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)
```

```
Sun Mar 10 19:50:49 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   51C    P8              10W /  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 22.2 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

Next steps: View recommended plots

```
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

Next steps: View recommended plots

```
# 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

Next steps: View recommended plots

```
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

Next steps: View recommended plots

```
# 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

Next steps: View recommended plots

```
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_2.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)
```

```
 78%|███████▊  | 29/37 [00:02<00:00, 10.27it/s]
 84%|████████▍ | 31/37 [00:03<00:00, 10.35it/s]
 89%|████████▉ | 33/37 [00:03<00:00, 10.40it/s]
 95%|█████████▍| 35/37 [00:03<00:00, 10.11it/s]
100%|██████████| 37/37 [00:03<00:00,  9.64it/s]

  0%|          | 0/5 [00:00<?, ?it/s]
 20%|██        | 1/5 [00:00<00:00,  5.84it/s]
 40%|████      | 2/5 [00:00<00:00,  5.72it/s]
 60%|██████    | 3/5 [00:00<00:00,  5.86it/s]
 80%|████████  | 4/5 [00:00<00:00,  6.10it/s]
100%|██████████| 5/5 [00:00<00:00,  6.85it/s]
 67%|██████▋   | 20/30 [01:39<00:45,  4.60s/it]
  0%|          | 0/37 [00:00<?, ?it/s]
  3%|▎         | 1/37 [00:00<00:06,  5.50it/s]
  5%|▌         | 2/37 [00:00<00:06,  5.45it/s]
  8%|▊         | 3/37 [00:00<00:06,  5.31it/s]
 11%|█         | 4/37 [00:00<00:06,  5.11it/s]
 14%|█▎        | 5/37 [00:00<00:06,  5.18it/s]
 16%|█▌        | 6/37 [00:01<00:05,  5.37it/s]
 19%|█▉        | 7/37 [00:01<00:05,  5.43it/s]
 22%|██▏       | 8/37 [00:01<00:04,  6.31it/s]
 24%|██▍       | 9/37 [00:01<00:03,  7.06it/s]
 27%|██▋       | 10/37 [00:01<00:03,  7.65it/s]
 32%|███▏      | 12/37 [00:01<00:02,  8.84it/s]
 35%|███▌      | 13/37 [00:01<00:02,  9.05it/s]
 38%|███▊      | 14/37 [00:02<00:02,  8.79it/s]
 41%|████      | 15/37 [00:02<00:02,  8.78it/s]
 43%|████▎     | 16/37 [00:02<00:02,  8.61it/s]
 46%|████▌     | 17/37 [00:02<00:02,  8.91it/s]
 51%|█████▏    | 19/37 [00:02<00:01,  9.61it/s]
 54%|█████▍    | 20/37 [00:02<00:01,  9.58it/s]
 59%|█████▉    | 22/37 [00:02<00:01, 10.09it/s]
 62%|██████▏   | 23/37 [00:02<00:01,  9.95it/s]
 68%|██████▊   | 25/37 [00:03<00:01, 10.05it/s]
 70%|███████   | 26/37 [00:03<00:01,  9.66it/s]
 73%|███████▎  | 27/37 [00:03<00:01,  9.39it/s]
 76%|███████▌  | 28/37 [00:03<00:00,  9.44it/s]
 78%|███████▊  | 29/37 [00:03<00:00,  9.47it/s]
 84%|████████▍ | 31/37 [00:04<00:01,  5.16it/s]
 86%|████████▋ | 32/37 [00:04<00:00,  5.70it/s]
 92%|█████████▏| 34/37 [00:04<00:00,  6.98it/s]
 95%|█████████▍| 35/37 [00:04<00:00,  7.32it/s]
 97%|█████████▋| 36/37 [00:04<00:00,  7.54it/s]

  0%|          | 0/5 [00:00<?, ?it/s]
 20%|██        | 1/5 [00:00<00:00,  9.60it/s]
 60%|██████    | 3/5 [00:00<00:00, 10.74it/s]
100%|██████████| 5/5 [00:00<00:00, 11.80it/s]
 70%|███████   | 21/30 [01:45<00:43,  4.81s/it]
  0%|          | 0/37 [00:00<?, ?it/s]
  3%|▎         | 1/37 [00:00<00:03,  9.65it/s]
  5%|▌         | 2/37 [00:00<00:03,  9.30it/s]
  8%|▊         | 3/37 [00:00<00:03,  9.40it/s]
 11%|█         | 4/37 [00:00<00:03,  9.40it/s]
 14%|█▎        | 5/37 [00:00<00:03,  9.30it/s]
 19%|█▉        | 7/37 [00:00<00:03,  9.91it/s]
 22%|██▏       | 8/37 [00:00<00:02,  9.79it/s]
```

```
mean_score, std_score
```

```
(0.8328736900165471, 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=...
				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)
```

```
[19:56:48] Explicit valence for atom # 1 N, 4, is greater than permitted
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] Explicit valence for atom # 6 N, 4, is greater than permitted
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
```

```
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] Explicit valence for atom # 11 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 12 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] Explicit valence for atom # 5 N, 4, is greater than permitted
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:48] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] WARNING: not removing hydrogen atom without neighbors
[19:56:49] 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".
```

