Weekly Sprint Planning

2020-05-29 /10:00-11:30 / Zoom

THEMES	WEEKS	DATES	GOALS
Investigation and Data Prep	3	27 April - 14 May	Identify project objectives and key data features + understand data dictionaries + transform data for machine learning tasks.
Model	1	15 - 21 May	Develop a classification model to apply group tags to end-uses for the Pharmacy building.
Model	1	22 - 28 May	Validate and evaluate models.
Scale + Analysis	2	29 May - 11 June	Expand the model to other UBC buildings (if time permits) + complete user-acceptance testing of model + identify conclusions + create visualizations of results + complete user-acceptance testing of dashboards + UBC mid-term presentation
Wrap-Up	2	12 - 26 June	Final report + package final code + UDL final presentation + UBCO final presentation
Total Weeks	**9**		

Week 6 - Initial Modeling, completed when you can run main.py through the 3 models and get some type of output
Week 7 - Tuning Model, completed when the models work well
Week 8 - Alternative 1 (Dashboard) vs Alternative 2 (Placeholder Dashboard) &Wrap-Up, completed with Grafana dashboard with sensors/counts of sensors and start on final report, and start finalizing unit testing
Week 9 - Wrap-Up, presentations + final report + package final code + complete unit testing

1. What was our goal/theme from last week?
   Goal: Feature scaling and modeling

2. Which tasks did we complete?
   - **Make categorical fields into smaller levels**
   - **Decide which feature selection technique to implement**
   - **Join metadata with pharmacy data**
   - **Develope code and tools to standardize inconsistent/missing information**

- **Detailed flowchart**
- **Meeting to brainstorm predicting EC tags using NC data**

3. Was there anything stopping us from finishing specific tasks?
    - Lack of clarity on NRCan labels
    - Aggregation functions took much longer than expected (some dependencies)

4. What tasks are still in progress?
    - Identify relevant features
    - Feature engineering
    - Finalizing the training and testing datasets

5. Are there any changes that need to be made?
    - Add docstrings to functions as you create them
    - Touch base more frequently to ensure we are all on the same track

6. What is our goal/theme for this week?
   Goal: Start modeling/comparing different models

7. What tasks need to be added/replenished to the Backlog?

8. What tasks are most important and should be pulled from Backlog to In progress?
    - Task 1
    - Task 2
    - Task 3

9. Are there any dependencies between In Progress tasks?
    a. If so, how will that be organized?

10. Who is going to be assigned to which tasks and update in Jira?

Person	In progress Tasks	New Tasks
Claudia	●	●
Connor	●	●
Eva	●	●
Alex	●	●

- **Investigate and choose data time range for dataset (Alex)**
    - Determine when SkySpark stops making changes to decide how far back and recent do we want for data pull
    - Query additional days of data and save as csv's (if needed)
- **Expand Function Flow & Status Flowchart (if possible) (All)**
    - Add input and output formats
        - If it can't be added to flowchart then represent in some other way (another document, prepopulated docstrings in the main, etc…)
- **Populating the Main function (Alex and Eva)**
    - Finalize functions for everything (merging data, cleaning UOM, scaling data etc.) and integrate into main function
    - Encode data
    - Scale numerical data
    - Drop unwanted features
    - Feed the above into clustering models
- **Develop/test various models for NC sensors (Connor)**
    - Decide which models to implement (meet and brainstorm? And test various options, we have code that we can use to test the following)
        - DBSCAN (Accounts for noise values so they don't mess with the clusters)
        - HDBSCAN (Accounts for noise values so they don't mess with the clusters)

        - Gaussian Mixture Models
        - Variational Bayesian estimation of a Gaussian mixture
        - K-Means
        - Hierarchical
        - Fuzzy C-Means
        - Mean Shift
    - Write code to join labels from the clustering process with the raw NC sensor data
        - Will probably need an intermediate step of joining cluster labels with the unique sensor ids first (separate function? Or embed within? We probably only need to do this once, so embedding would be fine if that is easier)
    - Write code to test different models (grid search?)
- **Develop code for getting predictor variables for the EC tags from the NC tags (Model EC and NC relationship) (Claudia and Eva)**
    - Alternative 1 (prefered?): Cluster NC sensors and fit a linear model to each EC sensor with the clusters as predictor variables, take the coefficients from the model fits and use as predictor variables for the EC sensors
        - Decide if we want LM, LASSO, RELAXO, or some combo of the three (probably LASSO or RELAXO)
        - Find packages that perform the type of modeling we want
        - Write code to fit linear models for predicting the value of the EC sensor measurement given the NC Sensor cluster data
        - Write code to test the above fit, and if the error is better than some threshold to extract and return the coefficients?
            - Otherwise try dropping some features or some interaction effects?
            - Or once we have tested if this is going to work should we just drop the testing data part of this and just fit the model to all of the data and take the coefficients?
        - Extract model coefficients

    - Alternative 2 (if 1 doesn't work?): Cluster non-NRCan sensors and do the following per group and use these values as predictors for NRCan labels
        - Average non-NRCan sensor value per hour of the day
        - Average daily non-NRCan sensor value
        - Average non-NRCan sensor value per month (or week?)
    -
- **Develop various classification models for NRCan tags (Alex and Connor)**
    - Supervised (possible options that may work well with our data, b/c quite flexible)
        - Random forest
        - Neural Net
            - ANN or RNN?
        - Bagging
        - Boosting
    - Semi-Supervised (possible options that may work well with our data)
        - Label Spreading
        - Label Propagation
        - HiddenMarkovModel Semi-supervised
        - BayesClassifier Semi-supervised
        - NaiveBayes Semi-supervised
    - Unsupervised
        - Clustering
            - Could be an option if we are having troubles, it may give insightful clusters (which is better than nothing if our NRCan-ish labels aren't working)
        - Boltzmann Machine (an unsupervised Neural Net) (not sure if this will work but it may be worth looking into if the others don't? Probably more of an interesting thing to try

after the capstone is done as it may take too long to implement)
        - Boltzmann Machines attempt to model a system and interpret how different sensors interact by taking values and applying weight to nodes (commonly used in recommendation systems)
        - Theoretically we could create a Boltzmann Machine to emulate the system with the different NRCan tags as the visible nodes and then calculate the probability of a given sensor belonging to that node given the current (or average) reading from the sensor
- **Compare models for effectiveness (move to next sprint)**
    - Research ways to evaluate models/choose performance metrics for comparison
    - Develop function for the chosen performance metrics (if needed)
    - Outline reasons for chosen models
    - Develop code to grid search for optimal model/model configuration/features
        - Sklearn has something for this that works well would just need to code it up
- **Recommend models (move to next sprint)**
    - Create a flowchart/report to show chosen model and outputs