Bioinformatics Diploma

Course (CIT-655 -fa21)

QSAR Model for Predicting the Activity of Beta-Secretase (BACE1) Inhibitors Associated with Alzheimer's Disease

Author: - Ahmed Badr, ID: - 2011054

Under supervision of: - Dr. Tamer M. Ibrahim

Abstract

Alzheimer's disease is one of the most common neurodegenerative disorders and a cause of progressive dementia worldwide. The two basic pathological features of AD are extra-neuronal plaques of misfolded β-amyloid proteins and intraneuronal neurofibrillary tangles of hyperphosphorylated tau protein. the drugs available to treat it do not cure or prevent the disease progression.

The β-site amyloid cleavage enzyme 1 (BACE1) is the major constituent of amyloid plaques and plays a central role in this brain pathogenesis. The cleavage of APP by β-secretase (BACE1) followed by γ-secretase generates Aβ which forms insoluble plaques, the strategy to inhibit BACE1 would result in a significant decrease in the Aβ load and would result in improved cognitive functions. So, (BACE1) inhibitor can be a promising anti-Alzheimer agent.

Computational approaches have proved to be very useful in saving time and money. a QSAR model for prediction of potential inhibitors of BACE1 protein is designed by using regression method with the aid of machine learning algorithms, getting data from ChEMBL database, and building the model on IC50 values from obtained inhibitors.

The features for the analysis are the PubChem fingerprints binary representation as our descriptors, and the model is built on Random Forest Regression Algorithm to be able to predict the inhibitory effect of a compound on BACE1.

Keywords

BACE-1 inhibitors, QSAR, Beta-secretase, pIC50, Random Forest Algorithm

1 Introduction

The story started in the year 1907 when Alois Alzheimer reported a woman with rapid and progressive memory deterioration and psychiatric disturbances. She was reported dead 4 years later. Alzheimer’s disease (AD) is the most common cause of dementia affecting 47 million people worldwide. (1)

Alzheimer’s disease (AD) is an irreversible and progressive neurodegenerative disease that slowly results in the state of dementia. The primary pathological

findings are the extracellular β-amyloid plaques (Aβ) and intracellular neurofibrillary tangles (NFTs).

According to Alzheimer's Association, the basic symptoms of AD are memory loss, difficulty in solving problems, difficulty doing work at home, confusion with remembering time and dates, the problem in speaking or writing, and loss of ability to retrace events and steps. (2)

1.2 Pathophysiology of AD

The two most widely accepted biochemical reasons for cognitive impairment in AD patients are a continuous loss of acetylcholine (Ach) due to hyperactive acetylcholinesterase enzyme (Ache) and hyperactive N-methyl-D-aspartate (NMDA) glutamate receptors.

Since it is widely known that Ach is primarily responsible for cognition and behavioral aspect, loss of Ach results in cognitive impairment. Hyperactive glutamate NMDA receptor leads to increased influx of calcium ions (neuronal excitotoxicity) and increased free radical generation that becomes detrimental to the neurons. (3)

The amyloid precursor protein or APP is cleaved by three different enzymes in two separate pathways. The initial two enzymes are α-secretase and β-secretase while γ-secretase comes later in the picture.

When APP is cleaved by α-secretase followed by γ-secretase, soluble fragments of 40 amino acid length are generated. But when the same APP is cleaved by β-secretase followed by γ-secretase, the amyloidogenic Aβ fragments are generated which are 42 amino acids long.

These Aβ fragments are hydrophobic and tend to aggregate creating extra-neuronal plaques. With an increase in the aggregation of these insoluble fragments, the toxicity increases which leads to synaptic dysfunction and gradual neuronal death. (4)

The cleavage of APP by β-secretase (BACE1) followed by γ-secretase generates Aβ which forms insoluble plaques, while the Aβ fragments are not generated if the initial enzyme is α-secretase followed by γ-secretase action. Thus, the strategy to inhibit BACE1 would result in a significant decrease in the Aβ load and would result in improved cognitive functions (5). So, (BACE1) inhibitor can be a promising anti-Alzheimer agent.

1.3 QSAR

quantitative structure-activity relationship (QSAR) is a way to find a simple equation that can be used to predict some property from the molecular structure of a compound, and this can be done by using curve fitting to find the equation coefficients.

Several (QSAR) models have been developed in order to predict potential inhibitors for protein BACE1, QSAR methods correlate molecular structure to different biological properties such as activity or ADMET properties, providing a relevant data to help during the development of drug design projects.

A key step in QSAR studies is the definition of chemical structure by molecular descriptors such as constitutional, topological, thermodynamic, functional groups, quantum mechanical, geometrical, etc.

2 Materials and Methods

The analysis is based on BASH and Python and tested on Colab and jupyter notebook.

Sublime text editor only used for visualization of the code.

2.1 Bioactivity Data

The data was installed from ChEMBL database, searching for the target protein (BACE1), Then selecting and retrieving bioactivity data for Human BACE1 (first entry) CHEMBL4822.

we will retrieve only bioactivity data for Human BACE1 (CHEMBL4822) that are reported as pChEMBL values, filtering inhibitors for our target by IC50 values.