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

	smiles	HIV_active
0	[Cl].CC(C)NCC(O)COc1cccc2ccccc12	0.07791923731565475
1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl	0.052813779562711716
2	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...	0.5678612589836121
3	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C	0.0569111704826355
4	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...	0.4187469184398651

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	0.1297084391117096
2046	[C@H]3([N]2C1=C(C(=NC=N1)N)N=C2)[C@@H]([C@@H](...	0.23322035372257233
2047	[O+]1=N[N](C=C1[N-]C(NC2=CC=CC=C2)=O)C(CC3=CC=...	0.3322729766368866
2048	C1=C(OC)C(=CC2=C1C(=[N+](C(=C2CC)C)[NH-])C3=CC...	0.3129490911960602
2049	[N+](=NCC(=O)N[C@@H]([C@H](O)C1=CC=C([N+]([O-]...	0.2837357521057129

```
bp_preds_df.describe()
```

	smiles	HIV_active
count	2050	2050
unique	2050	2004
top	[N+](=NCC(=O)N[C@@H]([C@H](O)C1=CC=C([N+]([O-]...	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	2003
top	[N+](=NCC(=O)N[C@@H]([C@H](O)C1=CC=C([N+]([O-]...	0.0426582507789135
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	0.077919	0
1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl	0.052814	0
2	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...	0.567861	0
3	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C	0.056911	0
4	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...	0.418747	0

Next steps: View recommended plots

```
bp_preds_df.describe()
```

	HIV_active	HIV_active_2
count	2039.000000	2039.000000
mean	0.263883	0.009809
std	0.201033	0.098576
min	0.000762	0.000000
25%	0.097246	0.000000
50%	0.208566	0.000000
75%	0.408756	0.000000
max	0.988302	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 20 of 20 entries Filter

