Assignment 3
APA track and TESU Masters (R/Python users)

Q1. [2 points] Calculate the lift by quintile (1/5th) for each of the 2 partisanship models that you created in the lesson 2 exercises.

df_voter_ID_modelscore_log_reg (logistic regression)

```
First Quartile:1.00
Second Quartile:1.00
Third Quartile:1.00
100th Percentile:1.00
1st Percentile:1.00

See attached voter id files and python code.
```

df_voter_ID_modelscore_decison_tree_df (descison tree)

```
First Quartile:0.00
Second Quartile:1.00
Third Quartile:1.00
100th Percentile:1.00
1st Percentile:0.00
```

See attached voter id files and python code.

df_voter_ID_modelscore_log_reg.csv

df_voter_ID_modelscore_decison_tree

quintile for partisanship model_decison_tree.ipynb

quintile for partisanship model _log_reg.ipynb

Assignment 3
APA track and TESU Masters (R/Python users)

- `Series.quartile()` function returns the specific value of a quant
- Here is a table that summarizes various quantiles:

Value of `q`	Quantile
0.05	1st quintile
0.1	1st Decile/2nd quintile
0.2	2nd Decile/4th quintile
0.25	1st quarter/5th quintile/ 25th percentile
0.3	3rd Decile/6th quintile/ 30th percentile
0.4	4th Decile/8th quintile/ 40th percentile
0.5	1st half/2nd quarter/5th Decile/10th quintile/50th percentile
0.6	6th Decile/12th quintile/60th percentile
0.7	7th Decile/14th quintile/70th percentile
0.75	3rd quarter/15th quintile/ 75th percentile
0.9	9th Decile/18th quintile/90th percentile
1.0	10th Decile/20th quintile/100th percentile

Q2. [2 points] Combine the two partisanship models made in lesson 2 to create an ensemble model predicting partisanship.

See attached python code

ensemble model predicting partisanship.ipynb

Q3. [2 points] Calculate the quintile lift for the combined partisanship model.

See attached python code

ensemble model predicting partisanship.ipynb

Q4. [3 points] Build one or more models predicting candidate support, rather than partisanship.

See attached python code

Assignment 3
APA track and TESU Masters (R/Python users)

Q4. [3 points] Build one or more models predicting candidate support, rather than partisanship.ipynb

q4_df_voter_ID_modelscore_log_reg.csv file

Q5. [3 points] Build a model predicting the overall persuadability of voters in FX

voter changed their mind in some way between the first and second waves of IDs.

See attached python code

Q5. [3 points] Build a model predicting the overall persuadability of voters in FX (wave 1 to wave 2)

q5_df_voter_ID_modelscore_log_reg.csv file

Q6. [4 points] Build two uplift models predicting how likely it is that a voter will become more likely to support the Democratic candidate based on the test mailings for message A and message B.

See attached python code

Q6. [4 points] Build two uplift models.ipynb

q6_df_voter_ID_modelscore_log_reg.csv

Q7. [4 points]

a. **Perform the lift calculation from question 1 for the models built in last lesson using the small and full datasets**

   See attached python code

   Q7_df_voter_ID_modelscore_log_reg_quintile (scores for Messge_A & Message_B).ipynb

b. **How does the lift differ in models built using the two different datasets?**

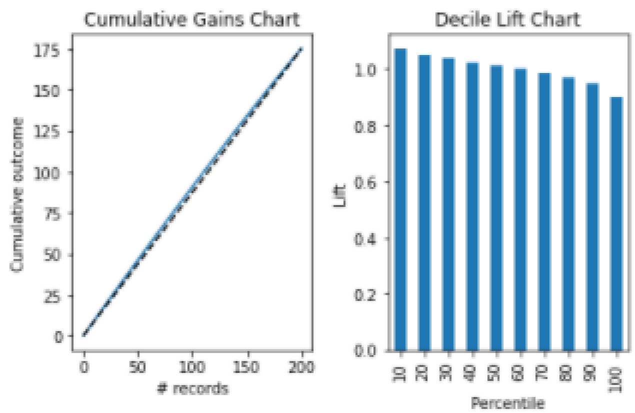
Assignment 3
APA track and TESU Masters (R/Python users)

Computation time is much more when running data with large dataset as opposed to small dataset and it crashed the processor few times with Catboost uplift model for large data set.

<u>Decile lift and gains chart for small dataset (with Catboost uplift model)</u>

```
# code for generating cumulative gains and decile for Uplift_Catboost
|

pred_v = pd.Series(ct_model.predict(X_test))
pred_v = pred_v.sort_values(ascending=False)
fig, axes = plt.subplots(nrows=1, ncols=2)
ax = gainsChart(pred_v, ax=axes[0])
ax.set_ylabel('Cumulative outcome')
ax.set_title('Cumulative Gains Chart')
ax = liftChart(pred_v, ax=axes[1], labelBars=False)
ax.set_ylabel('Lift')
plt.tight_layout()
plt.show()
```



Assignment 3

APA track and TESU Masters (R/Python users)