After we got our data frame (10156 rows × 45 columns) of inhibitors that share the category of IC50, we need to clean our data by handling missing values and removing the duplicates.

Some data pre-processing would be convenient to ease our work by Combining the 3 columns (molecule_chembl_id, canonical_smiles, standard_value) into a new data frame to append class activity later to it.

Labeling compounds as either being active, inactive or intermediate by their IC50 value. Compounds having values of less than 1000 nM will be active while those greater than 10,000 nM will be inactive. As for those values in between 1,000 and 10,000 nM will be referred to as intermediate.

Colab note: - https://colab.research.google.com/drive/1YSlFVWJidzSHcWy-lHCU3_x4pDLcfRfB

2.2 Exploratory Analysis

Loading our curated data (7062 rows × 4 columns), our columns are (molecule_chembl_id, canonical_smiles, standard_value, class), First analysis is Calculating Lipinski descriptors, this is the first step to our QSAR model.

Christopher Lipinski, a scientist at Pfizer, came up with a set of rule-of-thumb for evaluating the drug likeness of compounds. Such drug likeness is based on the Absorption, Distribution, Metabolism and Excretion (ADME) that is also known as the pharmacokinetic profile. Lipinski analyzed all orally active FDA-approved drugs in the formulation of what is to be known as the Rule-of-Five or Lipinski's Rule. (6)

The Lipinski's Rule stated the following:

- Molecular weight < 500 Dalton
- Octanol-water partition coefficient (LogP) < 5
- Hydrogen bond donors < 5
- Hydrogen bond acceptors < 10

Calculating these parameters for our set of data then appending the output of Lipinski descriptors to our curated data, another pre-processing step is to convert IC50 to pIC50.

To allow IC50 data to be more uniformly distributed, we will convert IC50 to the negative logarithmic scale which is essentially -log10(IC50).

This custom function pIC50() will accept a Data Frame as input and will:

- Take the IC50 values from the standard_value column and converts it from nM to M by multiplying the value by 10 ^ −9.
- Take the molar value and apply -log10.
- Delete the standard_value column and create a new pIC50 column.

Applying our function to standard values after normalizing it, then removing the intermediate values in class category, to prepare our data for statistical analysis between actives and inactives.

Chemical Space Analysis via Lipinski descriptors by Detecting significance in our data according to Lipinski descriptors and IC50 values by using box plots and Mann–Whitney U statistical test to get P-values for each.

U statistic equation (7): -

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} S(X_i, Y_j), \quad \text{with} \quad \begin{cases} 0, & \text{if } Y > X. \end{cases}$$

Colab note: - https://colab.research.google.com/drive/1HUpSqZqKg5LHobQGzViCmQnb2b0nnB7l

2.3 Descriptor Calculation and Dataset Preparation

Descriptor analysis can be a challenging task to choose which type of descriptors to work with, in our analysis, the descriptors are PubChem Fingerprints.

The fingerprints provided by PubChem are a binary representation of the presence and absence of a library of 881 substructure features, in this system every molecular structure is described by 881 bits where 1 indicates the presence and 0 the absence of a feature. Compared to atom pairs, the PubChem fingerprints are a knowledge-based system that stores less information than the much more complex and unbiased atom pair concept. For database searching fingerprints are often much more time and memory efficient, but they are less sensitive than atom pair descriptors. (8)

Loading our curated data then extracting (molecule_chembl_id, canonical_smiles) columns to be processed by PaDEL descriptors, a bash command, to get fingerprints for our inhibitors and using pIC50 values to build a regression model, next step is to Prepare the X and Y data matrices.

- X data matrix: - descriptors output.
- Y data matrix: - pIC50 values.

```
#Descriptor Calculation and Dataset Preparation
#Download PaDEL-Descriptor
! wget https://github.com/dataprofessor/bioinformatics/raw/master/padel.zip
! wget https://github.com/dataprofessor/bioinformatics/raw/master/padel.sh
! unzip padel.zip

#Load bioactivity data
import pandas as pd
df3 = pd.read_csv('BACE1_04_bioactivity_data_3class_pIC50.csv')
selection = ['canonical_smiles','molecule_chembl_id']
df3_selection = df3[selection]
df3_selection.to_csv('molecule.smi', sep='\t', index=False, header=False)

#Calculate fingerprint descriptors
! bash padel.sh

#Preparing the X and Y Data Matrices
#X data matrix
df3_X = pd.read_csv('descriptors_output.csv')
df3_X = df3_X.drop(columns=['Name'])

#Y variable
#Convert IC50 to pIC50
df3_Y = df3['pIC50']

#Combining X and Y variable
dataset3 = pd.concat([df3_X,df3_Y], axis=1)

dataset3.to_csv('BACE1_06_bioactivity_data_3class_pIC50_pubchem_fp.csv', index=False)
```

Colab note: -
https://colab.research.google.com/drive/15d_p1eA1yS8K7_q7buxW-eLU6nSrjFa5#scrollTo=75npGyvhae0e

2.4 Regression Model with Random Forest

The BACE1 data set (7062,881) contains 881 input features, 1 output variable (pIC50 values) and 7062 curated inhibitors, but before building a machine learning model we need to remove the low variance features in the binary fingerprints to yield a data set of (7062, 160) with high variance.

