**Full Scenario Process Flow**

1. Regex Clean both **Train** and **Test** Emails:
   1. Input:
      1. Data path for .csv
   2. Output:
      1. Panda’s data frame of cleaned email data
2. Build TF-IDF matrix for **Train** data:
   1. Input:
      1. Panda’s data frame of cleaned **train** data
   2. Output:
      1. Vectorizer of **train** feature names
      2. TF-IDF matrix of **train** data, saved copy of TF-IDF matrix (.npy)
   3. Emails split by scenario
3. Build TF-IDF matrix for **Test** data:
   1. Input:
      1. Panda’s data frame of cleaned **test** data
      2. vectorizer of **train** feature names
   2. Output:
      1. TF-IDF matrix of **test** data
      2. saved copy of TF-IDF matrix (.npy)
   3. Emails split by scenario
4. Build LSA Matrix for **Train** and **Test** Data:
   1. Input:
      1. TF-IDF matrix of **train** data
      2. TF-IDF matrix of **test** data
   2. Output:
      1. saved copy of **train** LSA (.npy)
      2. saved copy of **test** LSA (.npy)
      3. saved copy of truncated svd (.npy)
   3. Process Flow:
      1. Fit Truncated SVD object on **train** data
      2. Transform fitted svd on **train** data
      3. Transform same fitted svd on **test** data
5. Combine cleaned email data frame and LSA matrix for both **Train** and **Test:**
   1. Input:
      1. LSA matrix data path (.npy)
      2. panda’s data frame of cleaned emails
   2. Output:
      1. panda’s data frame of combined LSA Matrix and cleaned emails
   3. Emails split by scenario
6. Train Tree on **Train** data:
   1. Input:
      1. panda’s data frame of combined LSA and cleaned email data for **train** data
   2. Output:
      1. pickled version of trained tree
7. Evaluate tree on **Test** data:
   1. Input:
      1. data path of pickled trained tree
      2. panda’s data frame of LSA and cleaned email data for **test** data
   2. Output:
      1. Evaluation Statistics for full scenario Run