Forest************

Depth =  3
Metric.......  Mean    Std. Dev.
MSE.......... 0.2793    0.0180

Depth =  4
Metric.......  Mean    Std. Dev.
MSE.......... 0.2636    0.0168

Depth =  5
Metric.......  Mean    Std. Dev.
MSE.......... 0.2510    0.0156

Depth =  6
Metric.......  Mean    Std. Dev.
MSE.......... 0.2445    0.0140
```

```
Depth =  7
Metric.......  Mean    Std. Dev.
MSE.......... 0.2380    0.0154

Depth =  8
Metric.......  Mean    Std. Dev.
MSE.......... 0.2340    0.0154

Depth =  9
Metric.......  Mean    Std. Dev.
MSE.......... 0.2310    0.0160

Depth =  10
Metric.......  Mean    Std. Dev.
MSE.......... 0.2307    0.0166

Depth =  11
Metric.......  Mean    Std. Dev.
MSE.......... 0.2303    0.0161

Depth =  12
Metric.......  Mean    Std. Dev.
MSE.......... 0.2305    0.0171

Depth =  13
Metric.......  Mean    Std. Dev.
MSE.......... 0.2312    0.0169

Depth =  14
Metric.......  Mean    Std. Dev.
MSE.......... 0.2310    0.0155

Depth =  15
Metric.......  Mean    Std. Dev.
MSE.......... 0.2318    0.0175
Best Depth using Random Forest =  11




METRICS FOR DECISION TREE

**********Decision Tree***********

Depth =  3
Metric.......  Mean    Std. Dev.
MSE.......... 0.3031    0.0185
```

```
Depth =  4
Metric.......  Mean    Std. Dev.
MSE.......... 0.2879    0.0161

Depth =  5
Metric.......  Mean    Std. Dev.
MSE.......... 0.2785    0.0202

Depth =  6
Metric.......  Mean    Std. Dev.
MSE.......... 0.2775    0.0190

Depth =  7
Metric.......  Mean    Std. Dev.
MSE.......... 0.2809    0.0154

Depth =  8
Metric.......  Mean    Std. Dev.
MSE.......... 0.2958    0.0191

Depth =  9
Metric.......  Mean    Std. Dev.
MSE.......... 0.3107    0.0161

Depth =  10
Metric.......  Mean    Std. Dev.
MSE.......... 0.3282    0.0300

Depth =  11
Metric.......  Mean    Std. Dev.
MSE.......... 0.3469    0.0255

Depth =  12
Metric.......  Mean    Std. Dev.
MSE.......... 0.3564    0.0339

Depth =  13
Metric.......  Mean    Std. Dev.
MSE.......... 0.3677    0.0259

Depth =  14
Metric.......  Mean    Std. Dev.
MSE.......... 0.3891    0.0236

Depth =  15
Metric.......  Mean    Std. Dev.
MSE.......... 0.3873    0.0238
Best Depth using Decision Tree =  6
```

##3) MODEL SELECTION

In case of Random Forest, we selected a model with depth=11 and for decision trees we selected a model with depth=6 since these models have the lowest MSE.

##4) COMPARISON BETWEEN DECISION TREE AND RANDOM FORREST MODEL

For Random forest, best model has:
depth=11
MSE=0.2303

For Decision tree, best model has:
depth=6
MSE=0.2775

Thus, using random forest gives a better solution as its MSE is lower than that of decision tree model.