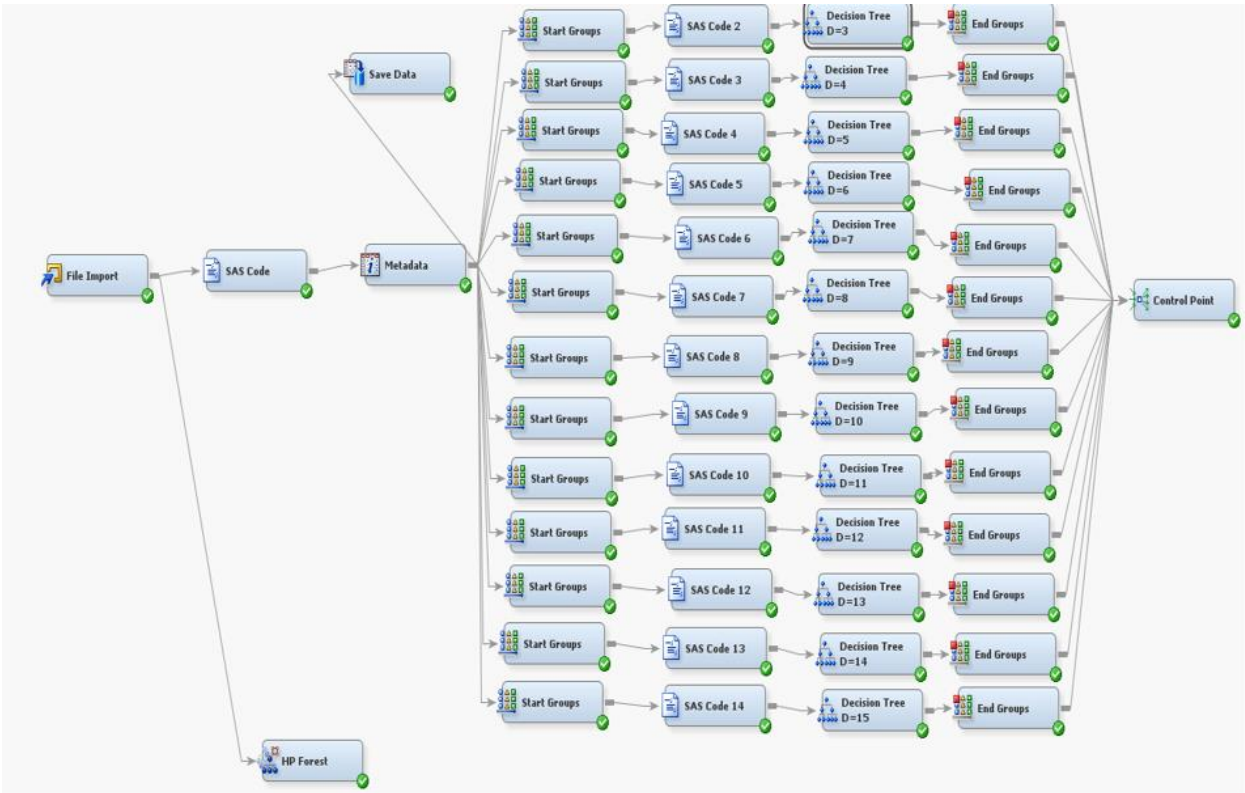
SAS Part

1. A SCREEN SHOT OF YOUR PROJECT DIAGRAM



2. A SCREEN SHOT OR LISTING OF ALL SAS CODE USED IN YOUR DIAGRAM

```
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

