Setup

```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
Mon Mar 11 05:36:31 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   66C    P8              12W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
  print('Not using a high-RAM runtime')
else:
  print('You are using a high-RAM runtime!')
```

```
Your runtime has 13.6 gigabytes of available RAM

Not using a high-RAM runtime
```

```
!pip install chemprop
!pip install rdkit-pypi  # should be included in above after Chemprop v1.6 release

import chemprop
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.offsetbox import AnchoredText
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.decomposition import PCA
```

```
Requirement already satisfied: imagesize in /usr/local/lib/python3.10/dist-packages (from sphinx>=3.1.2->chemprop
Requirement already satisfied: requests>=2.5.0 in /usr/local/lib/python3.10/dist-packages (from sphinx>=3.1.2->ch
Requirement already satisfied: protobuf>=3.20 in /usr/local/lib/python3.10/dist-packages (from tensorboardX>=2.0-
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->chemprop)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->c
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->chemprop) (1.
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->chemprop) (2
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->chemp
Collecting typing-inspect>=0.7.1 (from typed-argument-parser>=1.6.1->chemprop)
  Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Collecting docstring-parser>=0.15 (from typed-argument-parser>=1.6.1->chemprop)
  Downloading docstring_parser-0.15-py3-none-any.whl (36 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.0->flas
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.5.0->sph
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.5.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.5.
Collecting mypy-extensions>=0.3.0 (from typing-inspect>=0.7.1->typed-argument-parser>=1.6.1->chemprop)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.4.0-
Building wheels for collected packages: typed-argument-parser
  Building wheel for typed-argument-parser (setup.py) ... done
  Created wheel for typed-argument-parser: filename=typed_argument_parser-1.9.0-py3-none-any.whl size=25615 sha25
  Stored in directory: /root/.cache/pip/wheels/f0/94/0f/9539f578bed7e1bd423c702e403712f5ee8989f831a71db000
Successfully built typed-argument-parser
Installing collected packages: tensorboardX, rdkit, mypy-extensions, docstring-parser, typing-inspect, typed-argu
Successfully installed chemprop-1.6.1 docstring-parser-0.15 mypy-extensions-1.0.0 pandas-flavor-0.6.0 rdkit-2023.
Collecting rdkit-pypi
  Downloading rdkit_pypi-2022.9.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (29.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 29.4/29.4 MB 53.7 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from rdkit-pypi) (1.25.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from rdkit-pypi) (9.4.0)
Installing collected packages: rdkit-pypi
Successfully installed rdkit-pypi-2022.9.5
```

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

```
hiv_df = pd.read_csv("HIV.csv")
hiv_df.head()
```

	smiles	activity	HIV_active
0	CCC1=[O+][Cu-3]2([O+]=C(CC)C1)[O+]=C(CC)CC(CC)...	CI	0
1	C(=Cc1ccccc1)C1=[O+][Cu-3]2([O+]=C(C=Cc3ccccc3...	CI	0
2	CC(=O)N1c2ccccc2Sc2c1ccc1ccccc21	CI	0
3	Nc1ccc(C=Cc2ccc(N)cc2S(=O)(=O)O)c(S(=O)(=O)O)c1	CI	0
4	O=S(=O)(O)CCS(=O)(=O)O	CI	0

```
hiv_df.describe()
```

	HIV_active
count	41127.000000
mean	0.035086
std	0.184001
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
unique_values = hiv_df['HIV_active'].unique()
print(f"Unique values in 'HIV_active': {unique_values}")
```

```
Unique values in 'HIV_active': [0 1]
```

```
unique_values = hiv_df['smiles'].unique()
print(f"Unique values in 'smiles': {unique_values}")
print(f"length of uniqe value: {len(unique_values)}")
```

```
Unique values in 'smiles': ['CCC1=[O+][Cu-3]2([O+]=C(CC)C1)[O+]=C(CC)CC(CC)=[O+]2'
 'C(=Cc1ccccc1)C1=[O+][Cu-3]2([O+]=C(C=Cc3ccccc3)CC(c3ccccc3)=[O+]2)[O+]=C(c2ccccc2)C1'
 'CC(=O)N1c2ccccc2Sc2c1ccc1ccccc21' ...
 'Cc1ccc(N2C(=O)C3c4[nH]c5ccccc5c4C4CCC(C(C)(C)C)CC4C3C2=O)cc1'
 'Cc1cccc(N2C(=O)C3c4[nH]c5ccccc5c4C4CCC(C(C)(C)C)CC4C3C2=O)c1'
 'CCCCCC=C(c1cc(Cl)c(OC)c(-c2nc(C)no2)c1)c1cc(Cl)c(OC)c(-c2nc(C)no2)c1']
length of uniqe value: 41127
```

```
# Filter rows where 'your_column' is not equal to 1 or 0
filtered_df = hiv_df[(hiv_df['HIV_active'] != 1) & (hiv_df['HIV_active'] != 0)]
filtered_df
```

	smiles	activity	HIV_active

