



University  
of Glasgow | School of  
Computing Science

Honours Individual Project Dissertation

# PREDICTING DRUGS THAT CAN CROSS THE BLOOD-BRAIN BARRIER

George Iniatis  
April 1, 2022

# Abstract

**Background:** The blood-brain barrier (BBB) prevents the vast majority of all compounds from entering the brain, protecting it from diseases and infections. However, it can also prevent useful therapeutics combating brain or central nervous system (CNS) related diseases from reaching their target.

**Motivation:** Checking whether a specific drug or compound can penetrate the BBB with experimental trials is expensive, time consuming and highly inefficient. Therefore, a predictive system can be a highly valuable tool that can test thousands of drugs and compounds in an inexpensive, fast and efficient manner.

**Aims:** This project aimed to create a new curated data set and then using it train machine learning models that make use of a drug's or compound's chemical properties to predict whether it can penetrate the BBB or not.

**Methods:** Both classification and regression models were trained using subsets of a curated data set of 2396 publicly available drugs and compounds and 6 hydrogen-bonding chemical descriptors. The classification models were further improved through the addition of the available side effects and indications to the chemical descriptors. Unfortunately this could not be replicated for the case of the regression models due to the subset size. All models were checked for robustness and evaluated using dummy models and holdout test sets.

**Results:** Our best classification model with just chemical descriptors used as features was the Random Forest Classifier which achieved an F1 score of 0.8506, an Accuracy of 0.8116, a Recall score of 0.9250, a Precision score of 0.7872 and a Matthews Correlation Coefficient of 0.6145. Our best classification model with chemical descriptors and a selection of side effects and indications as features was again the Random Forest Classifier, which achieved an F1 score of 0.8642, an Accuracy of 0.8406, a Recall score of 0.8750, a Precision score of 0.8537 and a Matthews Correlation Coefficient of 0.6716. Our best regression model with chemical descriptors used as features was the Support Vector Regression model, which achieved an R2 score of 0.4746, and a Negated Mean Absolute Error of -0.3968. A Streamlit web application was then created to showcase all of our work.

# Education Use Consent

I hereby grant my permission for this project to be stored, distributed and shown to other University of Glasgow students and staff for educational purposes.

Signature: George Iniatis Date: 1 April 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation	1
1.2	Objectives	1
1.3	Outline	1
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Traditional Approach Utilising Chemical Descriptors	3
2.1.1	Applicability Domain	3
2.1.2	Pitfalls	4
2.1.3	Improvements	4
2.2	Novel Approach Utilising Side Effects & Indications	5
2.3	Important Chemical Descriptors	5
2.4	Important Substructures	6
2.5	Valuable Strategies & Concepts Discovered	6
<b>3</b>	<b>Requirements Analysis</b>	<b>7</b>
3.1	Model Decisions	7
3.1.1	Machine Learning Library	7
3.1.2	Model Types	7
3.1.3	Classification Metrics	8
3.1.4	Regression Metrics	9
3.1.5	Model Evaluation Methods	10
3.1.6	Model Optimisation	10
3.1.7	Model Robustness	10
3.1.8	Model Interpretability	11
3.1.9	Presenting Findings	11
3.2	Data Set Decisions	11
3.2.1	Labels	11
3.2.2	Chemical Descriptors	11
3.2.3	Other Characteristics	12
3.2.4	Data Set Sources	12
<b>4</b>	<b>Design &amp; Implementation</b>	<b>13</b>
4.1	Data Set Creation Process	13
4.1.1	Combining Publicly Available Data Sets	13
4.1.2	Automated Google Searches	13
4.1.3	Medical APIs	14
4.1.4	Retrieving Chemical Descriptors, Side Effects & Indications	15

