Sina Khosravi

Department of Medicinal Chemistry, School of Pharmacy, Mashhad University of Medical Sciences, Mashhad, Iran

Gmail : Khosravi.sina2001@gmail.com

Overview

For this competition, I developed predictive models for two proteins, SARS and MERS, using a variety of machine learning techniques and descriptor sets. My approach involved a consensus model strategy, where predictions from individual models were averaged without weighting. The process began with data analysis and preprocessing, followed by feature engineering and model training, with performance assessed using 5-fold cross-validation (CV=5).

Data Analysis and Preprocessing

I started by examining the dataset with DataWarrior software to assess scaffold diversity and the distribution between the training (train) and test sets. The analysis showed a strong overlap between the two sets, indicating a well-established applicable domain. For each protein, I identified missing values (NaN) and duplicates in the data. I removed all NaN entries and addressed duplicates by computing the mean for pairs and the median for groups of three or more. Additionally, I attempted to eliminate significant outliers to improve data quality.

Feature Engineering

Descriptors were calculated using multiple tools: the OCHEM web platform, the scikit-fingerprints package, and embedding features from Chemprop. To prepare the data for modeling, I filtered out features with constant values (identical across all entries), variance below 0.01, or correlations exceeding 95%. Feature selection varied by model and protein, often leveraging SHAP values or genetic algorithms, as detailed below.

Model Tuning and Evaluation

Hyperparameters were tuned using grid search to prevent overfitting, with careful selection of parameters to maintain model generalization. All models were evaluated using 5-fold cross-validation (CV=5) to ensure robust performance metrics.

Modeling Approaches

Below, I outline the methods applied to each protein, including feature selection and CV=5 performance scores.

SARS

1. **XGBoost**: Trained on all extracted features, followed by SHAP-based feature selection retaining the top 3.5%. Feature count reduced from 5,440 to 191. CV=5 score: 0.366.
2. **LightBoost**: Used all extracted features, with SHAP selecting the top 2.5%. Feature count dropped from 5,599 to 140. CV=5 score: 0.4655.
3. **CatBoost**: Optimized feature combination (-map4-mold2-Estate) identified via genetic algorithm, with the top 20% of SHAP-selected features retained. Feature count decreased from 1,376 to 275. CV=5 score: 0.401.
4. **SVR**: Trained on a descriptor combination (PubChem, ECFP, MACCS) without further feature selection, using 1,109 features. CV=5 score: 0.4232.

MERS

1. **XGBoost**: Applied all extracted features, with SHAP selecting the top 4.5%. Feature count reduced from 5,599 to 252. CV=5 score: 0.4622.

2. **LightBoost**: Used all extracted features, with SHAP retaining the top 2.5%. Feature count dropped from 5,599 to 140. CV=5 score: 0.4655.

3. **CatBoost**: Optimized feature combination (-map4-mold2-RDKit-FFN(Chemprop)) via genetic algorithm, with the top 20% of SHAP-selected features kept. Feature count decreased from 1,672 to 335. CV=5 score: 0.441.

4. **SVR**: Trained on a descriptor combination (PubChem, ECFP, MACCS) without additional feature selection, using 1,186 features. CV=5 score: 0.4562.

5. **XGBoost with Clustering**: Reduced dimensionality to two using PCA with PubChem descriptors, followed by clustering of train and test data via the nearest neighbor algorithm. SHAP-selected features were used to train models for each cluster. Cluster 0: train = 285, test = 133, CV=5 = 0.44; Cluster 1: train = 562, test = 164, CV=5 = 0.46.