3.1 Problem Understanding

In chapter 2, we identified hits with high similarity to one or more Malaria Box compounds. Our next task will be to develop a ligand-based virtual screening model to identify hits that pose cardiotoxic risk. Our model is just one component of a much bigger virtual screening pipeline. A perfectly accurate model will be useless if it doesn't fulfill the company's expected use cases and end goal, which leads us to the first question we need to ask: what are the project requirements, constraints, and goals?

As input, we will use our file containing compounds that have high similarity to at least one of the Malaria Box compounds. Though these compounds are promising hits, we want to further filter out any compounds that are active to a cardiotoxicity antitarget. An *antitarget* (or off-target) is a receptor, enzyme, or other biological target that, when affected by a drug, causes undesirable side-effects. Whereas we want to identify compounds that are active against a target, we want to remove compounds that are active against an antitarget. Antitargets are commonly biomolecules that play important roles in normal physiological processes that are not directly related to the condition that we want to treat.

The antitarget we'll consider is the human ether-a-go-go-related gene (hERG) potassium channel. Drug-induced blockage of the hERG channel is considered the primary cause of cardiotoxicity and must be screened for early in the drug discovery process. The expected output of your model is binary – the compound does not block the hERG channel and should be kept or it does block the hERG channel and should be removed. Furthermore, the company cares about two failure cases: (1) we miss a cardiotoxic compound that proceeds through the pipeline to clinical trials and fails (incurring cost and damaging brand reputation) and (2) we incorrectly classify an otherwise therapeutically beneficial compound as cardiotoxic and cut its candidacy short.

Coming to a second question to consider, how will we benchmark the success of our model? We could compare our model's performance against other known models, as well as methods that don't involve ML, such as rule-based filters to detect HERG blockage by proxy. For instance, blockage of the hERG channel is associated with long QT syndrome, which is a heart disorder that can cause arrhythmia (fast, rapid heartbeats). To estimate hERG blockage, AI we might construct a small set of structural alerts that are associated with arrhythmia endpoints. If a compound contains several structural alerts greater than a user-defined threshold of these structures, it is filtered out.

TANGENT QT in "long QT" is a reference to the heart's electrical activity as recorded with an electrocardiogram (ECG). Different waves on the ECG graph are marked as P, Q, R, S, and T. Activity for Q through T corresponds to heart cells electrically "recharging" after muscle contraction. Long QT indicates slow recharging, which is associated with heart abnormalities.

3.1.1 Your Machine Learning Task

Where to begin? What ML model to use? Why do we expect an ML model to be better than a simple rule-based approach like structural alerts? Furthermore, suppose we did know the target's structure. Why might we still think that machine learning can add value, rather than moving directly to a structure-based method? There are a few guiding principles for assessing when machine learning may be of value:

1. Existing data is available or can be collected.
2. The data contains complex patterns, and our model has the capacity to learn how those patterns correspond to known outcomes. Here, *capacity* refers to the model's ability to learn a variety of possible functions, where greater model complexity implies greater capacity in learning.
3. We have enough data that important patterns may be repeated multiple times, making it easier for the model to remember them.
4. Our model can solve problems by using what it has learned to make predictions on new data, assuming that the new data is similar to the model's training data.
5. Our model can be applied to new data at scale and speed that is not possible with current solutions.

Our task is a typical classification task: the model needs to output which class each compound belongs. A compound belongs to either the positive class if hERG blocking or negative class if not. Since there are only two classes to predict, this is a binary classification task. If we were trying to predict more than two possible classes, it would be a multiclass classification problem. In general, binary classification is easier than multiclass classification. Lastly, we can train a model using a labeled dataset of compounds known to either block or not block the hERG channel, which makes this a supervised learning task.

3.2 Data Acquisition, Exploration, & Curation

As we've detailed for reference in appendix B, there are many public databases such as PubChem and ChEMBL with existing data for molecular property prediction. Another valuable source of data are publications that make the data used in their experiments publicly available. If you are already familiar with a specific subfield in cheminformatics, then you likely know of common benchmark datasets derived from individual publications. If you are not familiar with a field and what dataset publications exist, it is helpful to reference a dataset aggregator. For example, Papers With Code open sources datasets across a variety of machine learning fields and tasks, including some pertaining to drug discovery. Another collection specifically for drug discovery is aggregated by Therapeutics Data Commons (TDC) [1].

To run the code examples through this chapter, navigate to https://github.com/nrflynn2/ml-drug-discovery. This leads to a GitHub repository with the full list of available Jupyter notebooks for each chapter of this book. To follow along for chapter 3, open the chapter labeled *CH03_Flynn_ML4DD.ipynb*.

3.2.1 Loading and Exploring the hERG Blockers Dataset

If we navigate to TDC datasets for toxicity, we can find references for three datasets related to hERG blockers. For ease of use, let's start off with downloading and loading the "hERG blockers" dataset into a Pandas data frame.

