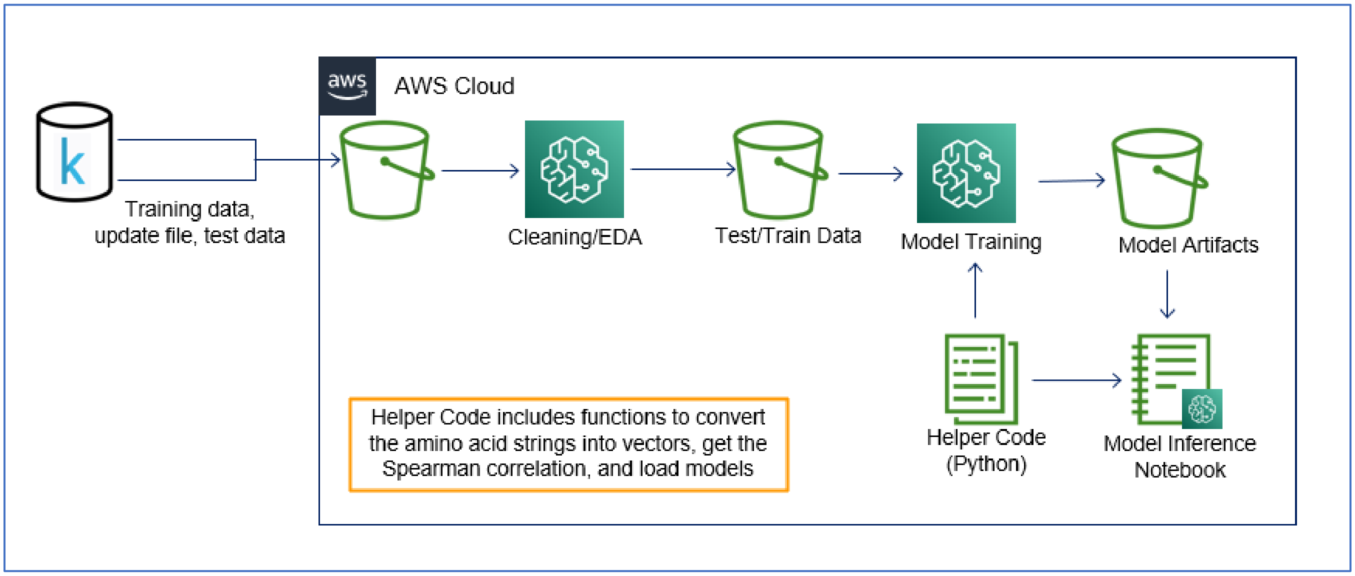
Team 4 – An Army of One

Enzyme Stability Prediction Model

Created by: Rivka Howell, April 2023

Run Notes: Shutdown of Notebooks seems to require pip installing biopython even if done previously

Architecture and Design:



Helper Code includes functions to convert the amino acid strings into vectors, get the Spearman correlation, and load models

Latest Version Summary (v3):

Model: Random Forest

Features: amino acid words with 1, 2, and 3 letters, pH, sequence length, molecular weight, isoelectric point, percent helix, percent sheet, percent turn, aromaticity, instability, charge, and gravy (grand average hydropathy)

Trained on: 20,689 sequences with pH from 6-8 inclusive

- Data – stored in S3 (**tech-x-final-project**)
    - Input from Kaggle is stored with prefix "**raw-data/**"
        - Includes file to update training data (train_updates_2022-929.csv)
    - Post EDA/cleaned data is stored with prefix "**clean-data/**"
        - Includes test and train split (clean_test and clean_train, respectively)
        - Clean data is NOT vectorized
    - All stored in csv format
    - Vectorized files are not stored
- Artifacts – stored in S3 (**tech-x-final-project/models**)
    - 3 artifacts per version
        - Model – the regression or random forest model artifact
        - Scaler – the StandardScaler object associated with the features
        - Features – the amino-acid sequence features (aka, the vocabulary) used
    - Best version saved with plain name in addition to the version name

- Files/Code – stored in Notebook Instance nb2-rsh (and Git Repo)
    - DataIngestAndEDA.ipynb – reads in raw training data, does cleaning/updates (from Kaggle)
        - does EDA/plots, test/train split
        - Removes duplicates, including duplicate amino acid sequences (protein_sequence variable)
        - Limits to pH of 6-8 inclusive
    - Features_Modules.py (a.k.a "helper code")
        - Functions to read clean data, preprocess it, use CountVectorizer (sklearn), and load model artifacts
    - Features_Modules2.py (a.k.a "helper code")
        - Second version of Feature Modules
        - Includes options for biopython (Bio library) features
    - Regression1.ipynb – runs benchmark model
        - Uses amino acid frequency, pH, and the length of the sequence with a linear regression (scaled inputs)
        - Includes printout of Spearman Correlation/R-squared for test data
        - Includes printout of Spearman Correlation/R-squared for Kaggle data
        - Persisted artifacts start with "**benchmark_**"
    - Regression2.ipynb – modified from benchmark
        - Adds 2-amino acid long sequences to vocabulary, otherwise the same as Regression1
        - Persists as "**reg2_**"

    - **RandomForest3**.ipynb **– BEST MODEL**
        - Runs a random forest model with biopython feature options, pH, sequence length, and amino acid sequences of length 1, 2, and 3
        - Utilizes Features_Modules2
        - Persist is stored as “**rf3_**” and also without prefix (i.e. just model.pkl, features.pkl, and scaler.pkl) as the best model
- Inference Notebooks - stored in Notebook Instance nb2-rsh (and Git Repo)
    - Make Inference
        - Manually enter a sequence and pH and runs model, returning prediction value
    - Make Inference-kaggle
        - Reads in Kaggle test data set and returns the metrics (Spearman Correlation/R^2)