```
# Filter rows where 'target_column' is equal to 1h
hiv_df_filtered_active = hiv_df[hiv_df['HIV_active'] == 1]
hiv_df_filtered_active
```

	smiles	activity	HIV_active
11	O=C(O)Cc1ccc(SSc2ccc(CC(=O)O)cc2)cc1	CM	1
16	NNP(=S)(NN)c1ccccc1	CM	1
80	O=Nc1ccc(O)c(N=O)c1O	CM	1
203	Oc1ccc(Cl)cc1C(c1cc(Cl)ccc1O)C(Cl)(Cl)Cl	CM	1
234	NNC(=O)c1ccccc1SSc1ccccc1C(=O)NN	CM	1
...	...	...	...
41090	Cc1cn(COCCCOCC(=O)c2ccccc2)c(=O)[nH]c1=O	CM	1
41092	Cc1cn(C2CC3C(COC(CCC[Se]c4ccccc4)N3O)O2)c(=O)[...	CM	1
41093	Cc1cn(C2CC3C(COC(CCCC[Se]c4ccccc4)N3O)O2)c(=O)...	CM	1
41098	Cc1cn(C2CC3C(COC(CC[Se]C#N)N3O)O2)c(=O)[nH]c1=O	CM	1
41099	C[Se]CCC1OCC2OC(n3cc(C)c(=O)[nH]c3=O)CC2N1O	CA	1

1443 rows × 3 columns

```
# Filter rows where 'target_column' is equal to 1h
hiv_df_filtered_inactive = hiv_df[hiv_df['HIV_active'] == 0]
hiv_df_filtered_inactive = hiv_df_filtered_inactive.sample(n=1500, axis=0, random_state=42)
hiv_df_filtered_inactive
```

	smiles	activity	HIV_active
2428	O=C1c2ccccc2-c2nc3ccccc3nc21	CI	0
6197	O=C(CSc1cc(-c2ccc(Cl)cc2)s[s+]1)c1ccccc1	CI	0
17138	O=C(C=Nc1ccccc1C(=O)O)c1ccco1	CI	0
12261	CCCCCCCCCCCCCCCCCC[N+](C)(C)Cc1ccc(C[N+](C)(C)...	CI	0
3588	N#CSC1CCCCCCC1SC#N	CI	0
...	...	...	...
18477	CC(=O)OC1(C#N)CC2OC1C1C2N1C(=O)OC(C)(C)C	CI	0
1189	CCOC(=O)C1Cc2cc(C)c(C)cc2N(C)C1=O	CI	0
36657	CCOC(=O)N1CCN(c2ccc3c(C)cc(C)nc3n2)CC1	CI	0
27919	CN(C)C=Nc1ccc2c3c(cccc13)-c1ccccc1-2	CI	0
13479	CCC1CC2CC3c4[nH]c5ccc(OC)cc5c4CCN(C2)C13.Cl	CI	0

1500 rows × 3 columns

```
hiv_df_sampled = pd.concat([hiv_df_filtered_active, hiv_df_filtered_inactive], axis=0, ignore_index=True)
hiv_df_sampled
```

	smiles	activity	HIV_active
0	O=C(O)Cc1ccc(SSc2ccc(CC(=O)O)cc2)cc1	CM	1
1	NNP(=S)(NN)c1ccccc1	CM	1
2	O=Nc1ccc(O)c(N=O)c1O	CM	1
3	Oc1ccc(Cl)cc1C(c1cc(Cl)ccc1O)C(Cl)(Cl)Cl	CM	1
4	NNC(=O)c1ccccc1SSc1ccccc1C(=O)NN	CM	1
...	...	...	...
2938	CC(=O)OC1(C#N)CC2OC1C1C2N1C(=O)OC(C)(C)C	CI	0
2939	CCOC(=O)C1Cc2cc(C)c(C)cc2N(C)C1=O	CI	0
2940	CCOC(=O)N1CCN(c2ccc3c(C)cc(C)nc3n2)CC1	CI	0
2941	CN(C)C=Nc1ccc2c3c(cccc13)-c1ccccc1-2	CI	0
2942	CCC1CC2CC3c4[nH]c5ccc(OC)cc5c4CCN(C2)C13.Cl	CI	0

2943 rows × 3 columns

```
# Randomly shuffle rows
hiv_df_sampled = hiv_df_sampled.sample(frac=1, random_state=42)
```

```
hiv_df_sampled.head()
```

	smiles	activity	HIV_active
240	Cc1cc2c(c(=O)o1)C1=S(SC(c3ccccc3)=C1)S2	CM	1
2325	N#CN1CCC=C(c2cc3ccccc3[nH]2)C1	CI	0
1676	CCC1SC(C)C(=O)NC1=O	CI	0
1952	O=C1CC2(CCN(Cc3ccccc3)CC2)CC(=O)N1	CI	0
677	CC(=O)OC1SC(c2c(F)cccc2F)n2c1nc1ccccc12	CM	1

```
hiv_df_sampled.to_csv('HIV_2.csv', index=False)
# .drop(['activity'], axis=1).
hiv_df_sampled_2 = pd.read_csv("HIV_2.csv")
hiv_df_sampled_2.head()
hiv_df_sampled_2.tail()
```

	smiles	activity	HIV_active
0	Cc1cc2c(c(=O)o1)C1=S(SC(c3ccccc3)=C1)S2	CM	1
1	N#CN1CCC=C(c2cc3ccccc3[nH]2)C1	CI	0
2	CCC1SC(C)C(=O)NC1=O	CI	0
3	O=C1CC2(CCN(Cc3ccccc3)CC2)CC(=O)N1	CI	0
4	CC(=O)OC1SC(c2c(F)cccc2F)n2c1nc1ccccc12	CM	1

	smiles	activity	HIV_active
2938	O=C(CS)Nc1cccc(O)c1	CI	0
2939	O=C(Nc1ccc(N=Nc2ccc(S(=O)(=O)O)cc2)cc1)c1ccc(N...	CM	1
2940	NC(=O)CCN(CCC(N)=O)CCC(N)=O	CI	0
2941	Cn1nc2ccccc2cc1=O	CI	0
2942	C[n+]1c(C=NNC(=O)c2ccc(C(=O)NN=Cc3cn4ccccc4[n+...	CI	0