4.1.5	Final Adjustments	15
4.2	Model Training & Testing Process	15
4.2.1	Classification Models	16
4.2.2	Regression Models	17
4.3	Streamlit Web App	17
<b>5</b>	<b>Results &amp; Evaluation</b>	<b>19</b>
5.1	Data Exploration	19
5.1.1	Principal Component Analysis (PCA)	19
5.1.2	Chemical Descriptors Ranges	20
5.1.3	Important Side Effects & Indications	22
5.2	Models Performance	23
5.2.1	Classification Models Utilising Chemical Descriptors	23
5.2.2	Classification Models Utilising Chemical Descriptors, Side Effects & Indications	24
5.2.3	Regression Models Utilising Chemical Descriptors	25
5.3	Project Evaluation	25
<b>6</b>	<b>Conclusion</b>	<b>27</b>
6.1	Summary	27
6.2	Reflection	28
6.3	Future work	28
	<b>Bibliography</b>	<b>29</b>

# 1 | Introduction

This chapter will introduce the project on a high level and examine its motivations and objectives.

## 1.1 Motivation

The blood-brain barrier (BBB) can have a complex medical definition. However, for the sake of simplicity and the purposes of this project, we can think of it as a semi-permeable membrane that only allows molecules that are small or fat-soluble to enter the brain, and by definition, the central nervous system (CNS), while preventing larger ones from gaining entry (Woodruff and Götz 2017). This process is called passive diffusion, but it should also be noted that there are certain types of larger molecules, one of them being glucose, that can still enter the brain using different methods, such as through the usage of transport proteins (Woodruff and Götz 2017; Gao et al. 2017).

Just as the skull and cerebrospinal fluid, the fluid surrounding the brain, protect it from physical damage, the blood-brain barrier protects it from internal threats, such as harmful toxins and pathogens, that can cause infections and diseases (Woodruff and Götz 2017). This protective barrier prevents 98% of compounds from entering the brain. However, this can also mean preventing useful drugs from reaching their target, and this can be especially important when trying to deliver life-saving medicine like chemotherapy agents to combat brain tumours (Gao et al. 2017).

Manually checking whether a drug can penetrate the blood-brain barrier or not using laboratory experiments is expensive, time-consuming and can only be done one drug at a time, making the whole process highly inefficient (Singh et al. 2020). On the other hand, a prediction system can test thousands of drugs quickly and cheaply and can be used effectively as an early screening process, leading to a better allocation of time and resources for manual checks by discovering those drugs or compounds worth checking in more detail.

## 1.2 Objectives

The project aimed to gather publicly available data on drugs known to cross into the brain and those that cannot and place them into a new curated data set and then using this new data set train multiple machine learning models that use a drug's or compound's chemical properties to predict whether it can pass into the brain or not. The models should then be evaluated and compared in terms of robustness and performance, and a rudimentary system using these models should be constructed.

## 1.3 Outline

The dissertation consists of 6 chapters, including this one, where each discusses and examines a different stage of the project's life-cycle.

- **Chapter 2 - Background**  
Explores how other researchers have tackled the same problem and the valuable strategies, techniques, and knowledge discovered from their experiments.
- **Chapter 3 - Requirements Analysis**  
Discusses the high-level decisions made to narrow the scope of the project.
- **Chapter 4 - Design & Implementation**  
Explores how the data set and various machine learning models were created, using the decisions already discussed in Chapter 3.
- **Chapter 5 - Results & Evaluation**  
Discusses data exploration findings and the predictive performances of our trained models.
- **Chapter 6 - Conclusion**  
Summarises the project and discusses valuable lessons learned and any possible future work that could potentially improve our findings.

## 2 | Background

In this chapter we will explore how other researchers have tackled the same problem and the valuable strategies, techniques, and knowledge discovered from their experiments.

This background research was one of the first steps in the project's life-cycle and was instrumental in better understanding this previously unknown problem which included complex chemical and medical concepts, while also introducing us to key machine learning concepts and best practices.

Due to the important nature of the problem, as discussed in Section 1.1, there have been numerous attempts to construct classification and regression machine learning models using a variety of different methods, techniques, chemical properties and other characteristics that can be extracted from the drugs themselves, such as the drug's side effects and indications, what illness they treat. (Singh et al. 2020; Saber et al. 2020; Zhao et al. 2007; Gao et al. 2017; Zhang et al. 2008).