```
data mylib.temp1;
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.temp1;
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;
```

3. A table of the metrics for each of the decision tree cross-validation folds.

Depth	3	4	5	6	7	8	9	10	11	12	13	14	15
ASE-Fold A	0.321	0.300	0.286	0.274	0.269	0.265	0.264	0.264	0.264	0.264	0.264	0.264	0.264
ASE-Fold B	0.286	0.266	0.262	0.255	0.250	0.249	0.249	0.247	0.247	0.247	0.247	0.247	0.247
ASE-Fold C	0.291	0.270	0.263	0.251	0.248	0.248	0.248	0.246	0.246	0.246	0.246	0.246	0.246
ASE-Fold C	0.300	0.277	0.270	0.269	0.264	0.263	0.261	0.261	0.261	0.261	0.261	0.261	0.261
Average	0.300	0.278	0.270	0.262	0.258	0.256	0.256	0.255	0.255	0.255	0.255	0.255	0.255
Minimum	0.255	Q6											

4. Describe which decision tree you selected and why.

The tree with depth=10. It is the minimum depth with the minimum lowest Validation Average Square Error.

5. Compare the best decision tree to the random forest solution.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Log Cu...	Log Cu...	ASE	Average ...	0.221755	.	.
Log Cu...	Log Cu...	DIV	Divisor f...	4752	.	.
Log Cu...	Log Cu...	MAX	Maximu...	3.384702	.	.
Log Cu...	Log Cu...	NOBS	Sum of ...	4752	.	.
Log Cu...	Log Cu...	RASE	Root Av...	0.470909	.	.
Log Cu...	Log Cu...	SSE	Sum of ...	1053.781	.	.

Average Square Error for Random forest= 0.221755 vs a decision tree where lowest ASE was 0.254626. So, random forest performs better

Python Part

1. A listing of your python code.

```
from AdvancedAnalytics import DecisionTree
from sklearn.ensemble import RandomForestRegressor
#other needed classes
from AdvancedAnalytics import ReplaceImputeEncode
from sklearn.model_selection import cross_validate
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
import math

df=pd.read_excel(r"C:\Users\haris\Documents\stat
656\week6\OilProduction(1).xlsx")

df['Operator'] = df['Operator'].astype(str)
df['County'] = df['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')]
        }


rie = ReplaceImputeEncode(data_map=attribute_map, interval_scale='std', \
                           nominal_encoding='one-hot', drop=False,
display=True)
encoded_df = rie.fit_transform(df)
varlist = ['Log_Cum_Production']
X = encoded_df.drop(varlist, axis=1)
y = encoded_df[varlist]
np_y = np.ravel(y) #convert dataframe column to flat array
col = rie.col
col.remove('Log_Cum_Production')

print("\n******** RANDOM FOREST ********")
max_depth_list = [3,4,5,6,7,8,9,10,11,12,13,14,15]
n_trees_list = [10, 50, 100]
score_list = ['neg_mean_squared_error', 'neg_mean_absolute_error']
score_names = ['MSE', 'MAE']
min_mse = 1e64
for n_trees in n_trees_list:
    for d in max_depth_list:
        print("\nNumber of Trees: ", n_trees, " Max_Depth: ", d)
        rfr = RandomForestRegressor(n_estimators=n_trees, criterion='mse',
max_depth=d, \
                                    max_features='auto', min_samples_split=2,
n_jobs=1, \
                                    random_state=12345)
        scores = cross_validate(rfr, X, np_y, scoring=score_list,
return_train_score=False, cv=5)
        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 = d
                best_n_trees = n_trees


print("\nBest based on MSE from a forest with ", best_n_trees, " trees.")
print("Best Depth (trees) = ", best_depth)

#Decision Tree
from sklearn.tree import DecisionTreeRegressor
print("\n***************Decision Tree************")
best_depth_dt=0
min_mse_dt = 1e64
for d in max_depth_list:
    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 = mean
            best_depth_dt = d
```

2. A table of the metrics calculated for each of your 4 cross-validation folds.

```
******** RANDOM FOREST ********

Number of Trees:  10  Max_Depth:  3
Metric.......  Mean    Std. Dev.
MSE.......... 0.5854    0.0489
MAE.......... 0.5809    0.0266

Number of Trees:  10  Max_Depth:  4
Metric.......  Mean    Std. Dev.
MSE.......... 0.5515    0.0497
MAE.......... 0.5630    0.0279

Number of Trees:  10  Max_Depth:  5
Metric.......  Mean    Std. Dev.
MSE.......... 0.5238    0.0486
MAE.......... 0.5473    0.0260