```
arguments = [
    '--data_path', 'HIV.csv',
    '--dataset_type', 'classification',
    '--save_dir', 'test_checkpoints_multimolecule',
    '--epochs', '30',
    '--save_smiles_splits',
    '--quiet',
    '--batch_size', '64',
    '--ignore_columns', 'activity',
    '--depth', '5',
    '--hidden_size', '300'
]

args = chemprop.args.TrainArgs().parse_args(arguments)


mean_score, std_score = chemprop.train.cross_validate(args=args, train_func=chemprop.train.run_training)
```

```
 82%|████████  | 422/515 [01:38<00:22,  4.18it/s]
 82%|████████  | 424/515 [01:38<00:16,  5.45it/s]
 83%|████████▎ | 425/515 [01:38<00:15,  5.95it/s]
 83%|████████▎ | 426/515 [01:39<00:21,  4.14it/s]
 83%|████████▎ | 427/515 [01:39<00:19,  4.51it/s]
 83%|████████▎ | 428/515 [01:39<00:19,  4.54it/s]
 83%|████████▎ | 429/515 [01:40<00:39,  2.15it/s]
 84%|████████▎ | 431/515 [01:40<00:24,  3.41it/s]
 84%|████████▍ | 432/515 [01:40<00:20,  4.01it/s]
 84%|████████▍ | 433/515 [01:40<00:18,  4.50it/s]
 84%|████████▍ | 434/515 [01:41<00:27,  2.97it/s]
 84%|████████▍ | 435/515 [01:41<00:21,  3.64it/s]
 85%|████████▍ | 436/515 [01:41<00:18,  4.35it/s]
 85%|████████▍ | 437/515 [01:42<00:25,  3.02it/s]
 85%|████████▌ | 439/515 [01:42<00:15,  4.83it/s]
 86%|████████▌ | 441/515 [01:42<00:11,  6.47it/s]
 86%|████████▌ | 443/515 [01:43<00:14,  4.98it/s]
 86%|████████▌ | 444/515 [01:43<00:13,  5.34it/s]
 86%|████████▋ | 445/515 [01:43<00:18,  3.75it/s]
 87%|████████▋ | 447/515 [01:44<00:12,  5.28it/s]
 87%|████████▋ | 449/515 [01:44<00:09,  6.67it/s]
 87%|████████▋ | 450/515 [01:44<00:14,  4.34it/s]
 88%|████████▊ | 452/515 [01:44<00:10,  5.75it/s]
 88%|████████▊ | 453/515 [01:45<00:20,  2.99it/s]
 88%|████████▊ | 455/515 [01:46<00:14,  4.25it/s]
 89%|████████▊ | 457/515 [01:46<00:10,  5.65it/s]
 89%|████████▉ | 459/515 [01:46<00:13,  4.27it/s]
 89%|████████▉ | 460/515 [01:46<00:11,  4.78it/s]
 90%|████████▉ | 461/515 [01:47<00:14,  3.63it/s]
 90%|████████▉ | 463/515 [01:47<00:10,  5.14it/s]
 90%|█████████ | 465/515 [01:47<00:07,  6.62it/s]
 91%|█████████ | 467/515 [01:48<00:09,  5.32it/s]
 91%|█████████ | 469/515 [01:48<00:10,  4.35it/s]
 92%|█████████▏| 472/515 [01:49<00:06,  6.36it/s]
 92%|█████████▏| 474/515 [01:49<00:07,  5.13it/s]
 92%|█████████▏| 476/515 [01:49<00:06,  6.18it/s]
 93%|█████████▎| 477/515 [01:50<00:09,  4.04it/s]
 93%|█████████▎| 479/515 [01:50<00:06,  5.30it/s]
 93%|█████████▎| 481/515 [01:50<00:05,  6.61it/s]
 94%|█████████▍| 483/515 [01:51<00:06,  5.12it/s]
 94%|█████████▍| 484/515 [01:51<00:05,  5.56it/s]
 94%|█████████▍| 485/515 [01:52<00:12,  2.41it/s]
 95%|█████████▍| 487/515 [01:52<00:07,  3.58it/s]
 95%|█████████▍| 489/515 [01:53<00:05,  4.60it/s]
 95%|█████████▌| 491/515 [01:54<00:07,  3.25it/s]
 96%|█████████▌| 492/515 [01:54<00:06,  3.56it/s]
 96%|█████████▌| 493/515 [01:55<00:09,  2.42it/s]
 96%|█████████▌| 495/515 [01:55<00:05,  3.53it/s]
 97%|█████████▋| 497/515 [01:55<00:03,  4.69it/s]
 97%|█████████▋| 498/515 [01:56<00:05,  3.38it/s]
 97%|█████████▋| 499/515 [01:56<00:04,  3.85it/s]
 97%|█████████▋| 500/515 [01:56<00:03,  4.49it/s]
 97%|█████████▋| 501/515 [01:57<00:05,  2.42it/s]
 98%|█████████▊| 506/515 [01:57<00:01,  6.06it/s]
 99%|█████████▉| 509/515 [01:57<00:00,  7.72it/s]
100%|█████████▉| 514/515 [01:57<00:00, 12.06it/s]

  0%|          | 0/65 [00:00<?, ?it/s]
```

```
mean_score, std_score
```

```
(0.7778557998980004, 0.0)
```

```
bp_df = pd.read_csv("BBBP.csv")
bp_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)...