Listing 3.1 Loading hERG Blockers Dataset

```
from pathlib import Path
import pandas as pd
import urllib.request

def load_herg_blockers_data():
  herg_blockers_path = Path("datasets/herg_blockers.xlsx")

  if not herg_blockers_path.is_file():
    Path("datasets").mkdir(parents=True, exist_ok=True)
    url = " https://github.com/nrflynn2/ml-drug-
discovery/blob/main/data/hERG_blockers.xlsx"
    urllib.request.urlretrieve(url, herg_blockers_path)

  return pd.read_excel(
      herg_blockers_path,
      usecols="A:F",
      header=None,
      skiprows=[0,1],
      names=["SMILES", "Name", "pIC50", "Class", "Scaffold Split", "Random Split"],
    ).head(-68)                                                       #A
```

#A We remove the last 68 rows, which refer to an evaluation set used by the paper's authors for a different experiment.

UNDERSTANDING THE DATA STRUCTURE

We can peek at the top three rows via herg_blockers.head(3)(table 3.1).

Table 3.1 The first three rows of the hERG blockers dataset.

SMILES	Name	pIC50	Class	Scaffold Split	Random Split
Fc1ccc(cc1)Cn1c2c(nc1NC1CCN(CC1)CCc1ccc(O)cc1)cccc2	DEMETHYLASTEMIZOLE	9.0	1.0	Training I	Training II
Fc1ccc(cc1)C(OCC[NH+]1CC[NH+](CC1)CCCc1ccccc1)c1ccc(F)cc1	GBR-12909	9.0	1.0	Training I	Training II
O=[N+]([O-])c1ccc(cc1)CCCCN(CCCCCCC)CC	LY-97241	8.8	1.0	Training I	Training II

Let’s take a quick look at the non-null count and data type of each column:

Listing 3.2 View total number of rows, number of non-null values, and data types

```
>>> herg_blockers.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 587 entries, 0 to 586
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   SMILES          587 non-null    object
 1   Name            565 non-null    object
 2   pIC50           526 non-null    float64
 3   Class           587 non-null    float64
 4   Scaffold Split  587 non-null    object
 5   Random Split    587 non-null    object
dtypes: float64(2), object(4)
memory usage: 27.6+ KB
```

The hERG blockers data is a set of 587 compounds. Each row represents one compound. There are six columns:

- Name: The name of the compound.
- SMILES: Textual representation of each compound’s structure.
- pIC$_{50}$: Higher pIC$_{50}$ indicates greater drug potency. pIC$_{50}$ is derived from IC$_{50}$, an experimental measure of the concentration of a drug at which it inhibits a specific biological process by 50%. The “IC” stands for *inhibitory concentration.* Reported IC$_{50}$ values can have a large range and different units. To standardize comparison, IC$_{50}$ is often converted to pIC$_{50}$. pIC$_{50}$ is the negative of the log of the IC$_{50}$ in molar (M). Molar (M) is a measurement of the concentration of a chemical species. For example:
  - An IC$_{50}$ of 1 nM is 10^{-9} M, which is a pIC$_{50}$ of 9.0. An IC$_{50}$ of 1 μM is 10^{-6} M, which is a pIC$_{50}$ of 6.0.
  - This attribute is null for several compounds, but we can’t use this attribute for training and it did not affect the number of non-null class attributes that we will use for labels.

- Class: Each drug is labeled with a “1” if it blocks the hERG channel and a “0” if it does not block the hERG channel. This is the column that we want to train a model to be able to predict. The authors used a threshold of IC$_{50}$ < 40 μM to define drugs as hERG blockers. This threshold corresponds to a pIC$_{50}$ > 4.4.
- Set I: The authors split the dataset into training and test data using a scaffold split. We won’t use the set I column until we cover scaffold splitting later.

- Set II: The authors split the dataset into training and test data using a random split. We will use the set II column to define our training and test data.

There is a clear demarcation between the compound classes based on pIC$_{50}$. However, we can’t use the pIC$_{50}$ column as a feature for model training as pIC$_{50}$ was used to define the class variable we are predicting. Furthermore, we may not want to rely on pIC50 measurements as it requires experimental data, which may be costly, and pIC50 data quality may vary due to missing values or inconsistent experiment conditions or readouts. Furthermore, pIC$_{50}$ is experimentally derived. The purpose of our model is to predict whether a compound is a hERG blocker or not prior to conducting an experiment.

Regardless of whether the data comes from a large, well-established public or commercial database or a small dataset in literature, curation is always necessary to mitigate possible mistakes in the dataset. Manually reviewing all dataset entries is not feasible and even a random selection of a small subset, while useful, might miss problematic entries. Instead, we can harness exploratory data analysis (EDA) to plot and summarize features of the data. EDA helps us understand the range of molecular properties that the data covers and can be used to spot inconsistencies.

VERIFYING THE TARGET PROPERTY DISTRIBUTION

Let’s get familiar with our data by plotting the distribution of pIC$_{50}$ values:

Listing 3.3 Distribution of pIC$_{50}$ for real dataset and dataset with simulated error

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(
    herg_blockers["pIC50"], kde=True,
    stat="density", kde_kws=dict(cut=3),
    alpha=.4, edgecolor=(1, 1, 1, .4),
)
plt.title("Distribution of pIC50")

simulated_error = herg_blockers["pIC50"] + 3.0                    #A
sns.histplot(
    herg_blockers["pIC50"].append(simulated_error, ignore_index=True), kde=True,
    stat="density", kde_kws=dict(cut=3),
    alpha=.4, edgecolor=(1, 1, 1, .4),
)
plt.title("Distribution of pIC50, Simulated Annotation Error")
```

#A Shifting the pIC50 distribution to simulate annotation error due to units disagreement.