PH291-Exploratory project
Abhimanyu Mandal
19165001

Neurotoxicity induced by drugs

Introduction

Neurotoxicity is the form of toxicity caused due to exposure with biological, chemical, or physical agents and adversely affects the structure or function of the central and/or peripheral nervous system by altering the normal activity of the nervous system. This can either cause permanent or reversible damage to the nervous tissues, or disrupt the functioning of neurons. Numerous chemicals affect the neural cells, which causes neurotoxic diseases in humans, while others interfere with the metabolic processes on which the nervous system is specially dependent.

Most drugs have non-negligible neurotoxic effects many of which are primarily mediated by several dopaminergic and glutamatergic neurotransmitter systems.

Symptoms of Neurotoxicity

Neurotoxic disorders follow overexposure/overdose of abused substances (e.g., ethanol, narcotics), therapeutic drugs, toxic substances such as pesticides, insecticides, herbicides, fungicides, or industrial chemical warfare agents.

The disorders caused due to neurotoxic agents may appear immediately after exposure or it may get delayed by months to years of exposure. Some of the common symptoms of neurotoxicity includes:

- Limb weakness or numbness
- Loss of memory

Loss of eyesight

Uncontrolled obsessive and/or compulsive behaviors

Delusions

Headache

Sexual dysfunction

Causes of neurotoxicity

In some cases, the level or the tenure of exposure may be critical, with some substances only becoming neurotoxic in certain doses or time periods. One such example is the overdose of aspirin. With a dosage of more than 250 mg/kg aspirin, it is likely to cause mild toxicity, which includes nausea, vomiting, tinnitus, lethargy or dizziness, while an intake of more than 500 mg/kg aspirin leads to severe and possibly fatal toxicity.

Tenure of exposure is another important factor for neurotoxicity. Some of the most common naturally occurring brain toxins that lead to neurotoxicity due to long term drug usage are beta amyloid (Aβ), glutamate, dopamine, and oxygen radicals. When these drugs are consumed for a long term, the concentration of these drugs increases, resulting in neurotoxicity and apoptosis.

LD_{50} value

LD stands for “Lethal Dose”. LD_{50} value is the amount of chemical administered in a dosage form that leads to death of 50% of a group of test animals, usually rats and mice. The LD_{50} value is a potential method of measuring the short-term poisoning of a chemical substance. The LD_{50} value of a chemical substance can be determined for any route of administration, however dermal (skin) and oral (mouth) administration are most commonly used.

Mechanism of toxicity

The mechanism of action of a chemical agent, or a drug compound that lead to nervous system dysfunction varies with chemical agents. Some chemical agents act directly on the

nervous tissue and disrupt the functioning of neurons; while others induce neurological or behavioral dysfunction indirectly by inducing changes in the electrolytic balance either in the cerebral blood flow, or in metabolism of glucose, or in the levels of critical intermediary metabolites. Hence, due to the distinctive sensitivity of neural tissue to the disruption of body homeostasis, pathophysiological changes are expressed clinically as neurobehavioral disorders.

Importance

Current studies have shown that it is very difficult to estimate the number of people suffering from neurological disorders as well as the extent of neurological dysfunction resulting from direct/indirect exposure to a sufficient dose of toxic compounds.

However, with the goal of increasing the therapeutic activity of a drug, drug molecules are studied in vivo to understand the chemical groups producing toxicity and are then altered with the aim of improving their toxicity at a particular dose. This is one of the best techniques of formulation of a new drug based on its toxicophore rather than the primitive method of drug formulation based on pharmacophore.

Objective

To detect neurotoxic and neuroprotective agents with the help of molecular descriptors and Support Vector Machine Model.

Plan Of Action

The project work can be divided into four sections:

Part 1: Listing of compounds in a spreadsheet

1. Make a list of drugs which includes both neurotoxic and neuroprotective compounds.

2. Derive the SMILES of the compounds from PubChem (PubChem (nih.gov)).
3. Ranking of all the compounds as Neurotoxic (1) or Neuroprotective (0).

Part 2: Creating an *.smi file or an *.sdf file and

1. Create an *.smi file or an *.sdf file which includes the list of SMILES only.
2. The above *.smi file or a *.sdf file is then used to find all the descriptors from BioTriangle (Index-BioChem-BioTriangle-BioTrangle (scbdd.com)).