Next steps: View recommended plots

```
bp_df.tail()
```

	num	name	p_np	smiles
2045	2049	licostinel	1	C1=C(Cl)C(=C(C2=C1NC(=O)C(N2)=O)[N+](=O)[O-])Cl
2046	2050	ademetionine(adenosyl-methionine)	1	[C@H]3([N]2C1=C(C(=NC=N1)N)N=C2)[C@@H]([C@@H](...
2047	2051	mesocarb	1	[O+]1=N[N](C=C1[N-]C(NC2=CC=CC=C2)=O)C(CC3=CC=...
[illegible]	[illegible]	[illegible]	[illegible]	C1=C(OC)C(=CC2=C1C(=[N+](C(=C2CC)C)

```
bp_df.drop(['num', 'name', 'p_np'], axis=1).to_csv('BBBP_2.csv', index=False)
```

```
bp_df_2 = pd.read_csv("BBBP_2.csv")
bp_df_2.head()
bp_df_2.tail()
```

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

	smiles
2045	C1=C(Cl)C(=C(C2=C1NC(=O)C(N2)=O)[N+](=O)[O-])Cl
2046	[C@H]3([N]2C1=C(C(=NC=N1)N)N=C2)[C@@H]([C@@H](...
2047	[O+]1=N[N](C=C1[N-]C(NC2=CC=CC=C2)=O)C(CC3=CC=...
2048	C1=C(OC)C(=CC2=C1C(=[N+](C(=C2CC)C)[NH-])C3=CC...
2049	[N+](=NCC(=O)N[C@@H]([C@H](O)C1=CC=C([N+]([O-]...

```
arguments = [
    '--test_path', 'BBBP_2.csv',
    '--preds_path', 'BBBP_preds.csv',
    '--checkpoint_dir', 'test_checkpoints_multimolecule'
]

args = chemprop.args.PredictArgs().parse_args(arguments)
preds = chemprop.train.make_predictions(args=args)
```

```
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:30] WARNING: not removing hydrogen atom without neighbors
[07:04:31] WARNING: not removing hydrogen atom without neighbors
[07:04:31] WARNING: not removing hydrogen atom without neighbors
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:557: UserWarning: This DataLoader will cre
  warnings.warn(_create_warning_msg(
Test size = 2,039
  0%|          | 0/1 [00:00<?, ?it/s]Loading pretrained parameter "encoder.encoder.0.cached_zero_vector".
Loading pretrained parameter "encoder.encoder.0.W_i.weight".
Loading pretrained parameter "encoder.encoder.0.W_h.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.bias".
Loading pretrained parameter "readout.1.weight".
Loading pretrained parameter "readout.1.bias".
Loading pretrained parameter "readout.4.weight".
Loading pretrained parameter "readout.4.bias".
Moving model to cuda

  0%|          | 0/41 [00:00<?, ?it/s]
  2%|▏         | 1/41 [00:02<01:21,  2.04s/it]
 10%|▉         | 4/41 [00:02<00:22,  1.63it/s]
 22%|██▏       | 9/41 [00:03<00:08,  3.57it/s]
 29%|██▉       | 12/41 [00:03<00:06,  4.39it/s]
 37%|███▋      | 15/41 [00:03<00:04,  6.19it/s]
 41%|████▏     | 17/41 [00:04<00:04,  5.47it/s]
 49%|████▉     | 20/41 [00:04<00:03,  6.22it/s]
 56%|█████▌    | 23/41 [00:04<00:02,  8.27it/s]
 61%|██████    | 25/41 [00:05<00:02,  7.12it/s]
 68%|██████▊   | 28/41 [00:05<00:01,  8.08it/s]
 80%|████████  | 33/41 [00:05<00:00, 10.42it/s]
 88%|████████▊ | 36/41 [00:05<00:00, 12.54it/s]
100%|██████████| 1/1 [00:06<00:00,  6.40s/it]Saving predictions to BBBP_preds.csv
Elapsed time = 0:00:07
```

```
bp_preds_df = pd.read_csv("BBBP_preds.csv")
bp_preds_df.head()
```

	smiles	HIV_active
0	[Cl].CC(C)NCC(O)COc1cccc2ccccc12	0.021472735330462456
1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl	0.0014868687139824033
2	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...	0.03692841902375221
3	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C	0.0009061084710992873
4	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...	0.006709508132189512

Next steps: View recommended plots

```
bp_preds_df.tail()
```

	smiles	HIV_active
2045	C1=C(Cl)C(=C(C2=C1NC(=O)C(N2)=O)[N+](=O)[O-])Cl	0.0041548567824065685
2046	[C@H]3([N]2C1=C(C(=NC=N1)N)N=C2)[C@@H]([C@@H](...	0.015147102996706963
2047	[O+]1=N[N](C=C1[N-]C(NC2=CC=CC=C2)=O)C(CC3=CC=...	0.004494475666433573
2048	C1=C(OC)C(=CC2=C1C(=[N+](C(=C2CC)C)[NH-])C3=CC...	0.05315469577908516

```
bp_preds_df.describe()
```

	smiles	HIV_active
count	2050	2050
unique	2050	1994
top	[Cl].CC(C)NCC(O)COc1cccc2ccccc12	Invalid SMILES
freq	1	11

```
bp_preds_df = bp_preds_df[bp_preds_df['HIV_active'] != "Invalid SMILES"]
bp_preds_df.describe()
```

	smiles	HIV_active
count	2039	2039
unique	2039	1993
top	[Cl].CC(C)NCC(O)COc1cccc2ccccc12	0.0003313531633466482
freq	1	3

```
bp_preds_df['HIV_active'] = bp_preds_df['HIV_active'].astype(float)
```

```
bp_preds_df['HIV_active_2'] = bp_preds_df['HIV_active'].apply(lambda x: 1 if x > 0.8 else 0)
bp_preds_df.head()
```

	smiles	HIV_active	HIV_active_2
0	[Cl].CC(C)NCC(O)COc1cccc2ccccc12	0.021473	0
1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl	0.001487	0
2	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...	0.036928	0
3	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C	0.000906	0
4	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...	0.006710	0