Splitting our data to (80/20 ratio) 80% for the training set and 20% for the test set, then applying [RandomForestRegressor] function. (9)

```python
#Regression Models with Random Forest
#Import libraries
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

df = pd.read_csv('BACE1_06_bioactivity_data_3class_pIC50_pubchem_fp.csv')

# Input features
X = df.drop('pIC50', axis=1)

# Output features
Y = df.pIC50

# Remove low variance features
from sklearn.feature_selection import VarianceThreshold
selection = VarianceThreshold(threshold=(.8 * (1 - .8)))
X = selection.fit_transform(X)

#Data split (80/20 ratio)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)

#Building a Regression Model using Random Forest
model = RandomForestRegressor(n_estimators=100)
model.fit(X_train, Y_train)
r2 = model.score(X_test, Y_test)
Y_pred = model.predict(X_test)

#Scatter Plot of Experimental vs Predicted pIC50 Values
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(color_codes=True)
sns.set_style("white")

ax = sns.regplot(Y_test, Y_pred, scatter_kws={'alpha':0.4})
ax.set_xlabel('Experimental pIC50', fontsize='large', fontweight='bold')
ax.set_ylabel('Predicted pIC50', fontsize='large', fontweight='bold')
ax.set_xlim(0, 12)
ax.set_ylim(0, 12)
ax.figure.set_size_inches(5, 5)
plt.show
```

$$\hat{y} = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} W_j(x_i, x')\, y_i = \sum_{i=1}^{n}\left(\frac{1}{m}\sum_{j=1}^{m} W_j(x_i, x')\right) y_i.$$

Colab note: - https://colab.research.google.com/drive/15rlFVBMDHhSDqpYtQ_eo0EBGonOWP0rt#scrollTo=hfqpfjxw3IAK

2.5 model deployment

In model deployment we will do the same steps as the previous one to build our model then saving it as a pickle extension to use it in a python script that will run our prediction app.

After the deployment of the app we will be able to input any structure with chembl id and SMILES and the output is a prediction of inhibitory effect on BACE1 target.

The steps for building the app can be obtained from provided link.

```
import streamlit as st
import pandas as pd
from PIL import Image
import subprocess
import os
import base64
import pickle


# Molecular descriptor calculator
def desc_calc():
    # Performs the descriptor calculation
    bashCommand = "java -Xms2G -Xmx2G -Djava.awt.headless=true -jar ./PaDEL-Descriptor/PaDEL-Descriptor.jar -removesalt
    process = subprocess.Popen(bashCommand.split(), stdout=subprocess.PIPE)
    output, error = process.communicate()
    os.remove('molecule.smi')

# File download
def filedownload(df):
    csv = df.to_csv(index=False)
    b64 = base64.b64encode(csv.encode()).decode()  # strings <-> bytes conversions
    href = f'<a href="data:file/csv;base64,{b64}" download="prediction.csv">Download Predictions</a>'
    return href

# Model building
def build_model(input_data):
    # Reads in saved regression model
    load_model = pickle.load(open('BACE1.pkl', 'rb'))
    # Apply model to make predictions
    prediction = load_model.predict(input_data)
    st.header('**Prediction output**')
    prediction_output = pd.Series(prediction, name='pIC50')
    molecule_name = pd.Series(load_data[1], name='molecule_name')
    df = pd.concat([molecule_name, prediction_output], axis=1)
    st.write(df)
    st.markdown(filedownload(df), unsafe_allow_html=True)
```

My app: - https://drive.google.com/drive/folders/170o4EMP-XYmw3MxE9v4frQeM9CiYIeAn?usp=sharing

2.6 Further analysis

For further analysis we can compare several ML algorithms for build regression models of BACE1 inhibitors.

We will use Lazypredict to compare ML algorithms on our training dataset according to their performance, RMSE (root-mean-square error) and time taken for each one.

```
#Comparing Regressors
#Import libraries

! pip install lazypredict

import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
import lazypredict
from lazypredict.Supervised import LazyRegressor

#load data and pre-processing
df = pd.read_csv('BACE1_06_bioactivity_data_3class_pIC50_pubchem_fp.csv')
X = df.drop('pIC50', axis=1)
Y = df.pIC50

# Remove low variance features
from sklearn.feature_selection import VarianceThreshold
selection = VarianceThreshold(threshold=(.8 * (1 - .8)))
X = selection.fit_transform(X)
X.shape

# Perform data splitting using 80/20 ratio
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

#Compare ML algorithms
# Defines and builds the lazyclassifier
clf = LazyRegressor(verbose=0,ignore_warnings=True, custom_metric=None)
models_train,predictions_train = clf.fit(X_train, X_train, Y_train, Y_train)
models_test,predictions_test = clf.fit(X_train, X_test, Y_train, Y_test)

# Performance table of the training set (80% subset)
predictions_train

# Data visualization of model performance
# Bar plot of R-squared values
import matplotlib.pyplot as plt
import seaborn as sns

#train["R-Squared"] = [0 if i < 0 else i for i in train.iloc[:,0] ]

plt.figure(figsize=(5, 10))
sns.set_theme(style="whitegrid")
ax = sns.barplot(y=predictions_train.index, x="R-Squared", data=predictions_train)
ax.set(xlim=(0, 1))
```