index	smiles	HIV_active	H
11	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=O)NC1=O	0.9441607594490051	
235	OC[C@H]1O[C@H](C[C@@H]1O)N2C=C(F)C(=O)NC2=O	0.8643155097961426	
270	Cc1nnc2CN=C(c3ccccc3)c4cc(Cl)ccc4n12	0.8021654486656189	
279	OC[C@H]1O[C@@H](OC2=C(Oc3cc(O)cc(O)c3C2=O)c4ccc(O)c(O)c4)[C@H](O)[C@@H](O)[C@@H]1O	0.8800746202468872	
289	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)N=CNc23	0.8122326731681824	
319	CC1=CN([C@@H]2O[C@H](CO)C=C2)C(=O)NC1=O	0.9302963614463806	
445	CC1=CN([C@H]2C[C@H](N=[N+]=[N-])[C@@H](CO)O2)C(=O)NC1=O	0.9854286313056946	
540	Cc1nnc2CN=C(c3ccccc3Cl)c4cc(Cl)ccc4n12	0.8569784164428711	
581	CN1CCC23C4CCC(O)C2Oc5c(O)ccc(CC14)c35	0.816520631313324	
711	Cc1ncc2CN=C(c3ccccc3F)c4cc(Cl)ccc4n12.OC(=O)\C=C/C(O)=O	0.8424991369247437	
771	Cn1nnnc1SCC2=C(N3[C@H](SC2)[C@H](NC(=O)CS\C=C/C#N)C3=O)C(O)=O	0.808535635471344	
785	CC1=C(N2[C@H](SC1)[C@H](NC(=O)Cc3ccc(cc3)C4=NCCCN4)C2=O)C(O)=O	0.8538467288017273	
912	CO[C@H]1/C=C/O[C@@]2(C)Oc3c(C)c(O)c4c(O)c(NC(=O)C(=C\C=C\[C@H](C)[C@H](O)[C@@H](C)[C@@H](O)[C@@H](C)[C@H](OC(C)=O)[C@@H]1C)/C)cc(O)c4c3C2=O	0.8292530179023743	
945	Oc1ccc(cc1)/C=C([N+]#[C-])/C(=C/c2ccc(O)cc2)[N+]#[C-]	0.8675630688667297	
973	COC1(C(F)(F)C1(F)Cl)F	0.9545143246650696	
1052	C1=C(Br)SC2=C1C(=NCC3=NN=C([N]23)C)C4=C(C=CC=C4)Cl	0.8791234493255615	
1086	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C4CCCCC4)C5=CC=CC=C5Cl)Br	0.8506717085838318	
1365	[C@@H]25CC1=CC=C(C4=C1[C@@]3([C@H]2CC[C@@H]([C@@H]3O4)OC)CCN5C)O	0.8105615973472595	
1822	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C)C4=CC=CC=C4Cl)CC	0.8752243518829346	
2044	[N+](=[N-])=O	0.9883024096488953	

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)]
```

smiles	activity	HIV_active

