**Working Sessions Notes**

# Jan 22 – kickoff meeting with WB, and follow-up emails

* Met with the full team with self introductions
* WB introduced possible scope of work and tasks, including
  + Analysis of "Trade costs and volume of trade in agriculture and fertilizer products in Africa."
  + Improve the WB transport model – FlowMax (transport demand, route choice analysis, traffic pattern, etc.)
  + Development of a country scoring index to assess a country’s potential to produce sustainable aviation fuel.

# Jan 23 – First Class

* Agreed with professor to clarify with WB and identify the problem statement and research plans ASAP
* Jenny will create a Github
* Jichong to document the working session notes

# Feb 6 – 3rd Class

* Jenny to send Github link to professor
* Meeting scheduling (**starting next week – week of Feb 12**)
  + Setup a bi-weekly meeting withDr. Gupta to talk about progress
    - invite Prof. Jafari (Weds or Thursdays, 6-7pm)
  + Setup a weekly meeting for Jenny, Jichong, professor Jafari
  + Jenny should come to the class every 3 weeks
* Asking Dr. Gupta for more recent data (now data ends in 2020); the more recent the better
* Approach suggestions (from professor Jafari)
  + **Create modular functions to pre-process** the data (can be named Preprocessor), like
    - Normalization
    - Standardization
    - Find nulls (give datasets and return df)
    - Imputation methods
    - Categorical Encoding
  + In the Github repo, create utlilities.py, and use all the modular functions
  + In the code scripts:
    - Use Main.py;
      * e.g. from utilities import normalization
    - Create a class of Preprocessor
      * Put these methods as functions
  + PyCaret – can also do this; can be used to compare with our modulars
  + **Create modular functions for models**
    - SVN, decision tree, XGBoost, and CatBoost
    - Write a class of these models, to bring any datasets
    - Write a class/functions to train and fit the models for any datasets
  + **Create modular functions for displaying a table of results**
    - First week with initial data will be the benchmarks
    - Check benchmark results with papers Dr. Gupta has
  + **For improvements** 
    - Crate a package for feature selection, and feature engineering
    - FS packages:
      * TPOT, Featurewiz, Featuretools, Defeature
    - Then create a new set of data
      * Original data plus feature construction
    - synthetic data generator -> ask Prof. Jafari for code and paper for this
      * To create synthetic data
    - Improve the model:
      * CNN, Transformer, Deep Neural Networks

# Feb 14 – 4th Class

* **Tasks for next week**
  + Code
    - Break down the **imputation functions** to be more “dynamic”
      * Identify data type, then label the encoding
      * Use other imputation techniques, e.g. can predicting labels (so not only filling with mean, mode, median)
        + **Ask professor to send sample code**
    - Compare clean data results with PyCaret
    - Create a **data explanation dictionary** in the code
    - Write code for **feature selection** (PCA, random forest, auto feature, etc.) and **feature engineering**
      * **Ask professor to send links/readings/sample code for feature engineering**
    - Improve the baseline **model modules**
      * Each model can have a function to run results/plots, breakdown the function as detailed as possible, instead of running everything
  + Paper
    - Start documenting the work we did for this week in the paper
  + Logistics
    - Clean up files on Github
* **Meeting with Dr. Gupta on Friday Feb 16**
  + Prepare a presentation to explain what we did, and show the baseline model results (accuracy, F-1 scores), get his feedback
  + Ask for more recent data
  + Ask about variables (features)
    - What are the more important ones to him
    - Get variable definitions from him
  + **Add professor to this meeting as optional**

# Feb 21 – 5th Meeting

* Update the model results table in report
* Improve class code
  + Able to select in utils -> run single result or combined results
* Don’t use TPOT
* Add clustering models
* Benchmark – mean imputation; no complex preprocessing (stdization); include all features that make sense