Part 3: Descriptors

1. Once the descriptors are derived from BioTriangle, it is saved as *.csv file.
2. The descriptors are then shortlisted on the basis of less null value, i.e., those descriptors which have no/ very little value are removed. This is known as Data Wrangling, and is done to avoid any error in the determination of neurotoxicity.
3. The type of descriptors chosen for this purpose belong to the following categories:

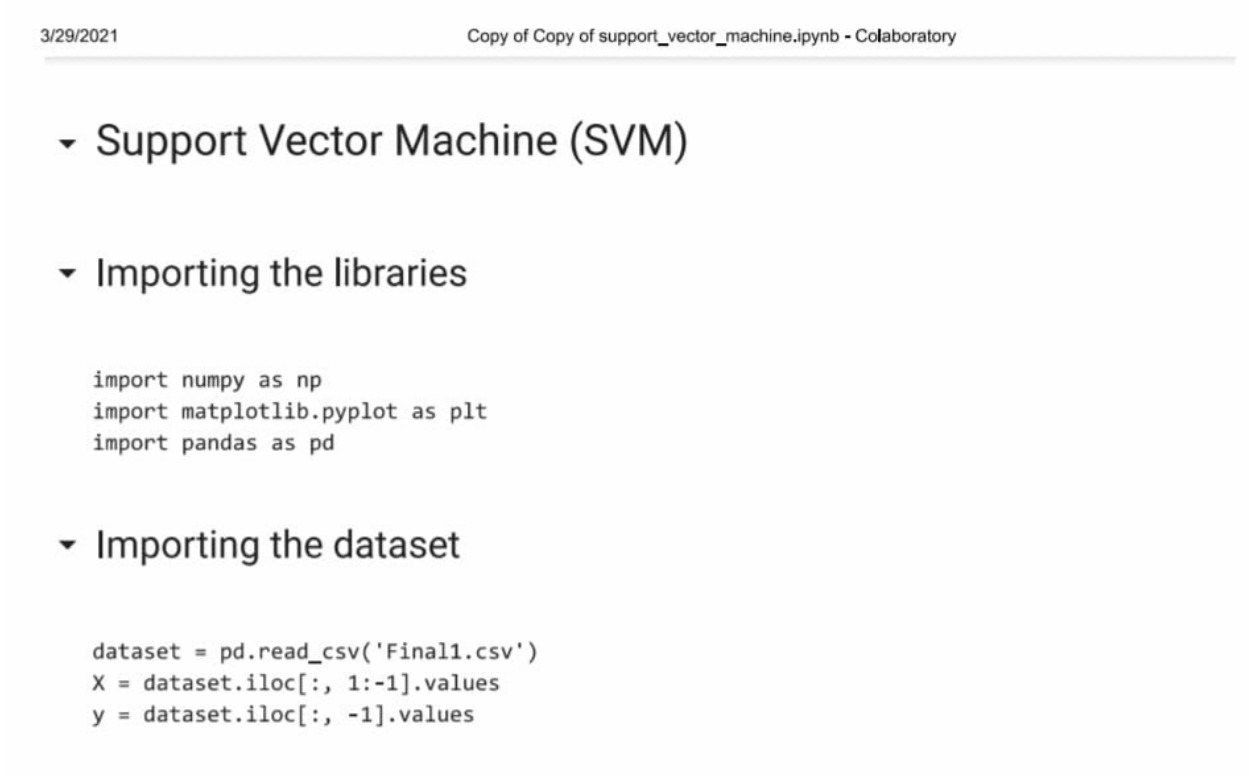
   Constitution, Topology, Connectivity, Kappa, EState, Autocorrelation-moran, Autocorrelation-geary, Autocorrelation-broto, Molecular properties, Charge, Moe-Type descriptors

4. The descriptor which here describes the activity is the oral LD50 value for rat.

Part 4: Choosing a Machine Learning model

1. An ML model must be created for the prediction of neurotoxicity of a drug using the programming language Python and scientific packages like Rdkit.
2. Attach the *.csv file created and check the efficiency and accuracy of the model for the prediction of neurotoxicity of the list of drugs.
3. In this case, the Support Vector Machine Model is used and an accuracy of approximately 81% is achieved which makes us conclude the project.

Here is the visual representation of the model used. The first steps begin by importing different packages which will help us follow with further steps. Then we import the dataset and divide the columns into x and y datasets. The y dataset contains the rank which determines whether the compound is neurotoxic or neuroprotective, While the x dataset contains all descriptors corresponding to the compounds. The datasets are further divided in training sets and test sets which will help us understand the accuracy of our model.
As there is a probability that some data is missing i.e there might be some nan values, so those values are converted into numerical values. Then, using a standard scaler the values are feature scaled.



Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state =
```

```
print(X_train)
```

```
[[7.8000e+02 1.0000e+00 2.0000e+00 ... 0.0000e+00 4.9230e+00 3.0332e+01]
 [8.9390e+03 4.0000e+00 2.7000e+01 ... 0.0000e+00 5.6781e+01 2.3801e+01]
 [1.8000e+02 1.0000e+00 9.0000e+00 ... 4.5240e+00 2.0475e+01 2.3095e+01]
 ...
 [       nan 1.0000e+00 1.2000e+01 ... 4.7370e+00 5.9690e+00 7.8863e+01]
 [4.7000e+03 0.0000e+00 3.0000e+00 ... 0.0000e+00 2.3427e+01 0.0000e+00]
 [5.0000e+03 4.0000e+00 1.8000e+01 ... 4.3050e+00 6.5186e+01 3.0341e+01]]
```

```
print(y_train)
```

```
[1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 1 0]
```

```
print(X_test)
```

```
[[7.5000e+02 1.0000e+00 3.0000e+00 ... 5.7340e+00 4.9230e+00 2.4265e+01]
 [3.5000e+02 0.0000e+00 1.7000e+01 ... 4.7370e+00 3.1644e+01 1.8199e+01]
 [4.7700e+02 1.0000e+00 1.0000e+01 ... 5.3170e+00 1.2831e+01 2.4265e+01]
 ...
 [6.0000e+01 0.0000e+00 1.0000e+00 ... 1.1685e+01 0.0000e+00 0.0000e+00]
 [5.0000e+01 0.0000e+00 7.0000e+00 ... 0.0000e+00 2.3476e+01 2.4526e+01]
 [1.2500e+03 1.0000e+00 1.0000e+01 ... 1.0217e+01 3.0576e+01 5.3523e+01]]
```



```
print(y_test)
```

```
[1 0 0 1 1 1 0 0 1 1 1 0]
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
print(X_train)
```

```
[[-0.80341904 -0.33859959 -1.23815074 ... -0.69816469 -1.13160888
   0.26643125]
 [ 1.55633291  1.63656468  2.22313327 ... -0.69816469  1.529547
   0.0048902 ]
 [-0.97695148 -0.33859959 -0.26899121 ...  0.30897241 -0.33353922
  -0.02338234]
 ...
 [        nan -0.33859959  0.14636287 ...  0.35639067 -1.07793212
   2.2099083 ]
 [ 0.33032623 -0.99698768 -1.09969938 ... -0.69816469 -0.18205378
  -0.94824692]
 [ 0.41709245  1.63656468  0.97707103 ...  0.26021843  1.96085973
   0.26679166]]
```

```
print(X_test)
```

```
[[-0.81209566 -0.33859959 -1.09969938 ...  0.57834374 -1.13160888
   0.02347159]
 [-0.92778396 -0.99698768  0.83861967 ...  0.35639067  0.2396115
  -0.21944803]
 [-0.89105292 -0.33859959 -0.13053985 ...  0.48551081 -0.72580031
   0.02347159]
 ...
 [-1.01165797 -0.99698768 -1.3766021  ...  1.90316091 -1.38423856
  -0.94824692]
 [-1.01455017 -0.99698768 -0.54589393 ... -0.69816469 -0.17953928
   0.03392362]
 [-0.6674853  -0.33859959 -0.13053985 ...  1.57635338  0.18480579
   1.19514025]]
```

Once the feature scaling is done , using the SVM model the training sets are used for preparing the predicting model and then applied on X_test. By using the y_pred, we can compare it with y_test to achieve the accuracy score.

Training the SVM model on the Training set

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
```



```
max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
verbose=False)
```

```
X_train = np.nan_to_num(X_train)
X_test = np.nan_to_num(X_test)
y_train = np.nan_to_num(y_train)
y_test = np.nan_to_num(y_test)
```

Predicting a new result

Predicting the Test set results

```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[1 1]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 0]
 [1 1]
 [1 1]
 [0 1]
 [0 0]]
```

Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[4 1]
 [1 6]]
0.8333333333333334
```

Once this is done and accuracy score of more than 80% is achieved (83%), the next step is to judge the importance of the descriptors used. This can be done by various methods but this particular method has been chosen.

```
import pandas as pd
%matplotlib inline
#do code to support model
#"data" is the X dataframe and model is the SKlearn object

feats = {} # a dict to hold feature_name: feature_importance
for feature, importance in zip(dataset.columns[:], model.feature_importances_):
    feats[feature] = importance #add the name/value pair
plt.figure(figsize=(16,6))
importances = pd.DataFrame.from_dict(feats, orient='index').rename(columns={0: 'Gini-importance'})
importances.sort_values(by='Gini-importance').plot(kind='bar', rot=90)
```

Using this method it was found out that there were two main descriptors with very high importance. They were MZM2 (Modified Zagreb index with order 1-2) and Chi5ch(rdkit descriptor). The LD50 was also found to have high importance when compared to the order descriptors.

Reference

1. Neurotoxicity - an overview | ScienceDirect Topics
2. Neurotoxicity - Wikipedia
3. https://patient.info/doctor/salicylate-poisoning#:~:text=However%2C%20ingestion%20of%20more%20than%20250%20mg%2Fkg%20aspirin,poisoning%20causes%20nausea%2C%20vomiting%2C%20tinnitus%2C%20lethargy%20or%20dizziness.
4. What is a LD50 and LC50? : OSH Answers (ccohs.ca)