```
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	
11	12	alovudine	1	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=(
235	237	floxuridine	0	OC[C@H]1O[C@H](C[C@@H]1O)N2C=C(F)C(=(
270	272	Alprazolam	1	Cc1nnc2CN=C(c3ccccc3)c4cc(C
279	281	Isoquercitrin	0	OC[C@H]1O[C@@H](OC2=C(Oc3cc(O)cc(O)c3C2=
289	291	Didanosine	0	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)I
319	321	Stavudine	1	CC1=CN([C@@H]2O[C@H](CO)C=C2)C(=(
445	447	zidovudine	0	CC1=CN([C@H]2C[C@H](N=[N+]=[N-])[C@@H](C(
540	542	Triazolam	1	Cc1nnc2CN=C(c3ccccc3Cl)c4cc(C
581	583	dihydromorphine	1	CN1CCC23C4CCC(O)C2Oc5c(O)ccc((
711	713	midazolam maleate	1	Cc1ncc2CN=C(c3ccccc3F)c4cc(Cl)ccc4n12.OC
771	773	cefivitril	0	Cn1nnnc1SCC2=C(N3[C@H](SC2)[C@H](NC(=O)C
785	787	cefrotil	0	CC1=C(N2[C@H](SC1)[C@H](NC(=O)Cc3ccc(cc3)
912	914	rifamycin	0	CO[C@H]1/C=C/O[C@@]2(C)Oc3c(C)c(O)c4c(O)
945	947	xantocillin	0	Oc1ccc(cc1)/C=C([N+]#[C-])/C(=C/c2ccc(C
973	975	aliflurane	1	COC1(C(F)(F)(
1052	1054	brotizolam	1	C1=C(Br)SC2=C1C(=NCC3=NN=C([N]23)C)C4=C(C
1086	1089	ciclotizolam	1	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C4CCCCC4)C5
1365	1369	methyldihydromorphine	1	[C@@H]25CC1=CC=C(C4=C1[C@@]3([C@H]2CC
1822	1826	etizolam	1	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C)C4=CC=CC=
2044	2048	nitrous-oxide	1	[N+]

```
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	s
0	12	alovudine	1	CC1=CN([C@H]2C[C@H](F)[C@@H](CO)O2)C(=O)N
1	237	floxuridine	0	OC[C@H]1O[C@H](C[C@@H]1O)N2C=C(F)C(=O)N
2	272	Alprazolam	1	Cc1nnc2CN=C(c3ccccc3)c4cc(Cl)c
3	281	Isoquercitrin	0	OC[C@H]1O[C@@H](OC2=C(Oc3cc(O)cc(O)c3C2=O)
4	291	Didanosine	0	OC[C@@H]1CC[C@@H](O1)n2cnc3C(=O)N=
5	321	Stavudine	1	CC1=CN([C@@H]2O[C@H](CO)C=C2)C(=O)N
6	447	zidovudine	0	CC1=CN([C@H]2C[C@H](N=[N+]=[N-])[C@@H](CO)(
7	542	Triazolam	1	Cc1nnc2CN=C(c3ccccc3Cl)c4cc(Cl)c
8	583	dihydromorphine	1	CN1CCC23C4CCC(O)C2Oc5c(O)ccc(CC
9	713	midazolam maleate	1	Cc1ncc2CN=C(c3ccccc3F)c4cc(Cl)ccc4n12.OC(=
10	773	cefivitril	0	Cn1nnnc1SCC2=C(N3[C@H](SC2)[C@H](NC(=O)CS\
11	787	cefrotil	0	CC1=C(N2[C@H](SC1)[C@H](NC(=O)Cc3ccc(cc3)C4
12	914	rifamycin	0	CO[C@H]1/C=C/O[C@@]2(C)Oc3c(C)c(O)c4c(O)c(
13	947	xantocillin	0	Oc1ccc(cc1)/C=C([N+]#[C-])/C(=C/c2ccc(O)c
14	975	aliflurane	1	COC1(C(F)(F)C1
15	1054	brotizolam	1	C1=C(Br)SC2=C1C(=NCC3=NN=C([N]23)C)C4=C(C=C
16	1089	ciclotizolam	1	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C4CCCCC4)C5=C
17	1369	methyldihydromorphine	1	[C@@H]25CC1=CC=C(C4=C1[C@@]3([C@H]2CC[C
18	1826	etizolam	1	C1=C(SC2=C1C(=NCC3=NN=C([N]23)C)C4=CC=CC=C4
19	2048	nitrous-oxide	1	[N+](=[

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, 60426.41it/s]
100%|██████████| 51/51 [00:00<00:00, 76780.15it/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,  2.54it/s]
100%|██████████| 1/1 [00:01<00:00,  1.20s/it]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, 90156.61it/s]
    100%|██████████| 892/892 [00:00<00:00, 51982.98it/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 = 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<00:48,  2.86s/it]
     11%|█         | 2/18 [00:03<00:20,  1.28s/it]
     50%|█████     | 9/18 [00:03<00:01,  4.71it/s]
     78%|███████▊  | 14/18 [00:03<00:00,  8.14it/s]
    100%|██████████| 1/1 [00:03<00:00,  3.84s/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	0.085422
1	ZINC000003807804	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	0.303771
2	ZINC000000120286	Nc1nc(N)c2nc(-c3ccccc3)c(N)nc2n1	0.093442
3	ZINC000242548690	C[C@H]1O[C@@H](O[C@H]2[C@@H](O)C[C@H](O[C@H]3[...	0.565260
4	ZINC000000008492	Oc1cccc2cccnc12	0.089892

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	892.000000
mean	0.257912
std	0.204399
min	0.003855
25%	0.097084
50%	0.190803
75%	0.389287
max	0.992705

	zinc_id	smiles	HIV_active	HIV_active_2
0	ZINC000001530427	C[C@@H]1O[C@@H]1P(=O)(O)O	0.085422	0
1	ZINC000003807804	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	0.303771	0
2	ZINC000000120286	Nc1nc(N)c2nc(-c3ccccc3)c(N)nc2n1	0.093442	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
```