Colab note: - https://colab.research.google.com/drive/1UT7xA9Aom8Q-XiTG5LhycvG1TytotLjW#scrollTo=MLsi5uSvbpC7

3 Results

3.1 Bioactivity Data

	cross_references	organism	pref_name	score	species_group_flag	target_chembl_id	target_components	target_type	tax_id
0	[]	Homo sapiens	Beta-secretase (BACE)	15.0	False	CHEMBL2111390	[{'accession': 'Q9Y5Z0', 'component_descriptio...	PROTEIN FAMILY	9606.0
1	[{'xref_id': 'Beta-secretase_1', 'xref_name': ...	Homo sapiens	Beta-secretase 1	13.0	False	CHEMBL4822	[{'accession': 'P56817', 'component_descriptio...	SINGLE PROTEIN	9606.0
2	[{'xref_id': 'P56818', 'xref_name': None, 'xre...	Mus musculus	Beta-secretase 1	13.0	False	CHEMBL4593	[{'accession': 'P56818', 'component_descriptio...	SINGLE PROTEIN	10090.0
3	[]	Rattus norvegicus	Beta-secretase 1	13.0	False	CHEMBL3259473	[{'accession': 'P56819', 'component_descriptio...	SINGLE PROTEIN	10116.0
4	[{'xref_id': 'PTGS1', 'xref_name': None, 'xref...	Homo sapiens	Cyclooxygenase-1	4.0	False	CHEMBL221	[{'accession': 'P23219', 'component_descriptio...	SINGLE PROTEIN	9606.0
...	...	...	...	...	...	...	...	...	...
3319	[]	Zika virus	Genome polyprotein	0.0	False	CHEMBL4523307	[{'accession': 'Q32ZE1', 'component_descriptio...	SINGLE PROTEIN	64320.0
3320	[]	Severe acute respiratory syndrome coronavirus 2	Replicase polyprotein 1ab	0.0	False	CHEMBL4523582	[{'accession': 'P0DTD1', 'component_descriptio...	SINGLE PROTEIN	2697049.0
3321	[]	Yellow fever virus (strain 17D vaccine) (YFV)	Genome polyprotein	0.0	False	CHEMBL4523585	[{'accession': 'P03314', 'component_descriptio...	SINGLE PROTEIN	11090.0
3322	[]	Homo sapiens	Cytochrome P450	0.0	False	CHEMBL4523986	[{'accession': 'P08684', 'component_descriptio...	PROTEIN FAMILY	9606.0
3323	[]	Rattus norvegicus	I-kappa-B kinase	0.0	False	CHEMBL4524001	[{'accession': 'Q9QY78', 'component_descriptio...	PROTEIN COMPLEX	10116.0

3324 rows × 9 columns

Fig (1) searching for the target protein (BACE1), Then selecting and retrieving bioactivity data for Human BACE1(first entry) CHEMBL4822

	activity_comment	activity_id	activity_properties	assay_chembl_id	assay_description	assay_type	assay_variant_accession	assay_variant_mutation	bao_endpoint	bao_forı
0	None	78857	[]	CHEMBL653511	Inhibitory activity against Beta-secretase 1 w...	B	None	None	BAO_0000190	BAO_0000
1	None	391560	[]	CHEMBL653332	Compound was tested for its inhibitory activit...	B	None	None	BAO_0000190	BAO_0000
2	None	391983	[]	CHEMBL653512	Inhibition of human Beta-secretase 1	B	None	None	BAO_0000190	BAO_0000
3	None	395858	[]	CHEMBL653512	Inhibition of human Beta-secretase 1	B	None	None	BAO_0000190	BAO_0000
4	None	395859	[]	CHEMBL653512	Inhibition of human Beta-secretase 1	B	None	None	BAO_0000190	BAO_0000
...	...	...	...	...	...	...	...	...	...	
10138	None	19482217	[]	CHEMBL4480749	Inhibition of human BACE1 (1 to 460 residues) ...	B	None	None	BAO_0000190	BAO_0000
10139	None	19482218	[]	CHEMBL4480749	Inhibition of human BACE1 (1 to 460 residues) ...	B	None	None	BAO_0000190	BAO_0000
10140	None	19482219	[]	CHEMBL4480749	Inhibition of human BACE1 (1 to 460 residues) ...	B	None	None	BAO_0000190	BAO_0000

Fig (2) retrieve only bioactivity data for Human BACE1 (CHEMBL4822) that are reported as pChEMBL values, filtering inhibitors for our target by IC50 values.