Classification models, being the ones that are more widely used, try to predict whether a particular compound or drug can pass the blood-brain barrier (BBB+) or not (BBB-), and Regression models try to predict the ratio between the concentration of a compound in the brain compared to the one in the blood. This ratio is called the Brain/Plasma ratio, but studies, more often than not, use its logarithmic version called logBB.

### 2.1 Traditional Approach Utilising Chemical Descriptors

The classic approach to solve the problem through the creation of classification models, as showcased by Singh et al. (2020), and regression models, as showcased by Zhang et al. (2008), was through the usage of special software that would use the SMILES notation of a drug or compound, that essentially describes its unique chemical structure, to produce thousands, if not more, chemical descriptors. Some of these special software included *Molconn-Z* (2022), *MOE* (2022), *Dragon* (2022), used by Zhang et al. (2008) and PaDEL-Descriptors (Yap 2011), used by Singh et al. (2020). Even if the descriptors with low predictive ability were removed, just as it was done in the case of Singh et al. (2020), hundreds of descriptors would still be left. These descriptors would then be used to train the various models.

Both Singh et al. (2020) and Zhang et al. (2008) concluded that a consensus model would provide superior predictive ability than a single model. This is because a consensus model combines multiple models and therefore mitigates overfitting problems associated with a single model. However, it naturally requires more computational power.

#### 2.1.1 Applicability Domain

Zhang et al. (2008) used a very interesting concept called applicability domain (AD) which essentially calculated the "Euclidean distance between each compound and its k-nearest neighbours" and compared it with a threshold, and if it exceeded that threshold, the prediction for that specific compound would not be made.



However, a case could be made that this takes away from the aim of building a predictive system used to test existing but also new drugs, drugs that could potentially be vastly different from their predecessors, leading to a large euclidean distance.

### 2.1.2 Pitfalls

The main pitfall to this traditional approach was the usage of special software that produced a massive number of descriptors, slowing down the models' training times while providing minimal improvements to performance as discussed by Zhao et al. (2007). Furthermore, the usage of these special, not widely available software would also make it very difficult for someone to use the created models to make predictions without having access to the specific one used to train them.

### 2.1.3 Improvements

Zhao et al. (2007) aimed to reduce the high number of descriptors needed to train classification models and using Algorithm Builder, a program developed by PharmaAlgorithms Inc, 19 molecular descriptors were calculated, showcased in Table 2.1. Small subsets of these descriptors were then used to train different models that achieve high predictive ability, effectively making the case that small numbers of descriptors are enough to create successful models.

The findings of Zhao et al. (2007) were further confirmed by Saber et al. (2020), making use of the same data set and choosing a subset of 8 uncorrelated chemical descriptors from the 19 previously used (Labelled as 6, 7, 8, 10, 11, 12, 13 and 19 in Table 2.1). It again proved that highly predictive models could be constructed using a tiny number of descriptors. These highly efficient combinations were uncovered through the use of sequential feature selection (SFS) and genetic algorithms (GA), with the study concluding that GA is a more robust approach than SFS in choosing the most relevant chemical descriptors.

Choosing a small number of highly predictive and widely available chemical descriptors leads to improved model training times and predictive performances.

**Table 2.1:** Table taken from Zhao et al. (2007) showcasing the 19 different chemical descriptors calculated.

no.	symbol	definition
1	E	excess molar refraction
2	S	polarizability/dipolarity
3	A	overall hydrogen-bond acidity
4	B	overall hydrogen-bond basicity
5	V	McGowan molecular volume in (mL/mol)/100
6	MW	molecular weight
7	PSA	polar surface area
8	logP	calculated octanol/water partition coefficient
9	logD(7.4)	octanol/water partition coefficient at pH = 7.4
10	NHD	number of hydrogen bonding donors
11	NHA	number of hydrogen bonding acceptors
12	pK <sub>a</sub> (acid)	pK <sub>a</sub> for acid
13	pK <sub>a</sub> (base)	pK <sub>a</sub> for base
14	Iv	indicator variable for carboxylic acid
15	F <sup>+</sup>	positively charged form fraction at pH = 7.4
16	F <sup>-</sup>	negatively charged form fraction at pH = 7.4
17	F <sup>±</sup>	zwitterion form fraction at pH = 7.4
18	F <sup>0</sup>	neutral form fraction at pH = 7.4
19	NRB	number of rotatable bonds