1 to 18 of 18 entries Filter

index	zinc_id	smiles	HIV_acti
31	ZINC000003816287	CNC(=O)c1ccccc1Sc1ccc2c(/C=C/c3ccccn3)n[nH]c2c1	0.8191019296
47	ZINC000003813010	O=c1[nH]c(=O)n([C@H]2C[C@H](O)[C@@H](CO)O2)cc1F	0.8848749995
81	ZINC000003818726	O=C(/C=C/c1cccc(S(=O)(=O)Nc2ccccc2)c1)NO	0.8046247959
94	ZINC000068153186	CC(C)(C)c1nc(-c2cccc(NS(=O)(=O)c3c(F)cccc3F)c2F)c(-c2ccnc(N)n2)s1	0.9387590289
197	ZINC000000005423	Cc1nc(-c2ccc(OCC(C)C)c(C#N)c2)sc1C(=O)O	0.8545335531
276	ZINC000000002212	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1Cl)=NC2	0.8569784164
321	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1	0.851600170
340	ZINC000001530621	CCN[C@H]1C[C@H](C)S(=O)(=O)c2sc(S(N)(=O)=O)cc21	0.824345052
499	ZINC000000896717	COc1cc(/C(O)=N/S(=O)(=O)c2ccccc2C)ccc1Cc1cn(C)c2ccc(NC(=O)OC3CCCC3)cc12	0.8092177510
540	ZINC000004474564	CC/C=C\C/C=C\C/C=C\C/C=C\C/C=C\C/C=C\CCC(=O)O	0.824715793
542	ZINC000005733652	COc1ccc(-c2cc(=O)c3c(O)cc(O)cc3o2)cc1O	0.8031748533
626	ZINC000003830993	N[C@@H](Cc1cc(I)c(Oc2cc(I)c(O)c(I)c2)c(I)c1)C(=O)O	0.8241854906
715	ZINC000169289767	Cc1cc(-c2ccc(/N=N/c3c(S(=O)(=O)O)cc4cc(S(=O)(=O)O)cc(N)c4c3O)c(C)c2)ccc1/N=N/c1c(S(=O)(=O)O)cc2cc(S(=O)(=O)O)cc(N)c2c1O	0.9927045702
727	ZINC000003807172	C[C@H]1CNc2c(cccc2S(=O)(=O)N[C@@H](CCCNC(=N)N)C(=O)N2CC[C@@H](C)C[C@@H]2C(=O)O)C1	0.8672307729
806	ZINC000000000903	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2	0.8021654486
819	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O	0.9346132278
821	ZINC000000897244	CC1(C)[C@H](C(=O)O)N2C(=O)C[C@H]2S1(=O)=O	0.8134081363
879	ZINC000003779042	Cc1cn([C@H]2C[C@H](N=[N+]=[N-])[C@@H](CO)O2)c(=O)[nH]c1=O	0.9854286313

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: ['CNC(=O)c1ccccc1Sc1ccc2c(/C=C/c3ccccn3)n[nH]c2c1', 'O=c1[nH]c(=O)n([C@H]2C[C@H](O)[C@@H](CO)O2)
```

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

	smiles	activity	HIV_active