	molecule_chembl_id	canonical_smiles	standard_value	class
0	CHEMBL406146	CC(C)C[C@H](NC(=O)[C@@H](NC(=O)[C@@H](N)CCC(=O...	413.00	active
1	CHEMBL78946	CC(C)C[C@H](NC(=O)[C@H](CC(N)=O)NC(=O)[C@@H](N...	2.00	active
2	CHEMBL324109	CCC(C)C[C@H](NC(=O)[C@H](CC(C)C)NC(C)=O)[C@@H]...	460.00	active
3	CHEMBL114147	CC(=O)NCC(=O)N[C@@H](Cc1ccccc1)[C@@H](O)CC(=O)...	9000.00	intermediate
4	CHEMBL419949	CC(=O)N[C@@H](Cc1ccccc1)C(=O)N[C@@H](Cc1ccccc1...	5600.00	intermediate
...	...	...	...	...
7057	CHEMBL4565226	CC(Cc1cc2ccccc2nc1N)C(=O)NC[C@@]12CCCO[C@@H]1C...	33113.11	inactive
7058	CHEMBL4520156	Nc1nc2ccccc2cc1CCC(=O)N1CC[C@H]2OCCC[C@@]2(Cc2...	85113.80	inactive
7059	CHEMBL4585673	Nc1nc2ccccc2cc1CCC(=O)NC[C@@]12CCCO[C@@H]1CCOC2	28840.32	inactive
7060	CHEMBL4546115	COc1ccc2c(c1)[C@@H](O)[C@@]1(CCN(C(=O)CCc3cc4c...	54954.09	inactive
7061	CHEMBL1821813	Nc1nc2ccccc2cc1CCC(=O)NCC1CCCCC1	30902.95	inactive

7062 rows × 4 columns

Fig (3) pre-processed data frame Combining 4 columns (molecule_chembl_id, canonical_smiles, standard_value, class)

3.2 Exploratory Analysis

	molecule_chembl_id	canonical_smiles	class	MW	LogP	NumHDonors	NumHAcceptors	pIC50
0	CHEMBL406146	CC(C)C[C@H](NC(=O)[C@@H](NC(=O)[C@@H](N)CCC(=O...	active	999.085	-1.4355	13.0	13.0	6.384050
1	CHEMBL78946	CC(C)C[C@H](NC(=O)[C@H](CC(N)=O)NC(=O)[C@@H](N...	active	893.005	-1.7361	12.0	12.0	8.698970
2	CHEMBL324109	CCC(C)C[C@H](NC(=O)[C@H](CC(C)C)NC(C)=O)[C@@H]...	active	751.988	2.3535	8.0	9.0	6.337242
5	CHEMBL116826	CCC(C)C[C@H](NC(=O)[C@H](CC(C)C)NC(C)=O)[C@@H]...	inactive	767.987	1.3690	8.0	9.0	4.468521
6	CHEMBL143239	CC(C)[C@H](NC(=O)C[C@H](O)[C@H](Cc1cc(F)cc(F)c...	active	717.722	4.3196	7.0	7.0	7.698970
...	...	...	...	...	...	...	...	...
7057	CHEMBL4565226	CC(Cc1cc2ccccc2nc1N)C(=O)NC[C@@]12CCCO[C@@H]1C...	inactive	383.492	2.6975	2.0	5.0	4.480000
7058	CHEMBL4520156	Nc1nc2ccccc2cc1CCC(=O)N1CC[C@H]2OCCC[C@@]2(Cc2...	inactive	429.564	4.3900	1.0	4.0	4.070000
7059	CHEMBL4585673	Nc1nc2ccccc2cc1CCC(=O)NC[C@@]12CCCO[C@@H]1CCOC2	inactive	369.465	2.4515	2.0	5.0	4.540000
7060	CHEMBL4546115	COc1ccc2c(c1)[C@@H](O)[C@@]1(CCN(C(=O)CCc3cc4c...	inactive	417.509	3.2666	2.0	5.0	4.260000
7061	CHEMBL1821813	Nc1nc2ccccc2cc1CCC(=O)NCC1CCCCC1	inactive	311.429	3.4461	2.0	3.0	4.510000

5739 rows × 8 columns

Fig (4) combined data frame (curated inhibitors and Lipinski descriptors) for further Chemical Space Analysis via Lipinski descriptor

Box plot: -

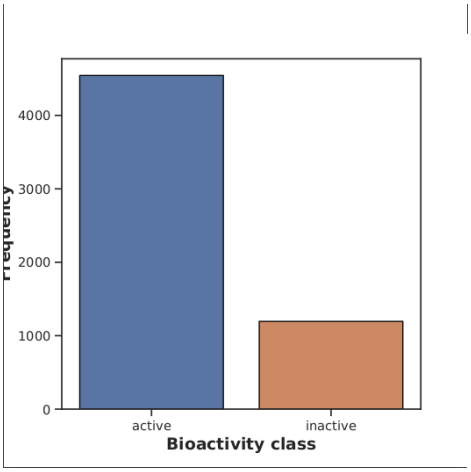


Fig (5)

