Challenge 1: Drug Activity | sub10

MICHAEL RAFFELSBERGER

ACM Reference Format:
Michael Raffelsberger. 2022. Challenge 1: Drug Activity | sub10. 1, 1 (April 2022), 1 page.

1 EXPLORATORY DATA ANALYSIS

I started with some data exploration with the following results: We have $n = 12000$ samples of smiles strings with a length between 1 and 1157 containing a total of 52 different characters. Most samples have 1 or 2 labels (i.e. not missing), though some also have 0 or up to 9 labels. Task 4 has only few labeled samples ($n_4 = 610$) while Task 7 has most with $n_7 = 4311$. Task 10 has the highest positive rate with 76% whereas Task 8 is extremely imbalanced with only 1% positives. For my personal convenience, I encoded positive labels with 1, negative labels with 0 and missing values with -999.

2 JACCARD PCA EMBEDDING

I extracted Morgan fingerprints (nBits=256, radius=3) for all 12000 molecules and computed all pairwise Jaccard distances resulting in a 12000×12000 matrix. Afterwards, I used PCA to basically get a low-dimensional molecule embedding for the following purposes:

(1) For k-means clustering to prevent very similar molecules to end up in different cross-validation folds.
(2) For feature extraction in a semi-supervised fashion, which justifies not performing this within cross-validation.

3 SEMI-SUPERVISED LSTM EMBEDDING

Originally, I wanted to train a multi-task LSTM directly on the raw smiles strings using a character embedding layer, which did not work too well though. To not just throw away all the code, I decided to train the LSTM with the PCA features from before as targets in order to obtain a character embedding. The whole idea was quite useless, but might work with more molecules and a more meaningful pretraining objective. Still, I used the average (or component-wise maximum) embedding of each smiles string as features in my supervised models.

4 CLUSTERING & DATA SPLITTING

Based on the PCA on the Jaccard distances from before, I used k-means clustering (k=100) to prevent too many similar molecules to end up in different cross-validation folds. k=100 may seem a bit high, but helped to keep the folds similar in terms of balancedness. In retrospect, I would still lower it a little bit. The folds were then generated with `GroupKFold` with $K = 5$.

5 FEATURE ENGINEERING

Feature ideas that I at least tried out:

- Morgan fingerprints: I played with the radius (2-3) and nBits (256-2048). [used]
- Smiles length [used]
- Document-term matrix (dtm): Used with characters as terms, unigrams and bigrams with a maximum of 100 features. [used]
- Tf-idf: Same idea as dtm. [not used]
- PCA embedding [semi-supervised][not used]
- LSTM embedding: E.g. averaged over all characters of a given smiles string. [semi-supervised][used]
- RDKit descriptors: Adjusted the Ipc feature as $\log(1 + Ipc)$ due to large values. [used]

6 MODELING

Although ROC-AUC takes into account imbalancedness to some extent, it is definitely a questionable metric in highly imbalanced settings, such as Task 8. Still, I tried to alleviate the imbalancedness by using sample weights: By doing so, I gave a class with share p in the training folds share $\tilde{p} = \sigma(\text{coef} \cdot p - \text{coef}/2)$, where I varied the parameter coef in the range $[0, 4]$. I used 4 different models:

- Random forest: I tuned `n_estimators` (200), `min_samples_split` (20) and `max_features` (8).
- SVC: I tuned `C` (0.1) and the `kernel` (rbf).
- Logistic regression: Ridge penalty (l2) with tuned `C` (0.01).
- LightGBM classifier: I tuned `num_leaves` (32), `max_depth` (5), `min_childs` (40), `learning_rate` (0.1), `n_estimators` (50) and `reg_lambda` (10).

Each model was trained separately for every task. The above hyperparameters might be wrong since I played around quite a bit, but give a good impression of what I used. The preprocessing is similar for all models but still individual. All models achieved very high training AUCs (≈ 0.99) and lower test AUCs ($\approx 0.75 - 0.77$). It looks like overfitting, but every regularization attempt turned out to hurt the validation AUC as well. I am not entirely sure, but attribute some of the weird behavior in this challenge to the unpleasent behavior of ROC-AUC under imbalancedness. I also trained an MLP for multitask-learning, but it did not work as well as expected which is why I focused on the above models.

7 ENSEMBLING & "LEADERBOARD OVERFITTING"

Finally, I performed a final fit of all models on all 12000 training samples and averaged their predictions (i.e. ensembling with equal weights). I did multiple submissions with e.g. more/less regularization/sample weighting/... and exchanged the Task 2 predictions from my best submission with those of another submission. Even though this looks a little like public leaderboard overfitting, I would argue it is not since the difference was more than significant. In the end, I just relied on the random forest and the LGBM classifier for Task 8. In particular, the logistic regression (presumably my least flexible model) predicted somewhat higher probabilities there and destroyed the public leaderboard score quite a bit. Without knowing for sure, I blame the ROC-AUC metric once again. Probably some metric-specific post-processing would have paid off.

Author's address: Michael Raffelsberger, k11772903@students.jku.at.