Next steps: View recommended plots

```
bp_preds_df.describe()
```

	HIV_active	HIV_active_2
count	2.039000e+03	2039.000000
mean	2.598170e-02	0.001962
std	6.617616e-02	0.044259
min	4.265374e-10	0.000000
25%	3.987846e-03	0.000000
50%	1.050125e-02	0.000000
75%	2.310448e-02	0.000000
max	8.798995e-01	1.000000

```
# Filter rows where 'target_column' is equal to 1
bp_preds_df_filtered = bp_preds_df[bp_preds_df['HIV_active_2'] == 1]
bp_preds_df_filtered
```

1 to 4 of 4 entries Filter

index	smiles	HIV_active	HIV_active_2
11	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=O)NC1=O	0.8460021018981934	1
289	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)N=CNc23	0.8281465768814087	1
319	CC1=CN([C@@H]2O[C@H](CO)C=C2)C(=O)NC1=O	0.8528364300727844	1
346	NC1=NC(=O)N(C=C1)[C@H]2CC[C@@H](CO)O2	0.8798995018005371	1

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots

```
smiles_to_check = bp_preds_df_filtered['smiles'].to_list()
```

```
hiv_df_sampled_2[hiv_df_sampled_2['smiles'].isin(smiles_to_check)]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-97b3985eab54> in <cell line: 1>()
----> 1 hiv_df_sampled_2[hiv_df_sampled_2['smiles'].isin(smiles_to_check)]

NameError: name 'hiv_df_sampled_2' is not defined
```

```
hiv_df[hiv_df['smiles'].isin(smiles_to_check)]
```

smiles	activity	HIV_active

```
bp_df[bp_df['smiles'].isin(smiles_to_check)]
```

	num	name	p_np	smiles
11	12	alovudine	1	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=O)NC1=O
289	291	Didanosine	0	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)N=CNc23
319	321	Stavudine	1	CC1=CN([C@@H]2O[C@H](CO)C=C2)C(=O)NC1=O
346	348	Zalcitabine	1	NC1=NC(=O)N(C=C1)[C@H]2CC[C@@H](CO)O2

```
bp_df_final = pd.merge(bp_df[bp_df['smiles'].isin(smiles_to_check)], bp_preds_df_filtered, on='smiles' )
bp_df_final
```

	num	name	p_np	smiles	HIV_active	HIV_active_2
0	12	alovudine	1	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=O)NC1=O	0.846002	1
1	291	Didanosine	0	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)N=CNc23	0.828147	1
2	321	Stavudine	1	CC1=CN([C@@H]2O[C@H]	0.852836	1

Next steps: View recommended plots

```
bp_df_final.to_csv('HIV_result.csv', index=False)
```

```
sub_df = pd.read_csv("substances.csv")
sub_df.head()
```

	zinc_id	smiles
0	ZINC000000000027	N[C@@H](CCc1ccc(N(CCCl)CCCl)cc1)C(=O)O
1	ZINC000016090786	N[C@H](CCc1ccc(N(CCCl)CCCl)cc1)C(=O)O
2	ZINC000001763088	N[C@H](CCCc1ccc(N(CCCl)CCCl)cc1)C(=O)O
3	ZINC000002033385	N[C@@H](CCCc1ccc(N(CCCl)CCCl)cc1)C(=O)O
4	ZINC000000001673	N[C@@H](Cc1ccc(N(CCCl)CCCl)cc1)C(=O)O

Next steps: View recommended plots

```
sub_df.tail()
```

	zinc_id	smiles
46	ZINC000196349655	O=C(O)CCSc1ccc(N(CCCl)CCCl)cc1
47	ZINC000064454242	N=NCCCc1ccc(N(CCCl)CCCl)cc1
48	ZINC000005161807	O=C(O)C/C=C/c1ccc(N(CCCl)CCCl)cc1
49	ZINC000001682294	O=C(O)CCOc1ccc(N(CCCl)CCCl)cc1
50	ZINC000079564304	O=C(O)CNC(=O)c1ccc(N(CCCl)CCCl)cc1

```
sub_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   zinc_id  51 non-null     object
 1   smiles   51 non-null     object
dtypes: object(2)
memory usage: 944.0+ bytes
```

```
arguments = [
    '--test_path', 'substances.csv',
    '--preds_path', 'substances_preds.csv',
    '--checkpoint_dir', 'test_checkpoints_multimolecule',
    '--smiles_columns', 'smiles'
]

args = chemprop.args.PredictArgs().parse_args(arguments)
preds = chemprop.train.make_predictions(args=args)
```

```
Loading training args
Setting molecule featurization parameters to default.
Loading data
51it [00:00, 39141.72it/s]
100%|██████████| 51/51 [00:00<00:00, 69586.70it/s]
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:557: UserWarning: This DataLoader will cre
  warnings.warn(_create_warning_msg(
Validating SMILES
Test size = 51
  0%|          | 0/1 [00:00<?, ?it/s]Loading pretrained parameter "encoder.encoder.0.cached_zero_vector".
Loading pretrained parameter "encoder.encoder.0.W_i.weight".
Loading pretrained parameter "encoder.encoder.0.W_h.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.bias".
Loading pretrained parameter "readout.1.weight".
Loading pretrained parameter "readout.1.bias".
Loading pretrained parameter "readout.4.weight".
Loading pretrained parameter "readout.4.bias".
Moving model to cuda

  0%|          | 0/2 [00:00<?, ?it/s]
 50%|█████     | 1/2 [00:00<00:00,  2.27it/s]
100%|██████████| 1/1 [00:00<00:00,  1.26it/s]Saving predictions to substances_preds.csv
Elapsed time = 0:00:01
```

```
fda_df = pd.read_csv("fda_approved.csv")
fda_df.head()
```