## 2.2 Novel Approach Utilising Side Effects & Indications

Gao et al. (2017) combined the usual approach of using the chemical characteristics of a drug or compound in order to predict its brain permeability with a new approach that makes use of their well-recorded side effects and indications found in the *SIDER* (2022) database.

As mentioned in Section 1.1, some larger molecules can pass into the brain using methods other than passive diffusion. These methods cannot be accurately described by the chemical characteristics of a drug or compound, but their recorded side effects and indications can capture them.

The side effects and indications were mapped to 43 subgroups, and SVM classification models were trained using multiple kernels. When both chemical descriptors and side effects and indications were available and combined, the models achieved significantly better performance than those based solely on the chemical descriptors, as showcased by Table 2.2

The study also used their created models on the *SIDER* (2022) database and identified 110 drugs that can potentially penetrate the blood-brain barrier and 1018 that potentially cannot.

**Table 2.2:** Table taken from Gao et al. (2017) showcasing the different model metrics when utilising the chemical descriptors, the clinical phenotypes (meaning the side effects and indications), and a combination of the two. There seems to be a substantial increase in performance for the model utilising the combination of the two. \*Prediction here means the average score achieved by cross-validation

T/P	Chemistry(C)		Clin. phenotype(CP)		C+CP	
	Training	Prediction	Training	Prediction	Training	Prediction
A.	.757±.043	.729±.022	.956±.019	.824±.022	.973±.016	.855±.023
AUC	.756±.043	.733±.022	.955±.020	.823±.023	.972±.017	.854±.024
F <sub>1</sub> <sup>8c</sup>	.754±.044	.725±.024	.956±.019	.824±.023	.973±.016	.854±.024
F <sub>1(+)</sub>	.743±.057	.715±.036	.959±.018	.836±.022	.975±.015	.863±.022
F <sub>1(-)</sub> <sup>#</sup>	.761±.058	.737±.034	.952±.022	.810±.027	.971±.018	.844±.028
p *		< e <sup>-90</sup>		< e <sup>-5</sup>		

Each data field shows average ± std from 1000 random split of Doniger et al.

<sup>8c</sup>: total samples macro weighted F<sub>1</sub>;

<sup>+</sup>: positive sample F<sub>1</sub>;

<sup>#</sup>: negative samples F<sub>1</sub>.

\*P-value for all of A., AUC and F<sub>1</sub><sup>8c</sup> of this column to last column (C+CP), T-test.

## 2.3 Important Chemical Descriptors

Zhang et al. (2008) after analysing the most frequent and vital descriptors, discovered that Polar surface area (PSA), Octanol/Water partition coefficient (logP) and the number of hydrogen bond donors and acceptor atoms were found to dominate the models. These findings were further confirmed and improved by the correlation study conducted by Zhao et al. (2007) which concluded that hydrogen-bonding properties (Labelled as 6–19 in Table 2.1) played a huge part in modelling brain permeability.

Singh et al. (2020) also illustrated the ranges that some of these crucial descriptors need to be within to successfully penetrate the blood-brain barrier and have an effect on the central nervous system (CNS), as showcased by Table 2.3.

However, it should be noted that even though CNS activity strongly implies BBB+, this cannot be said for the other way around, as BBB+ drugs and compounds can have no effect on the CNS.










