Drug character prediction using Blood-Brain Barrier Prediction dataset

Dhanusha Duraiyan,
Sai Jagan Lakku,
Durga Sai Sailesh Chodabattula &
Allampati Raja Simha Reddy Gangadhar.

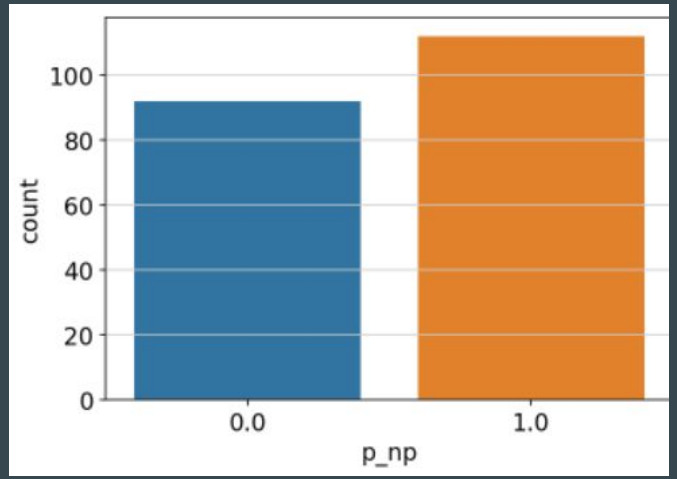
BBBP dataset

Blood-brain barrier penetration (BBBP)

- used to model and forecast barrier permeability.
- majority of drugs, hormones, and neurotransmitters are blocked by the blood-brain barrier, which separates circulating blood from brain extracellular fluid.
- As a result, getting past the barrier has long been a challenge in the creation of central nervous system drugs.
- includes binary labels for over 2000 compounds on their permeability properties.

```
bbbp_df.head()
```

	num	name	p_np	smiles
0	1	Propanolol	1	[Cl].CC(C)NCC(O)COc1cccc2ccccc12
1	2	Terbutylchlorambucil	1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl
2	3	40730	1	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...
3	4	24	1	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C
4	5	cloxacillin	1	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...



BBBP dataset contains 2053 items with four attributes :

- the index number from 1 to 2053 (“num”)
- the name of the compound (“name”)
- the penetrating or nonpenetrating properties (“p_np”)
- the SMILES string of the compound (“smiles”)

SMILES strings represent the chemical structures.

These SMILES are converted to fingerprint.

Fingerprint - fixed length array, where different elements indicate the presence of different features in the molecule.

Similar fingerprints ⇒ Same features and similar chemistry.

Scaffold split

Scaffold split algorithm MoleculeNet - split into training, validation, and test data; creates an unbalanced split ⇒ prediction harder.

split ⇒

(1) the molecular compounds - grouped into scaffold sets on the basis of the skeletal ring structures termed scaffolds

(2) the compounds and scaffold sets are sorted in reverse order (largest to the smallest)

(3) the dataset are split into training, validation, and test data in an 8:1:1 ratio from the top.

The simplified molecular-input line-entry system (SMILES) is a specification in form of a line notation for describing the structure of chemical species using short ASCII strings.

There are almost 1800 characteristics of each smile which needs to examined to determine which characteristics influence the penetration of a drug

```
coerce_to_numeric(bbbp_descriptors_df, bbbp_descriptors_df.columns)
bbbp_descriptors_df.head()
```

A	VR1_A	VR2_A	VR3_A	nAromAtom	nAromBond	nAtom	nHeavyAtom	nSpiro	nBridgehead	nHetero	nH	nB	nC	nN
N	NaN	NaN	NaN	10	11	41	20	0	0	4	21	0	16	1
5	210.341501	9.145283	6.181642	6	6	50	23	0	0	5	27	0	18	1
2	228.201019	8.776962	6.385738	10	11	46	26	0	0	8	20	0	18	3
7	317.983605	15.142076	6.503937	6	6	47	21	0	0	4	26	0	17	2
4	717.026462	24.725050	7.639823	11	11	47	29	0	0	10	18	0	19	3