	zinc_id	smiles
0	ZINC000001530427	C[C@@H]1O[C@@H]1P(=O)(O)O
1	ZINC000003807804	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1
2	ZINC000000120286	Nc1nc(N)c2nc(-c3ccccc3)c(N)nc2n1
3	ZINC000242548690	C[C@H]1O[C@@H](O[C@H]2[C@@H](O)C[C@H](O[C@H]3[...
4	ZINC000000008492	Oc1cccc2cccnc12

Next steps: View recommended plots

```
arguments = [
    '--test_path', 'fda_approved.csv',
    '--preds_path', 'fda_approved_preds.csv',
    '--checkpoint_dir', 'test_checkpoints_multimolecule',
    '--smiles_columns', 'smiles'
]

args = chemprop.args.PredictArgs().parse_args(arguments)
preds = chemprop.train.make_predictions(args=args)
```

```
Loading training args
Setting molecule featurization parameters to default.
Loading data
892it [00:00, 84347.53it/s]
100%|██████████| 892/892 [00:00<00:00, 63923.58it/s]Validating SMILES

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:557: UserWarning: This DataLoader will cre
  warnings.warn(_create_warning_msg(
Test size = 892
  0%|          | 0/1 [00:00<?, ?it/s]Loading pretrained parameter "encoder.encoder.0.cached_zero_vector".
Loading pretrained parameter "encoder.encoder.0.W_i.weight".
Loading pretrained parameter "encoder.encoder.0.W_h.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.weight".
Loading pretrained parameter "encoder.encoder.0.W_o.bias".
Loading pretrained parameter "readout.1.weight".
Loading pretrained parameter "readout.1.bias".
Loading pretrained parameter "readout.4.weight".
Loading pretrained parameter "readout.4.bias".
Moving model to cuda

  0%|          | 0/18 [00:00<?, ?it/s]
  6%|▌         | 1/18 [00:02<00:34,  2.05s/it]
 11%|█         | 2/18 [00:02<00:15,  1.03it/s]
 22%|██▏       | 4/18 [00:02<00:05,  2.46it/s]
 50%|█████     | 9/18 [00:02<00:01,  6.87it/s]
 67%|██████▋   | 12/18 [00:02<00:00,  9.41it/s]
100%|██████████| 1/1 [00:03<00:00,  3.30s/it]Saving predictions to fda_approved_preds.csv
Elapsed time = 0:00:04
```

```
fda_preds_df = pd.read_csv("fda_approved_preds.csv")
fda_preds_df.head()
```

	zinc_id	smiles	HIV_active
0	ZINC000001530427	C[C@@H]1O[C@@H]1P(=O)(O)O	0.000114
1	ZINC000003807804	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	0.021717
2	ZINC000000120286	Nc1nc(N)c2nc(-c3ccccc3)c(N)nc2n1	0.020179
3	ZINC000242548690	C[C@H]1O[C@@H](O[C@H]2[C@@H](O)C[C@H](O[C@H]3[...	0.021397
4	ZINC000000008492	Oc1cccc2cccnc12	0.006810

Next steps: View recommended plots

```
fda_preds_df = fda_preds_df[fda_preds_df['HIV_active'] != "Invalid SMILES"]
fda_preds_df.describe()
fda_preds_df['HIV_active'] = fda_preds_df['HIV_active'].astype(float)
fda_preds_df['HIV_active_2'] = fda_preds_df['HIV_active'].apply(lambda x: 1 if x > 0.8 else 0)
fda_preds_df.head()
```

	HIV_active
count	8.920000e+02
mean	3.120089e-02
std	7.911301e-02
min	2.711503e-08
25%	4.847819e-03
50%	1.207062e-02
75%	2.594892e-02
max	8.708455e-01

	zinc_id	smiles	HIV_active	HIV_active_2
0	ZINC000001530427	C[C@@H]1O[C@@H]1P(=O)(O)O	0.000114	0
1	ZINC000003807804	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	0.021717	0
2	ZINC000000120286	Nc1nc(N)c2nc(-c3ccccc3)c(N)nc2n1	0.020179	0
		C[C@H]1O[C@@H]		

Next steps: View recommended plots View recommended plots

```
# Filter rows where 'target_column' is equal to 1
fda_preds_df_filtered = fda_preds_df[fda_preds_df['HIV_active_2'] == 1]
fda_preds_df_filtered
```

	zinc_id	smiles	HIV_active	HIV_active_
321	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1	0.870845	
819	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H]	0.844080	