**Table 2.3:** Table taken from Singh et al. (2020) showcasing the ranges that the most important chemical properties need to be within in order to be able to penetrate the blood-brain barrier and have an effect on the central nervous system

Class	MW	AlogP	LogD	PSA	HBAs	HBDs	RBs
CNS	<450	1.5–2.5	>0 and < 3	60-70 Å <sup>2</sup>	<7	<3	<8

## 2.4 Important Substructures

Singh et al. (2020) discovered a list of corroborated substructures, as showcased in Table 2.4, that were more prevalent in BBB+ compounds and drugs. Even though this was something out of the scope for this project, future studies could possibly find this information helpful.

**Table 2.4:** Table taken from Singh et al. (2020) showcasing fragment occurrence in BBB+ and BBB- compounds.

Substructure	BBB+ %occurrence	BBB- %occurrence
	1.1	0
	0.8	0
	0.5	0
	8.5	3.6
	7.9	2
	2.5	1.1
	1.9	0.9
	1.6	0.6
	1.4	0.9

## 2.5 Valuable Strategies & Concepts Discovered

All papers highlighted the importance of gathering as big a data set as possible, checking the robustness and statistical significance of models, and using holdout test sets to evaluate model predictive performance.

The usage of small data sets leads to models not generalising effectively for unseen drugs or compounds outside their chemical space and thus making them unsuitable for high-throughput screening (HTS), which is the main objective of building such a system (Singh et al. 2020).

Zhang et al. (2008) also introduced us to the bias problem that can be caused by a class imbalance in the data set, which is often the case. This is due to the fact that most researchers are trying to discover drugs and compounds that can successfully penetrate the brain and not the other way around, naturally leading to a higher number of drugs and compounds that can enter the brain vs those that cannot, which then leads to a much higher predictive ability for the BBB+ class.

## 3 | Requirements Analysis

This chapter will discuss the high-level decisions made to narrow the scope of the project.

Even though Section 1.2 very clearly specified the objectives that this project aimed for, it gave us free rein on what methods and techniques we utilised to achieve them.

These decisions, largely influenced by the background research found in Chapter 2, could be roughly split up into two distinct but interconnected sections that would determine the project's direction.

It should also be noted that some of these decisions, made early on in the project's life-cycle, were later revisited and updated accordingly, given our improved understanding of the problem and the methods we had used up to that point.

### 3.1 Model Decisions

The decisions that needed to be made for this section were the following:

- Which machine learning library would we use.
- What type of models would we build.
- What metrics and methods would we use to evaluate the models' predictive performance.
- How would these models be optimised.
- How would we test these models for robustness.
- How would we make the models' predictions more interpretable.
- How would we present our findings.

#### 3.1.1 Machine Learning Library

We decided to use scikit-learn (Pedregosa et al. 2011) to build our models, as it is one of the best machine learning libraries with excellent documentation and tutorials available.

#### 3.1.2 Model Types

We had initially chosen to only build classification models but after going through the publicly available data sets, provided by the studies and papers we already discussed in Chapter 2, and noticing that the logBB values of some drugs and compounds were also provided in addition to their ability to penetrate the blood-brain barrier or not, we chose to build both classification and regression models.

We decided to use multiple metrics for both types of models to evaluate their performance. This was done as it is generally viewed as good practice and gives a more complete view of the models' performance and limitations.

### 3.1.3 Classification Metrics

The classification models would be evaluated based on Precision, Recall, F1 Score, Accuracy and Matthews correlation coefficient (MCC). These metrics make use of the confusion matrix, which is a table, as shown in Figure 3.1, that records the total number of true positives (TP) and negatives (TN) and false positives (FP) and negatives (FN).

Precision measures how many of the positive predictions made by the classifier were actually positive (*Scikit-Learn Precision Score* 2022). It ranges from 0 to +1, with +1 being the best and 0 being the worst.

$$Precision = \frac{TP}{TP + FP}$$

Recall, also known as Sensitivity, measures how many of the actual positives were labelled as positive by our classifier (*Scikit-Learn Recall Score* 2022). It ranges from 0 to +1, with +1 being the best and 0 being the worst.

$$Recall = \frac{TP}{TP + FN}$$

F1 score is the harmonic mean of precision and recall (*Scikit-Learn F1 Score* 2022). Other variations exist where precision or recall can be given more or less weight. It ranges from 0 to +1, with +1 being the best and 0 being the worst.

