**MTC Assurance Plan**

Created by Higley, Daniel, last modified by Miller, Scott on Aug 16, 2022

This is the assurance plan for updates to the MTC push model. This will be put in the model card for MTC.

References:

ML Assurance Strategy

Test Plan for Complaints Cause Code Classification

Required artifacts from model developers for beginning model validation

✓Model file:

✓Easily executable model training pipeline that uses data from DM

Easily executable model inference pipeline

✓Model Card

**Model Performance:**

✓Metrics + Uncertainties on test data set, comparison to original model

✓Model performance on subsets of data (likely chosen through clustering for now)

✓~~Assess impact of dynamic min-max scaling (Cancelled since we are planning on removing dynamic Min-Max-scaling)~~

✓~~Take dataset, add 100 new synthetic data points with completely random features, and compare model scores before and after adding synthetic points (James idea)~~

✓~~Take dataset, choose 100-random-10-symbol subsets including both positives and negatives, and run model on those 10-symbol subsets. Compare model scores for subsets runs versus entire day runs~~

Handling of unexpected inputs (like NANS)

Model performance on small datasets for which we have more confidence in labels (do if available) (actually, it doesn't look like we have this, so can't do)

Evaluate quality of labels for -10 observations based on documented labeling procedure (to be done once labeling procedure and newly labeled observations are available)

SHAP sensitivity analysis

Test for adversarial robustness (Navid will handle this)

**Data Quality:**

✓Training-Serving skew tests (I don't think we can do this that well given available information, but some stuff we can do)

✓TSfresh features should all be Min-Max scaled. In other words, they should all have a minimum of 0 and maximum of 1

Assess for potential data leakage

✓Performance on test set vs training

Check for features with suspiciously large importances

✓Check for duplicates, nulls in training/validation/test data

✓Check for multicollinearity→ this is a significant thing

✓Dependence of model performance on amount of data (learning curves)

**Model Selection & Training**

Reproducibility

✓Overfitting/underfitting analysis (select metrics on training, validation and test data)

✓Comparison to simpler or default models (e.g. logistic regression, random forest)

Evaluate impact of dimensionality reductions

Compare to model without dimensionality reduction

Compare to different, simple dimensionality reduction (PCA or ICA)

✓Validate model versioning infrastructure (MDLC requirement)

✓Determine how long model training/tuning takes

**Model monitoring plan**

Model drift analysis

**Exploratory**

Calculate feature importances

# Explore the distribution of features in the model

**Verifications**

Model meets its design objectives as defined in the Model Card

Model adheres to the data approach defined in the Model Card

Model adheres to the selection process defined in the Model Card

Model has appropriate risk-based monitoring plan

Other parts of model card are accurate

**Code quality/maintainability**

Lines of code

Would like to do, but probably won't be able to:

• Performance under data transformations that shouldn't impact model predictions (e.g., price scaling)

• Directional expectation tests (e.g., more profit more likely manipulation)

• Model performances at extreme ranges of input data (establish model boundaries). Will have to think about whether we can do this in a reasonable way

• Comparison to AutoML

•Evaluate potential of different methods of handling class imbalance

**Discovered risks:**

Business:

• HIGH: Min-Max scaling is applied dynamically per day instead of the standard practice of determining scaling parameters from training data and consistently applying the same scaling to new data. (existed in original version of model, but not caught)

• LOW: tsfresh does not take into account absolute time information, but only relative information. As a result... (existed in original version of model, but not caught)

• Medium: training-serving skew

Technical:

• Medium: Labeling procedure is not documented (existed in original model. I upgraded this from low to medium risk.)

• Medium: Feature selection is not automated and the manual process for performing feature selection is not documented (existed in original model, but not caught)