```
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
31	ZINC000003816287	CNC(=O)c1ccccc1Sc1ccc2c(/C=C/c3ccccn3)n[nH]c2c1
47	ZINC000003813010	O=c1[nH]c(=O)n([C@H]2C[C@H](O)[C@@H](CO)O2)cc1F
81	ZINC000003818726	O=C(/C=C/c1cccc(S(=O)(=O)Nc2ccccc2)c1)NO
94	ZINC000068153186	CC(C)(C)c1nc(-c2cccc(NS(=O)(=O)c3c(F)cccc3F)c2...
197	ZINC000000005423	Cc1nc(-c2ccc(OCC(C)C)c(C#N)c2)sc1C(=O)O
276	ZINC000000002212	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1Cl)=NC2
321	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1
340	ZINC000001530621	CCN[C@H]1C[C@H](C)S(=O)(=O)c2sc(S(N)(=O)=O)cc21
499	ZINC000000896717	COc1cc(/C(O)=N/S(=O)(=O)c2ccccc2C)ccc1Cc1cn(C)...
540	ZINC000004474564	CC/C=C\C/C=C\C/C=C\C/C=C\C/C=C\C/C=C\CCC(=O)O
542	ZINC000005733652	COc1ccc(-c2cc(=O)c3c(O)cc(O)cc3o2)cc1O
626	ZINC000003830993	N[C@@H](Cc1cc(I)c(Oc2cc(I)c(O)c(I)c2)c(I)c1)C(...
715	ZINC000169289767	Cc1cc(-c2ccc(/N=N/c3c(S(=O)(=O)O)cc4cc(S(=O)(=...
727	ZINC000003807172	C[C@H]1CNc2c(cccc2S(=O)(=O)N[C@@H](CCCNC(=N)N)...
806	ZINC000000000903	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2
819	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O
821	ZINC000000897244	CC1(C)[C@H](C(=O)O)N2C(=O)C[C@H]2S1(=O)=O
879	ZINC000003779042	Cc1cn([C@H]2C[C@H](N=[N+]=[N-])[C@@H](CO)O2)c(...

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

1 to 18 of 18 entries Filter

index	zinc_id_x	smiles	zinc_id_y
0	ZINC000003816287	CNC(=O)c1ccccc1Sc1ccc2c(/C=C/c3ccccn3)n[nH]c2c1	ZINC0000038
1	ZINC000003813010	O=c1[nH]c(=O)n([C@H]2C[C@H](O)[C@@H](CO)O2)cc1F	ZINC0000038
2	ZINC000003818726	O=C(/C=C/c1cccc(S(=O)(=O)Nc2ccccc2)c1)NO	ZINC0000038
3	ZINC000068153186	CC(C)(C)c1nc(-c2cccc(NS(=O)(=O)c3c(F)cccc3F)c2F)c(-c2ccnc(N)n2)s1	ZINC0000681
4	ZINC000000005423	Cc1nc(-c2ccc(OCC(C)C)c(C#N)c2)sc1C(=O)O	ZINC0000000
5	ZINC000000002212	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1Cl)=NC2	ZINC0000000
6	ZINC000013597823	O=c1[nH]cnc2c1ncn2[C@H]1CC[C@@H](CO)O1	ZINC0000135
7	ZINC000001530621	CCN[C@H]1C[C@H](C)S(=O)(=O)c2sc(S(N)(=O)=O)cc21	ZINC0000015
8	ZINC000000896717	COc1cc(/C(O)=N/S(=O)(=O)c2ccccc2C)ccc1Cc1cn(C)c2ccc(NC(=O)OC3CCCC3)cc12	ZINC0000008
9	ZINC000004474564	CC/C=C\C/C=C\C/C=C\C/C=C\C/C=C\C/C=C\CCC(=O)O	ZINC0000044
10	ZINC000005733652	COc1ccc(-c2cc(=O)c3c(O)cc(O)cc3o2)cc1O	ZINC0000057
11	ZINC000003830993	N[C@@H](Cc1cc(I)c(Oc2cc(I)c(O)c(I)c2)c(I)c1)C(=O)O	ZINC0000038
12	ZINC000169289767	Cc1cc(-c2ccc(/N=N/c3c(S(=O)(=O)O)cc4cc(S(=O)(=O)O)cc(N)c4c3O)c(C)c2)ccc1/N=N/c1c(S(=O)(=O)O)cc2cc(S(=O)(=O)O)cc(N)c2c1O	ZINC0001692
13	ZINC000003807172	C[C@H]1CNc2c(cccc2S(=O)(=O)N[C@@H](CCCNC(=N)N)C(=O)N2CC[C@@H](C)C[C@@H]2C(=O)O)C1	ZINC0000038
14	ZINC000000000903	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2	ZINC0000000
15	ZINC000000137884	Cc1cn([C@H]2C=C[C@@H](CO)O2)c(=O)[nH]c1=O	ZINC0000001
16	ZINC000000897244	CC1(C)[C@H](C(=O)O)N2C(=O)C[C@H]2S1(=O)=O	ZINC0000008
17	ZINC000003779042	Cc1cn([C@H]2C[C@H](N=[N+]=[N-])[C@@H](CO)O2)c(=O)[nH]c1=O	ZINC0000037