$$F1 = \frac{2 * (precision * recall)}{precision + recall}$$

Accuracy measures the number of correct predictions made by the classifier (*Scikit-Learn Accuracy Score* 2022). It ranges from 0 to +1, with +1 being the best and 0 being the worst.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Matthews correlation coefficient (MCC), also known as Phi coefficient, essentially calculates the correlation between the predicted and true values (*Scikit-Learn - Matthews Correlation Coefficient* 2022). It takes into account the whole confusion matrix and is generally thought of as a particularly useful metric, even when the classes are imbalanced.

MCC has a range of -1 to +1. A coefficient of +1 indicates a perfect relation between predicted and true values, 0 that our model is randomly guessing, and -1 an inverse relationship between predicted and true values (*Statology - Matthews Correlation Coefficient* 2022).

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

**Figure 3.1:** Figure taken from Mohajon (2020) showcasing the confusion matrix for a binary classification problem.

### 3.1.4 Regression Metrics

The regression models would be evaluated based on Negated Mean Absolute Error and R2 Score.

The Mean Absolute Error (MAE) calculates exactly what its name suggests. It finds the absolute error, also known as the difference, between the predicted and the true value for all data points in a set, sums them up and then calculates their mean (*Scikit-Learn Mean Absolute Error* 2022). It has a range of 0 to  $+\infty$ , with 0 being the best score.

The Negated Mean Absolute Error just adds a negative sign in front of MAE. It has a range of  $-\infty$  to 0, with 0 again being the best score. We chose to use this variation so that all the metrics followed the same 'greater is better' principle, thus making their interpretation easier.

$$\text{Mean Absolute Error}(y, \hat{y}) = \left(\frac{1}{n}\right) \sum_{i=1}^n |y_i - \hat{y}_i|$$

where:

$n$  = number of samples

$y$  = true values

$\hat{y}$  = predicted values

The R2 score measures how good the regression line fits our data (*Scikit-Learn R2 Score* 2022). It compares the fit to a baseline model that essentially always predicts the mean of the true values (Dauria 2022). Generally, it ranges from 0 to +1, with +1 being the best score, but it can also become negative, indicating a worse model.

$$\text{R2 Score}(y, \hat{y}) = 1 - \left(\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}\right)$$

where:

$n$  = number of samples  
 $y$  = true values  
 $\hat{y}$  = predicted values  
 $\bar{y}$  = mean of true values

### 3.1.5 Model Evaluation Methods

We decided to use holdout test sets and dummy models to evaluate our models.

Holdout test sets are untouched subsets of the data, meaning they have not been used for training or validation purposes, that are specifically employed to estimate the model's performance on unseen real-world data.

Dummy models usually predict the most frequent class in our data, in the case of classification models, and the mean of our labels, what we are trying to predict, in the case of regression models, although there are multiple variations (*Scikit-Learn Dummy Classifier* 2022; *Scikit-Learn Dummy Regressor* 2022). These would serve as the baselines for our models.

### 3.1.6 Model Optimisation

The models' hyper-parameters would be tuned using scikit-learn's GridSearch function (*Scikit-Learn GridSearchCV* 2022) which makes use of cross-validation to optimise the models for a specific metric.

GridSearch essentially splits the data provided into multiple training and validation subsets, with the default being five unless otherwise specified. Then it iteratively goes through the pairs of training and validation subsets, training the model on the training subset using a combination of its hyper-parameters and then uses the trained model to make predictions on the validation set and calculates a score. Once all the pairs have been used, it calculates the mean validation score, moves on to the following combination of hyper-parameters, and repeats the process until all the combinations have been exhausted.

The function can then return the scores for all combinations of hyper-parameters for inspection and the model with the best mean validation score, along with the hyper-parameters used to train it.

The metrics we chose to optimise our models were the F1 score for the classification models and the R2 score for the regression ones.