Next steps: View recommended plots

```
smiles_to_check = fda_preds_df_filtered['smiles'].to_list()
print(f"smiles to check: {smiles_to_check}")
```

```
smiles to check: ['O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1', 'Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O']
```

```
hiv_df_sampled_2[hiv_df_sampled_2['smiles'].isin(smiles_to_check)]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-47-97b3985eab54> in <cell line: 1>()
----> 1 hiv_df_sampled_2[hiv_df_sampled_2['smiles'].isin(smiles_to_check)]

NameError: name 'hiv_df_sampled_2' is not defined
```

```
hiv_df[hiv_df['smiles'].isin(smiles_to_check)]
```

smiles	activity	HIV_active

```
bp_df[bp_df['smiles'].isin(smiles_to_check)]
```

num	name	p_np	smiles

```
fda_df[fda_df['smiles'].isin(smiles_to_check)]
```

	zinc_id	smiles
321	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1
819	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O

```
fda_df_final = pd.merge(fda_df[fda_df['smiles'].isin(smiles_to_check)], fda_preds_df_filtered, on='smiles' )
fda_df_final
```

	zinc_id_x	smiles	zinc_id_y	HIV_acti
0	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1	ZINC000013597823	0.870
1	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H]	ZINC000000137884	0.844

Next steps: View recommended plots

```
fda_df_final.to_csv('fda_approved_result.csv', index=False)
```

```
# !wget https://zinc15.docking.org/substances/subsets/named.csv
```

```
--2024-03-10 05:58:28--  https://zinc15.docking.org/substances/subsets/named.csv
Resolving zinc15.docking.org (zinc15.docking.org)... 169.230.75.4
Connecting to zinc15.docking.org (zinc15.docking.org)|169.230.75.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘named.csv.1’

named.csv.1             [ <=>                ]   9.28K  --.-KB/s    in 0.04s

2024-03-10 05:58:29 (242 KB/s) - ‘named.csv.1’ saved [9499]
```

```
zinc_df = pd.read_csv("named.csv")
zinc_df.head()
zinc_df.tail()
```

	zinc_id	smiles
0	ZINC000030727788	C=C[C@]1(C)C[C@@H](OC(=O)CSC(C)(C)CNC(=O)[C@H]...
1	ZINC000150377216	CCCCCC/C=C\C/C=C\CCCCCCCC(=O)OC[C@H](COCCCCCCC...
2	ZINC000100780125	CC(=O)O[C@H]1C[C@](C)(O)[C@@H]2CC=C(C)[C@@H]2[...
3	ZINC000006580536	O=C(O)[C@H](Cc1ccccc1)N(CCCl)CCCl
4	ZINC000150351802	O=C1C[C@H](c2ccc(O)c(O)c2)Oc2c1c(O)cc(O[C@H]1O...

	zinc_id	smiles
22959	ZINC000015253718	C/C(=C\CO)CC/C=C(\C)CCC[C@@H](C)CCC[C@H](C)CCC...
22960	ZINC000043888360	CCCCCCCCCCCCCCCCCCCCC[C@@H](O)C[C@@H](O)CCCCC
22961	ZINC000096006009	Cc1noc(NS(=O)(=O)c2ccc(N)cc2)c1C
22962	ZINC000150375318	CC/C=C\C/C=C\C/C=C\CCCCCCCC(=O)OC[C@@H](COC(=O...

```
arguments = [
    '--test_path', 'named.csv',
    '--preds_path', 'named_preds.csv',
    '--checkpoint_dir', 'test_checkpoints_multimolecule',
    '--smiles_columns', 'smiles'
]

args = chemprop.args.PredictArgs().parse_args(arguments)
preds = chemprop.train.make_predictions(args=args)
```

```
 85%|████████▍ | 585/692 [02:15<00:24,  4.37it/s]
 85%|████████▍ | 587/692 [02:15<00:23,  4.48it/s]
 85%|████████▌ | 589/692 [02:17<00:34,  3.00it/s]
 86%|████████▌ | 593/692 [02:17<00:20,  4.92it/s]
 86%|████████▌ | 595/692 [02:17<00:20,  4.75it/s]
 86%|████████▋ | 597/692 [02:18<00:25,  3.68it/s]
 87%|████████▋ | 601/692 [02:18<00:15,  5.85it/s]
 87%|████████▋ | 603/692 [02:19<00:16,  5.29it/s]
 87%|████████▋ | 605/692 [02:20<00:22,  3.92it/s]
 88%|████████▊ | 609/692 [02:20<00:13,  6.19it/s]
 88%|████████▊ | 611/692 [02:20<00:14,  5.53it/s]
 89%|████████▊ | 613/692 [02:21<00:19,  4.00it/s]
 89%|████████▉ | 617/692 [02:21<00:12,  6.24it/s]
 89%|████████▉ | 619/692 [02:22<00:12,  5.82it/s]
 90%|████████▉ | 621/692 [02:23<00:16,  4.19it/s]
 90%|█████████ | 625/692 [02:23<00:10,  6.47it/s]
 91%|█████████ | 627/692 [02:24<00:15,  4.19it/s]
 91%|█████████ | 629/692 [02:24<00:15,  4.11it/s]
 92%|█████████▏| 634/692 [02:24<00:08,  7.02it/s]
 92%|█████████▏| 636/692 [02:25<00:10,  5.36it/s]
 92%|█████████▏| 638/692 [02:27<00:16,  3.19it/s]
 93%|█████████▎| 641/692 [02:27<00:11,  4.49it/s]
 93%|█████████▎| 643/692 [02:27<00:12,  4.03it/s]
 93%|█████████▎| 645/692 [02:28<00:14,  3.29it/s]
 94%|█████████▎| 648/692 [02:28<00:09,  4.79it/s]
 94%|█████████▍| 651/692 [02:29<00:07,  5.22it/s]
 94%|█████████▍| 653/692 [02:30<00:09,  3.93it/s]
 95%|█████████▍| 657/692 [02:30<00:05,  6.19it/s]
 95%|█████████▌| 659/692 [02:31<00:08,  4.07it/s]
 96%|█████████▌| 661/692 [02:31<00:07,  4.02it/s]
 96%|█████████▌| 664/692 [02:32<00:04,  5.65it/s]
 96%|█████████▌| 666/692 [02:32<00:03,  6.72it/s]
 97%|█████████▋| 668/692 [02:32<00:03,  6.06it/s]
 97%|█████████▋| 670/692 [02:33<00:05,  4.09it/s]
 97%|█████████▋| 673/692 [02:33<00:03,  5.98it/s]
 98%|█████████▊| 675/692 [02:34<00:03,  5.19it/s]
 98%|█████████▊| 677/692 [02:35<00:03,  3.93it/s]
 99%|█████████▊| 683/692 [02:35<00:01,  7.59it/s]
 99%|█████████▉| 685/692 [02:35<00:00,  8.15it/s]
100%|█████████▉| 691/692 [02:35<00:00, 10.32it/s]
100%|██████████| 1/1 [02:36<00:00, 156.28s/it]
Saving predictions to named_preds.csv
Elapsed time = 0:02:50
```

```python
zinc_preds_df = pd.read_csv("named_preds.csv")
zinc_preds_df.head()
zinc_preds_df = zinc_preds_df[zinc_preds_df['HIV_active'] != "Invalid SMILES"]
zinc_preds_df.describe()
zinc_preds_df['HIV_active'] = zinc_preds_df['HIV_active'].astype(float)
zinc_preds_df['HIV_active_2'] = zinc_preds_df['HIV_active'].apply(lambda x: 1 if x > 0.8 else 0)
zinc_preds_df.head()
```