Show 25 per page

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
34595	ZINC000005999135	COc1cc([C@@H]2Oc3c(OC)cc(/C=C/CO)cc3[C@H]2CO)c...
34596	ZINC000084710404	COC(=O)c1cc(OC)c2cc(OC)c(OC)c(O)c2c1O
34597	ZINC000150369761	CC/C=C\C/C=C\C/C=C\C/C=C\C/C=C\C/C=C\C/C=C\CCC(=O)O[...
34598	ZINC000095098911	CCCCCCCCCCCCC[C@H](O)[C@H]1CC[C@H]([C@H](O)CCCC...
34599	ZINC000000001009	C[N+](C)([O-])CC/C=C1\c2ccccc2C=Cc2c(Cl)cccc21

```
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)
```

```
 84%|████████▍ | 581/692 [02:16<00:19,  5.65it/s]
 84%|████████▍ | 584/692 [02:16<00:17,  6.35it/s]
 85%|████████▍ | 586/692 [02:17<00:18,  5.70it/s]
 85%|████████▍ | 587/692 [02:17<00:25,  4.09it/s]
 85%|████████▍ | 589/692 [02:17<00:21,  4.71it/s]
 85%|████████▍ | 591/692 [02:18<00:16,  6.06it/s]
 86%|████████▍ | 592/692 [02:18<00:24,  4.00it/s]
 86%|████████▍ | 594/692 [02:19<00:22,  4.31it/s]
 86%|████████▍ | 595/692 [02:20<00:36,  2.64it/s]
 86%|████████▌ | 597/692 [02:20<00:25,  3.71it/s]
 87%|████████▋ | 600/692 [02:20<00:18,  4.89it/s]
 87%|████████▋ | 602/692 [02:20<00:16,  5.30it/s]
 87%|████████▋ | 603/692 [02:21<00:23,  3.74it/s]
 87%|████████▋ | 605/692 [02:21<00:17,  4.87it/s]
 88%|████████▊ | 608/692 [02:22<00:14,  5.78it/s]
 88%|████████▊ | 610/692 [02:22<00:12,  6.42it/s]
 88%|████████▊ | 611/692 [02:23<00:26,  3.00it/s]
 89%|████████▊ | 613/692 [02:23<00:19,  3.96it/s]
 89%|████████▉ | 616/692 [02:24<00:14,  5.10it/s]
 89%|████████▉ | 618/692 [02:24<00:12,  5.87it/s]
 89%|████████▉ | 619/692 [02:24<00:18,  3.92it/s]
 90%|████████▉ | 621/692 [02:25<00:14,  4.98it/s]
 90%|████████▉ | 624/692 [02:25<00:12,  5.66it/s]
 90%|█████████ | 626/692 [02:25<00:11,  5.85it/s]
 91%|█████████ | 627/692 [02:26<00:15,  4.28it/s]
 91%|█████████ | 629/692 [02:26<00:11,  5.68it/s]
 91%|█████████ | 632/692 [02:26<00:09,  6.55it/s]
 92%|█████████▏| 634/692 [02:27<00:08,  6.71it/s]
 92%|█████████▏| 635/692 [02:27<00:12,  4.41it/s]
 92%|█████████▏| 637/692 [02:27<00:09,  5.58it/s]
 92%|█████████▏| 640/692 [02:28<00:08,  6.09it/s]
 93%|█████████▎| 642/692 [02:28<00:07,  6.39it/s]
 93%|█████████▎| 643/692 [02:29<00:11,  4.33it/s]
 93%|█████████▎| 646/692 [02:29<00:06,  6.69it/s]
 94%|█████████▎| 648/692 [02:29<00:06,  6.46it/s]