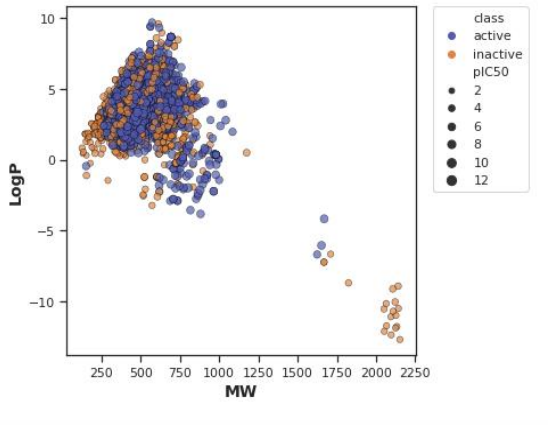


Fig (6)

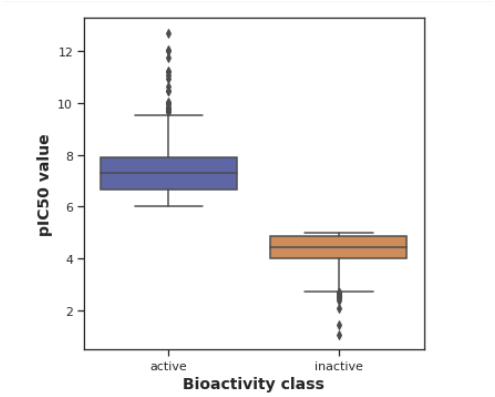


Fig (7)

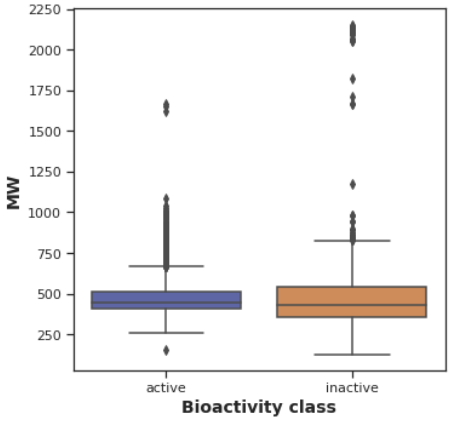


Fig (8)

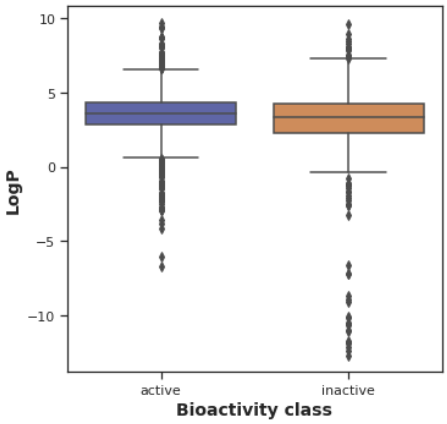


Fig (9)

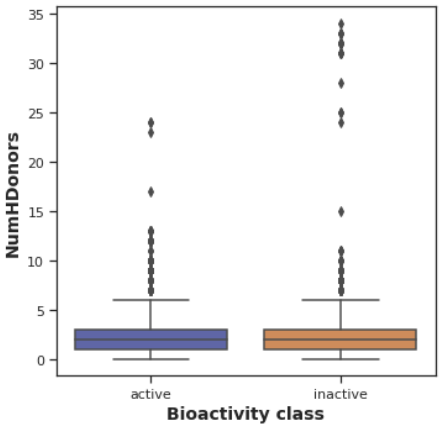
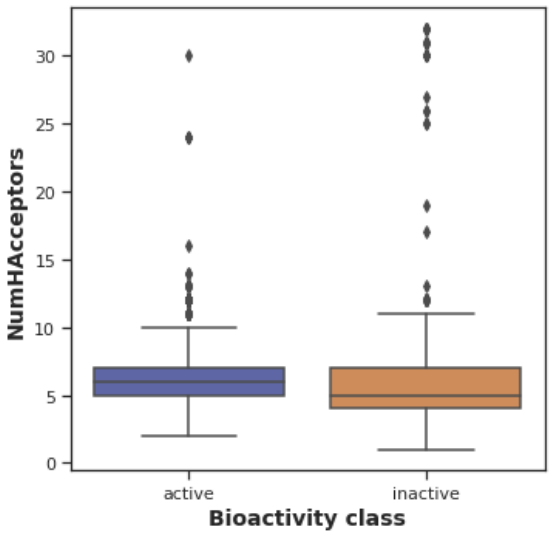


Fig (10)



Fig(11)

Fig (5) Frequency plot of the 2 bioactivity classes.

Fig (6) Scatter plot of MW versus LogP, the 2 bioactivity classes are spanning similar chemical spaces.