Number of Trees:  10  Max_Depth:  6
Metric.......  Mean    Std. Dev.
MSE.......... 0.5095    0.0486
MAE.......... 0.5384    0.0251

Number of Trees:  10  Max_Depth:  7
Metric.......  Mean    Std. Dev.
MSE.......... 0.5017    0.0478
MAE.......... 0.5322    0.0249

Number of Trees:  10  Max_Depth:  8
Metric.......  Mean    Std. Dev.
MSE.......... 0.4909    0.0435
MAE.......... 0.5255    0.0215

Number of Trees:  10  Max_Depth:  9
Metric.......  Mean    Std. Dev.
MSE.......... 0.4862    0.0438
MAE.......... 0.5224    0.0220

Number of Trees:  10  Max_Depth:  10
Metric.......  Mean    Std. Dev.
MSE.......... 0.4863    0.0479
MAE.......... 0.5213    0.0232

Number of Trees:  10  Max_Depth:  11
Metric.......  Mean    Std. Dev.
MSE.......... 0.4888    0.0458
MAE.......... 0.5216    0.0229

Number of Trees:  10  Max_Depth:  12
Metric.......  Mean    Std. Dev.
MSE.......... 0.4834    0.0435
MAE.......... 0.5177    0.0219

Number of Trees:  10  Max_Depth:  13
Metric.......  Mean    Std. Dev.
MSE.......... 0.4866    0.0483
MAE.......... 0.5193    0.0237

Number of Trees:  10  Max_Depth:  14
Metric.......  Mean    Std. Dev.
MSE.......... 0.4872    0.0441
MAE.......... 0.5199    0.0216

Number of Trees:  10  Max_Depth:  15
Metric.......  Mean    Std. Dev.
MSE.......... 0.4874    0.0472
MAE.......... 0.5200    0.0223

Best based on MSE from a forest with  10  trees.
Best Depth (trees) =  12
```

3. Describe which model you selected from Cross-Validation, and why

Chose the model based on MSE = depth = 12; MSE = 0.4834

4. Compare the best decision tree solution to the random forest solution.

```
****************Decision Tree************

Depth =  3
Metric.......  Mean    Std. Dev.
MSE.......... 0.6365    0.0389
MAE.......... 0.6112    0.0166

Depth =  4
Metric.......  Mean    Std. Dev.
MSE.......... 0.6046    0.0338
MAE.......... 0.5923    0.0140

Depth =  5
Metric.......  Mean    Std. Dev.
MSE.......... 0.5848    0.0425
MAE.......... 0.5773    0.0162

Depth =  6
Metric.......  Mean    Std. Dev.
MSE.......... 0.5827    0.0397
MAE.......... 0.5742    0.0185

Depth =  7
Metric.......  Mean    Std. Dev.
MSE.......... 0.5853    0.0371
MAE.......... 0.5699    0.0177

Depth =  8
Metric.......  Mean    Std. Dev.
MSE.......... 0.6202    0.0407
MAE.......... 0.5837    0.0185

Depth =  9
Metric.......  Mean    Std. Dev.
MSE.......... 0.6522    0.0337
MAE.......... 0.5895    0.0175

Depth =  10
Metric.......  Mean    Std. Dev.
MSE.......... 0.6781    0.0544
MAE.......... 0.6012    0.0206

Depth =  11
Metric.......  Mean    Std. Dev.
MSE.......... 0.7161    0.0631
MAE.......... 0.6117    0.0204

Depth =  12
Metric.......  Mean    Std. Dev.
MSE.......... 0.7629    0.0584
MAE.......... 0.6284    0.0154

Depth =  13
Metric.......  Mean    Std. Dev.
MSE.......... 0.7812    0.0602
MAE.......... 0.6418    0.0155

Depth =  14
Metric.......  Mean    Std. Dev.
MSE.......... 0.7843    0.0421
MAE.......... 0.6436    0.0135

Depth =  15
Metric.......  Mean    Std. Dev.
MSE.......... 0.8295    0.0653
MAE.......... 0.6574    0.0147
```

The model with depth = 6 is the best based on MSE = 0.5827.

This is clearly more than MSE of random forest's best. Hence, random forest turns out to give better results