Requirement for different models

By sorting the scaffold sets from the largest to smallest, the test data will consist of compounds that are less related to others

⇒ the prediction model easily overfits to the training data

However, finding a good model under this difficult condition is valuable in predicting unknown data. Hence we train our dataset using different models :

- Logistic Regression
- Lasso Classifier (L1)
- Elastic Net Classifier (L1 + L2)
- Support Vector Classifier
- Gaussian Naive Bayes Classifier
- Random forest

From the result we can find the model which works better for the dataset.

K - fold cross validation

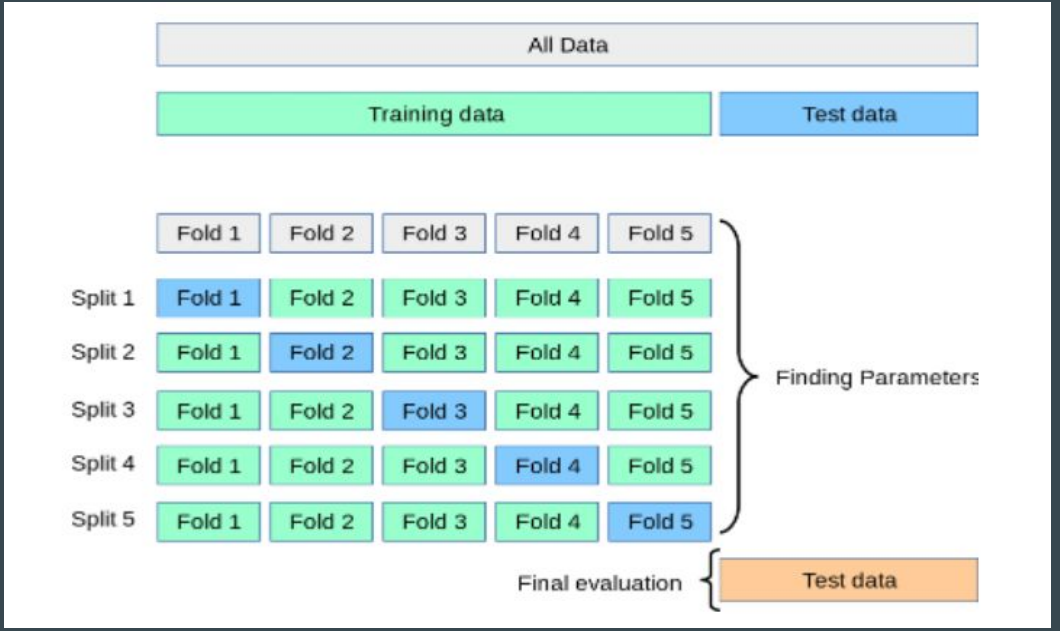
Implemented in order to reduce the overfitting of the data.

In our project, we used k = 5.

Also, we are using shuffle split

to increase randomness for better

results.



```python
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(penalty = "l2", C = 0.5)
train_scores, valid_scores, lr = k_fold_generator(fp_array, y, 5, lr, 'lr')

print('Mean training scores', np.mean(train_scores), ',  Mean validation scores', np.mean(valid_scores))
```

```
Mean training scores 0.9708720214974041 ,  Mean validation scores 0.9203109193734192
```

```python
print(classification_report(lr.predict(fp_test_array), y_test))
```