Fig (7) box plot of pIC50 values Vs bioactivity classes

Fig (8) box plot of molecular weight Vs bioactivity classes with P-value of 8.563718e12

Fig (9) box plot of LogP Vs bioactivity classes with P-value of 2.468324e-10

Fig (10) box plot of number of hydrogen donors in the compound Vs bioactivity classes with P-value of 0.000492

Fig (11) box plot of number of hydrogen acceptors in the compound Vs bioactivity classes with P-value of 5.697429e-12

Interpretation of Statistical Results

looking at pIC50 values, the actives and inactives displayed statistically significant difference, which is to be expected since threshold values (IC50 < 1,000 nM = Actives while IC50 > 10,000 nM = Inactives, corresponding to pIC50 > 6 = Actives and pIC50 < 5 = Inactives) were used to define actives and inactives, all

the 4 Lipinski's descriptors exhibited statistically significant difference between the actives and inactives.

3.3 Descriptor Calculation and Dataset Preparation

	Name	PubchemFP0	PubchemFP1	PubchemFP2	PubchemFP3	PubchemFP4	PubchemFP5	PubchemFP6	PubchemFP7	PubchemFP8	PubchemFP9	PubchemFP10	PubchemFP11	PubchemF
0	CHEMBL78946	1	1	1	1	0	0	0	0	0	1	1	1	
1	CHEMBL406146	1	1	1	1	0	0	0	0	0	1	1	1	
2	CHEMBL324109	1	1	1	1	0	0	0	0	0	1	1	1	
3	CHEMBL114147	1	1	1	1	0	0	0	0	0	1	1	1	
4	CHEMBL116826	1	1	1	1	0	0	0	0	0	1	1	1	
5	CHEMBL419949	1	1	1	1	0	0	0	0	0	1	1	1	
6	CHEMBL143239	1	1	1	1	0	0	0	0	0	1	1	1	
7	CHEMBL332948	1	1	1	1	0	0	0	0	0	1	1	1	
8	CHEMBL51386	1	1	1	1	0	0	0	0	0	1	1	1	
9	CHEMBL332260	1	1	1	1	0	0	0	0	0	1	1	1	
10	CHEMBL142715	1	1	1	1	0	0	0	0	0	1	1	1	
11	CHEMBL2370886	1	1	1	0	0	0	0	0	0	1	1	1	
12	CHEMBL326488	1	1	1	1	0	0	0	0	0	1	1	1	
13	CHEMBL290001	1	1	1	1	0	0	0	0	0	1	1	1	
14	CHEMBL114169	1	1	1	1	0	0	0	0	0	1	1	1	
15	CHEMBL324122	1	1	1	1	0	0	0	0	0	1	1	1	
16	CHEMBL309438	1	1	1	1	0	0	0	0	0	1	1	1	

Fig (12) The fingerprints provided by PubChem for the inhibitors data set as a binary representation of a library of 881 substructure features.

3.4 Regression Model with Random Forest

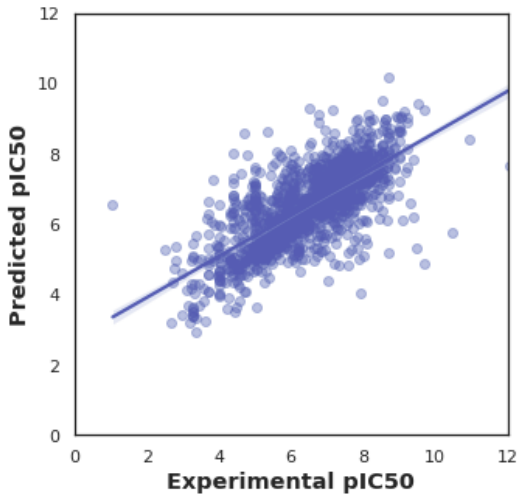


Fig (13) Scatter Plot of Experimental vs Predicted pIC50 Values.

3.5 model deployment

After evaluating the model

- the score of the model is 0.8721985246851728

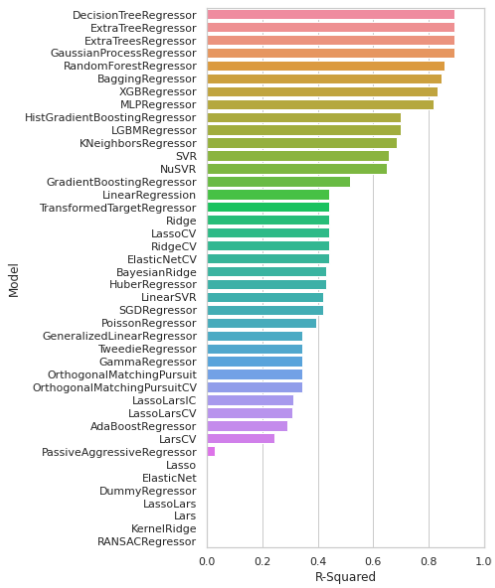
And the performance according to MSE and R squared.

