In [1]:
```
import pandas as pd
import numpy as np
import pickle
from tensorflow.keras.models import load_model
import warnings
from sklearn.metrics import log_loss
from tqdm import tqdm
warnings.filterwarnings("ignore")
```

In [2]:
```
def final_fun_1(X):

    #encoding categorical features
    X.iloc[:,1] = X.iloc[:,1].map({'trt_cp':0, 'ctl_vehicle':1})
    X.iloc[:,2] = X.iloc[:,2].map({24:0, 48:1, 72:2})
    X.iloc[:,3] = X.iloc[:,3].map({'D1':0, 'D2':1})

    #normalizing gene and cell columns
    transformer = pickle.load(open('transform.pkl','rb'))
    X.iloc[:,4:] = transformer.transform(X.iloc[:,4:])

    #getting more features using auto-encoder
    encoder = load_model('encoder.h5')
    encoded_features = encoder.predict(X.iloc[:,1:])

    #adding encoded features with original features
    total_X = pd.concat([X,pd.DataFrame(encoded_features)], axis=1)

    #loading pre-trained model
    model = pickle.load(open('final_model.pkl','rb'))

    #loading column names of target columns
    columns = pickle.load(open('target_columns.pkl','rb'))

    #predictions
    pred = model.predict(total_X.iloc[:,1:])
    pred_data = pd.DataFrame(pred, columns = columns)
    pred_data.insert(loc=0,column='sig_id',value = X['sig_id'])

    pred_prob = model.predict_proba(total_X.iloc[:,1:])
    pred_prob_data = pd.DataFrame(pred_prob,columns=columns)
    pred_prob_data.insert(loc=0,column='sig_id',value = X['sig_id'])

    return pred_data, pred_prob_data
```

In [3]:
```
def final_fun_2(X,y):

    #encoding categorical features
    X.iloc[:,1] = X.iloc[:,1].map({'trt_cp':0, 'ctl_vehicle':1})
    X.iloc[:,2] = X.iloc[:,2].map({24:0, 48:1, 72:2})
    X.iloc[:,3] = X.iloc[:,3].map({'D1':0, 'D2':1})

    #normalizing gene and cell columns
    transformer = pickle.load(open('transform.pkl','rb'))
    X.iloc[:,4:] = transformer.transform(X.iloc[:,4:])

    #getting more features using auto-encoder
    encoder = load_model('encoder.h5')
    encoded_features = encoder.predict(X.iloc[:,1:])

    #adding encoded features with original features
    total_X = pd.concat([X,pd.DataFrame(encoded_features)], axis=1)

    #loading pre-trained model
    model = pickle.load(open('final_model.pkl','rb'))

    #prediction
    pred_prob = model.predict_proba(total_X.iloc[:,1:])

    metric_value = []    #list to store log-loss of each target features

    for i in tqdm(range(y.shape[1]-1)):    #iterating over each target columns
        loss = log_loss(y.iloc[:,i],pred_prob[:,i], labels=[0, 1])    #computing log-loss
        metric_value.append(loss)


    return np.mean(metric_value)
```

In [10]:
```
test = pd.read_csv('test_features.csv')
train_X = pd.read_csv('train_features.csv')
train_y = pd.read_csv('train_targets_scored.csv')
```

In [11]:
```
#calling first function
pred, pred_prob = final_fun_1(test)
```

```
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
```

In [13]:
```
pred.head()
```

Out[13]:

	sig_id	5-alpha_reductase_inhibitor	11-beta-hsd1_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acetylcholine_receptor_antagonist	acetylcholinesterase_inhibitor	adenosine_receptor_ago
0	id_0004d9e33	0	0	0	0	0	0	
1	id_001897cda	0	0	0	0	0	0	
2	id_002429b5b	0	0	0	0	0	0	
3	id_00276f245	0	0	0	0	0	0	
4	id_0027f1083	0	0	0	0	0	0	

5 rows × 207 columns

In [14]:
```
pred_prob.head()
```

Out[14]:

	sig_id	5-alpha_reductase_inhibitor	11-beta-hsd1_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acetylcholine_receptor_antagonist	acetylcholinesterase_inhibitor	adenosine_receptor_ago
0	id_0004d9e33	0.000366	0.001299	0.001528	0.016108	0.009998	0.003968	0.00(
1	id_001897cda	0.000017	0.000689	0.000433	0.001397	0.002086	0.005395	0.00:
2	id_002429b5b	0.000073	0.000113	0.000447	0.002381	0.011240	0.002011	0.00(
3	id_00276f245	0.000134	0.000271	0.000577	0.007121	0.003752	0.007211	0.00(
4	id_0027f1083	0.000802	0.000710	0.000878	0.011682	0.019587	0.002861	0.00:

5 rows × 207 columns

In [12]:
```
#calling second function
metric_val = final_fun_2(train_X, train_y)
```

```
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
100%|██████████| 206/206 [00:03<00:00, 67.29it/s]
```

In [15]:
```
print("Metric value: ",metric_val)
```

```
Metric value:  0.028532469701987095
```

In [ ]:
```

```