 94%|█████████▍| 650/692 [02:30<00:06,  6.05it/s]
 94%|█████████▍| 651/692 [02:31<00:18,  2.25it/s]
 94%|█████████▍| 653/692 [02:32<00:12,  3.03it/s]
 95%|█████████▍| 655/692 [02:32<00:08,  4.13it/s]
 95%|█████████▍| 657/692 [02:32<00:08,  4.10it/s]
 95%|█████████▌| 658/692 [02:32<00:08,  4.21it/s]
 95%|█████████▌| 659/692 [02:33<00:10,  3.02it/s]
 96%|█████████▌| 661/692 [02:33<00:07,  4.17it/s]
 96%|█████████▌| 664/692 [02:34<00:05,  5.20it/s]
 96%|█████████▌| 666/692 [02:34<00:04,  5.65it/s]
 96%|█████████▋| 667/692 [02:34<00:06,  3.93it/s]
 97%|█████████▋| 669/692 [02:35<00:04,  5.17it/s]
 97%|█████████▋| 672/692 [02:35<00:03,  5.79it/s]
 97%|█████████▋| 674/692 [02:35<00:02,  6.22it/s]
 98%|█████████▊| 675/692 [02:36<00:04,  4.22it/s]
 98%|█████████▊| 677/692 [02:36<00:02,  5.55it/s]
 98%|█████████▊| 680/692 [02:36<00:01,  6.79it/s]
 99%|█████████▊| 683/692 [02:37<00:01,  8.29it/s]
 99%|█████████▉| 688/692 [02:37<00:00, 13.13it/s]
100%|█████████▉| 691/692 [02:37<00:00, 14.99it/s]
100%|██████████| 1/1 [02:38<00:00, 158.26s/it]
Saving predictions to named_preds.csv
Elapsed time = 0:02:52
```

```
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]...	0.345203
1	ZINC000150377216	CCCCCC/C=C\C/C=C\CCCCCCCC(=O)OC[C@H](COCCCCCCC...	0.619356
2	ZINC000100780125	CC(=O)O[C@H]1C[C@](C)(O)[C@@H]2CC=C(C)[C@@H]2[...	0.260698
3	ZINC000006580536	O=C(O)[C@H](Cc1ccccc1)N(CCCl)CCCl	0.043098
4	ZINC000150351802	O=C1C[C@H](c2ccc(O)c(O)c2)Oc2c1c(O)cc(O[C@H]1O...	0.716069

	HIV_active
count	34600.000000
mean	0.466442
std	0.243481
min	0.000041
25%	0.251037
50%	0.499464
75%	0.673119
max	0.998601

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

	zinc_id	smiles	HIV_active	HIV_ac
0	ZINC000030727788	C=C[C@]1(C)C[C@@H](OC(=O)CSC(C)	0.345203	

```
# 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
```

	zinc_id	smiles	HIV_activ
21	ZINC000040753343	COc1cc(/C=C/c2cc(O)c(CC=C(C)C)c(O)c2)cc2c1O[C@...	0.93427
37	ZINC000014615844	CC1(C)OC(=O)C=C[C@@]2(C)[C@@H]1CC(=O)[C@@]1(C)...	0.84531
91	ZINC000150366864	COc1cc(-c2[o+]c3cc(O)cc(O)c3cc2O[C@H]2O[C@H](C...	0.84361
110	ZINC000049888739	CCCCCC(=O)c1c(O)c2c(c3c1O[C@@]1(O)[C@H]([C@@H]...	0.88549
124	ZINC000150343906	CCCCC/C=C\C/C=C\CCCCCCCC(=O)OC[C@H]	0.82289