- Mean squared error (MSE): 0.25
- Coefficient of determination (R^2): 0.87

3.6 Further analysis

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
DecisionTreeRegressor	0.89	0.89	0.45	0.25
ExtraTreeRegressor	0.89	0.89	0.45	0.22
ExtraTreesRegressor	0.89	0.89	0.45	11.44
GaussianProcessRegressor	0.89	0.89	0.45	15.14
RandomForestRegressor	0.85	0.86	0.52	8.82
BaggingRegressor	0.84	0.84	0.55	1.25
XGBRegressor	0.83	0.83	0.57	3.60
MLPRegressor	0.81	0.82	0.59	11.22
HistGradientBoostingRegressor	0.69	0.70	0.76	3.92
LGBMRegressor	0.69	0.70	0.76	0.74
KNeighborsRegressor	0.67	0.68	0.78	10.20
SVR	0.65	0.66	0.81	18.27
NuSVR	0.64	0.65	0.82	13.17
GradientBoostingRegressor	0.50	0.52	0.96	3.33
LinearRegression	0.43	0.44	1.03	0.21
TransformedTargetRegressor	0.43	0.44	1.03	0.18
Ridge	0.43	0.44	1.04	0.12

Fig (14) comparing ML algorithms.



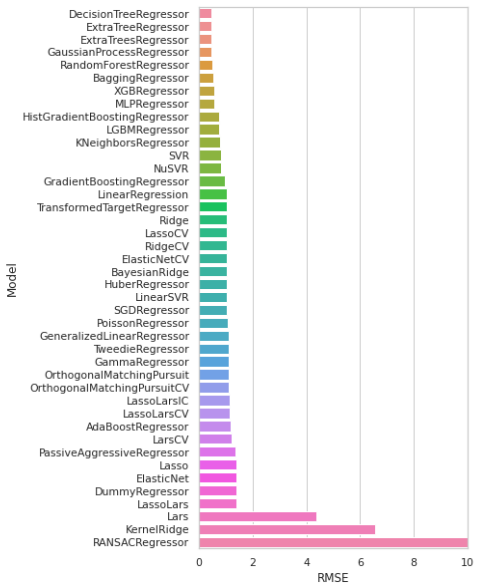


Fig (15) comparing ML according to R^2 Fig (16) comparing ML according to RMSE.

4 Discussion

Alzheimer's disease (AD) is the biggest cause of dementia in society and a gradually fatal neurodegenerative disease.

using BACE1 inhibitors as anti-Alzheimer agents is challenging as there are several failures in clinical trials and the possible reasons are firstly BACE1 inhibitors prevent amyloid production later in illness and may be more effective if used earlier, secondly the nature and complexity of AD (10), lastly the side effects from blocking completely the activity of BACE1. (10)

QSAR model prediction will not be proven right or wrong, unless there are some efforts of following-up experiments that can evaluate the model (11). And for further analysis we can optimize and validate the model by using MM/MD(Molecular Modeling/Molecular Docking), Pharmacophore models, another QSAR models and instead of machine learning algorithms we can train NN (Neural Networks) or deep learning approach to build our regression equation or switching to classification equation.

5 References

- **Articles: -**

1-. Wilson RS, Segawa E, Boyle PA et al (2012) The natural history of cognitive decline in Alzheimer â€™ s disease. Psychol Aging 27:1008–1017.

2- Alzheimer's Association (2013) Alzheimer's disease facts and figures. Alzheimer's Association, Chicago

3- Kumar A, Nisha CM, Silakari C, Sharma I, Anusha K, Gupta N et al (2016) Current and novel therapeutic molecules and targets in Alzheimer's disease. J Formos Med Assoc 115:3–10

4- Niedowicz DM, Nelson PT, Paul Murphy M(2011) Alzheimer's disease: pathological mechanisms and recent insights. Curr Neuropharmacol 9:674–684

5- Ghosh A, Osswald H (2014) BACE1 (β-Secretase) inhibitors for the treatment of Alzheimer's disease. Chem Soc Rev 43 (19):6765–6813

6- Hansch C, Hoekman D, Gao H. Comparative QSAR: Toward a deeper understanding of chemobiological interactions. Chem Rev 1996; 96: 1045 – 1075.

7- Chen X, Reynolds CH. Performance of similarity measures in 2D fragment-based similarity searching: comparison of structural descriptors and similarity coefficients. J Chem Inf Comput Sci. 2002 Nov-Dec;42(6):1407-14. doi: 10.1021/ci025531g. PMID: 12444738.

8- https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

9- https://en.wikipedia.org/wiki/Random_forest

10- Coimbra, J. R. et al. Highlights in BACE1 inhibitors for Alzheimer's disease treatment. Front. Chem. 6 (2018).

11- Chatila, Z. K. et al. BACE1 regulates proliferation and neuronal differentiation of newborn cells in the adult hippocampus in mice. eNeuro. 5(4) (2018)

- **Books: -**

12- - COMPUTATIONAL MODELING OF DRUGS AGAINST ALZHEIMER S DISEASE-HUMANA-Springer (2017)

13-. D. C. Young - Computational Drug Design_ A Guide for Computational and Medicinal Chemists-Wiley-Interscience (2009)