	zinc_id	smiles	HIV_active
0	ZINC000030727788	C=C[C@]1(C)C[C@@H](OC(=O)CSC(C)(C)CNC(=O)[C@H]...	0.034702
1	ZINC000150377216	CCCCCC/C=C\C/C=C\CCCCCCCC(=O)OC[C@H](COCCCCCCC...	0.021076

Next steps: View recommended plots View recommended plots View recommended plots

```
# Filter rows where 'target_column' is equal to 1
zinc_preds_df_filtered = zinc_preds_df[zinc_preds_df['HIV_active_2'] == 1]
zinc_preds_df_filtered
```

1 to 18 of 18 entries Filter

index	zinc_id	smiles	HIV_active
486	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O	0.84408873319625
664	ZINC000000039906	Nc1ccn([C@H]2CC[C@@H](CO)O2)c(=O)n1	0.87989950180053
1673	ZINC000005551645	CC(C)=CCN1Cc2c(Cl)ccc3[nH]c(=S)n(c23)C[C@@H]1C	0.85742247104644
2491	ZINC000001611085	Nc1nc(=O)n([C@H]2CC[C@H](CO)O2)cc1F	0.89037722349166
7718	ZINC000043313038	CCCCCCCCCCCCSC[C@@H](CO[P@@](=O)(O)OC[C@H]1O[C@@H](n2cc(C)c(=O)[nH]c2=O)C[C@@H]1F)OCCCCCCCCCC	0.83557295799255
8880	ZINC000003809864	CCCc1cc(=O)oc2c3c(c4c(c12)OC(C)(C)CC4)O[C@@H](C)[C@H](C)[C@@H]3O	0.84388917684555
11356	ZINC000016952419	Nc1nc(=O)n([C@H]2CC[C@@H](CO)O2)cc1F	0.89433723688125
13087	ZINC000032016993	Nc1ccn([C@@H]2CC[C@H](CO[P@@](=O)(O)O[P@](=O)(O)OP(=O)(O)O)O2)c(=O)n1	0.86944174766540
15124	ZINC000015042997	CC[C@H]1CC[C@@H](CC)O1	0.8721865415573
15334	ZINC000013516800	Nc1ccn([C@H]2CC[C@@H](CO[P@@](=O)(O)O[P@@](=O)(O)OP(=O)(O)O)O2)c(=O)n1	0.8677484989166
16695	ZINC000003870291	CCCc1cc(=O)oc2c3c(c4c(c12)OC(C)(C)C=C4)O[C@@H](C)[C@@H](C)[C@@H]3O	0.8273838162422
17320	ZINC000012503817	Cc1cn([C@H]2C[C@H](N=[N+]=[N-])[C@@H](COP(=O)(O)O)O2)c(=O)[nH]c1=O	0.91335463523864
20161	ZINC000012502287	Cc1cn([C@H]2C=C[C@@H](CO[P@](=O)(O)OP(=O)(O)O)O2)c(=O)[nH]c1=O	0.87298274040222
24565	ZINC000034932069	Cc1cn([C@H]2C[C@@H](N=[N+]=[N-])[C@H](CO[P@@](=O)(O)OP(=O)(O)O)O2)c(=O)[nH]c1=O	0.90486627817153
25932	ZINC000017175432	Nc1nc(=O)n([C@@H]2CC[C@@H](CO)O2)cc1F	0.8963612318038
26697	ZINC000027871222	CC(=O)N[C@@H](Cc1ccc(O[P@](=O)(O)OC[C@H]2O[C@@H](n3cc(C)c(=O)	0.84106647968292

Next steps: View recommended plots

```
zinc_preds_df_filtered.to_csv('zinc_final_result.csv', index=False)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!mkdir '/content/drive/My Drive/Chemprop_Backup_HIV_ALL_DATA/'
```

```
!pwd
```

```
/content
```

```
!ls -al
```

```
total 9612
drwxr-xr-x 1 root root    4096 Mar 11 07:15 .
drwxr-xr-x 1 root root    4096 Mar 11 05:35 ..
-rw-r--r-- 1 root root  107579 Mar 11 07:04 BBBP_2.csv
```