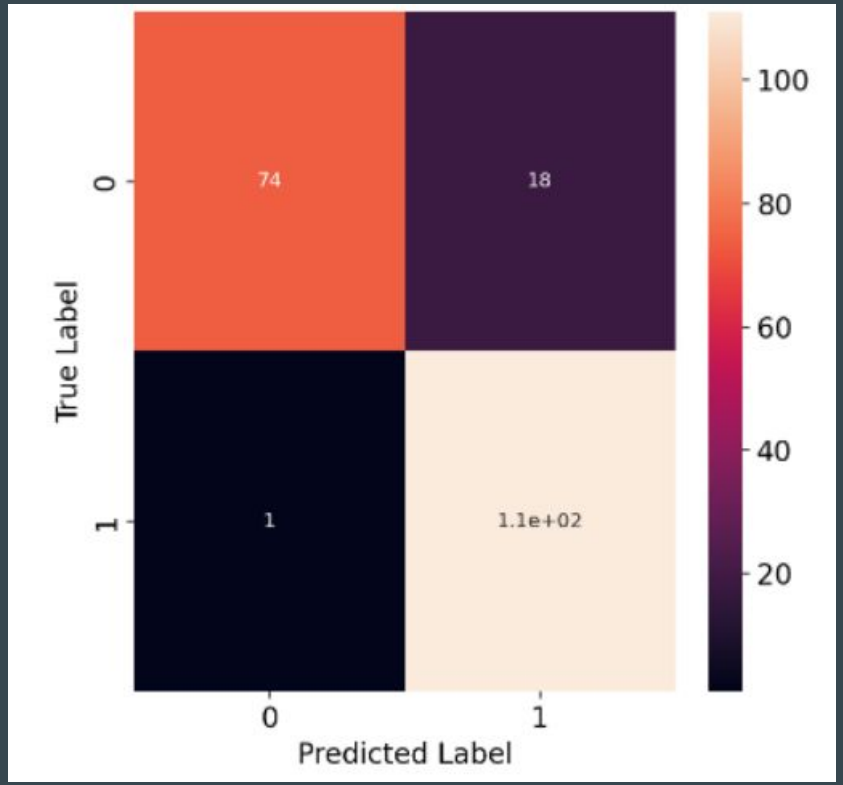
```
              precision    recall  f1-score   support

         0.0       0.80      0.99      0.89        75
         1.0       0.99      0.86      0.92       129

    accuracy                           0.91       204
   macro avg       0.90      0.92      0.90       204
weighted avg       0.92      0.91      0.91       204
```

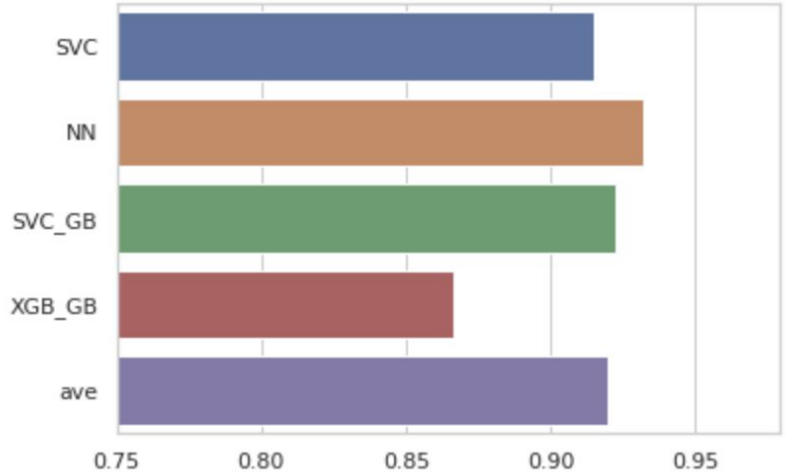
Confusion matrix for Linear model :

- Precision
- recall
- f1-score
- support



```
sns.set(style="whitegrid")
ax = sns.barplot(x=[svc_score,nn_score,svc_gb_score,xgb_gb_score,ave_score],
                 y=['SVC','NN','SVC_GB','XGB_GB','ave'])
ax.set(xlim=(0.75, None))
```

```
[(0.75, 0.9787037037037037)]
```



+ Code + Markdown

Future Scope

- We prefer to see more, like how, why and any interesting findings on the permeability. For example, which properties have played a decisive role on permeability, and how you can prove it.
- Implement this model in deep learning neural networks

Thank you!