### 3.1.7 Model Robustness

To check the robustness of the models, we would use scikit-learn's Permutation Test Score function. (*Scikit-Learn Permutation Test Score* 2022). This process is also known as y-scrambling.

This is essentially a statistical hypothesis test with a null hypothesis stating that a statistical relationship does not exist between the features and label using our model and an alternative hypothesis stating that a statistical relationship does indeed exist using our model.

Again this function just as GridSearch, found in Subsection 3.1.6, uses cross-validation. It iteratively goes through the pairs of training and validation sets and, for each permutation, randomly reorders the labels. Unless otherwise specified, this is repeated 100 times, removing any relationship between the features and the labels.

It returns a p-value which represents the probability that the model's predictions are just the result of random chance. The p-value is just the fraction of permutations that their mean validation score

is better than the mean validation score obtained using the original data that correctly matches the features with their labels (*Scikit-Learn Permutation Test Score User Guide* 2022). Generally, if this p-value is less than or equal to 0.05, we assume that we have a statistically significant result and reject the null hypothesis.

### 3.1.8 Model Interpretability

Model interpretability was one of the last and most challenging questions we tackled. So we decided to try and make our models more interpretable to shine some light in their inner workings and instill some confidence in their predictions, or at the very least help whoever is using them understand what led to that prediction.

We decided to use *ELI5* (2022) to examine the weights of the models' features and *Local Interpretable Model-Agnostic Explanations (LIME)* (2022) to explain how those features and their values led to a specific prediction.

### 3.1.9 Presenting Findings

The presentation of our findings was in flux until the very last stages of the project. We had previously discussed that, at the very least, we should produce a notebook or a simplistic web app. In the end, we decided to create a very simple *Streamlit* (2022) web application that showcases the project's processes and the models produced.

## 3.2 Data Set Decisions

The decisions that needed to be made for this category were the following:

- What would we use as our label.
- What chemical descriptors would we use.
- Could we make use of other drug characteristics.
- How would the data set be built, and more specifically, what sources and methods would we use to create it.

### 3.2.1 Labels

For our classification models, the label would be whether a drug or compound can penetrate the blood-brain barrier or not, designated by a 1 and a 0, respectively, and for our regression models, the label would be the logBB value.

### 3.2.2 Chemical Descriptors

After concluding our background research, discussed in Chapter 2, we had decided to use a tiny number of widely available chemical descriptors, specifically the same ones used by Saber et al. (2020).

Naturally, one of the world's largest and freely accessible databases of chemical information, *PubChem* (2022), was an excellent fit for our needs. However, we early on noticed that pKa (strongest base) and pKa (strongest acid) were not available, two of the eight chemical descriptors we wanted to use, and instead, we just had access to the pKa. So we made the decision to make use of this pKa descriptor, but we then observed that for the vast majority of drugs and compounds, this chemical descriptor was unavailable, so we finally settled on outright removing it from the data set altogether, reducing the number of chemical descriptors from 8 to 6.

The chemical descriptors we ended up using were:



- Molecular weight (MW)
- Topological polar surface area (TPSA)
- Octanol-water partition coefficient (XLogP)
- Number of hydrogen-bond donors (NHD)
- Number of hydrogen-bond acceptors (NHA)
- Number of rotatable bonds (NRB)

### 3.2.3 Other Characteristics

Inspired by the novel approach used by Gao et al. (2017) to solve the problem, as discussed in Section 2.2, we also decided to make use of the recorded side effects and indications of drugs and compounds stored in the *SIDER* (2022) database, to test whether the addition to the chemical descriptors improved the predictive performance of our models or not.

Side effects and indications pointing to central nervous system (CNS) issues should act as powerful indicators that a drug or compound can successfully penetrate the blood-brain barrier. However, just as it was already mentioned in Section 2.3 BBB+ drugs and compounds can have no effect on the CNS, making it harder to detect blood-brain barrier penetration solely through the usage of side effects and indications as features.

### 3.2.4 Data Set Sources

To create our data set, we decided to combine the publicly available data sets already provided by the studies and papers we previously discussed in Chapter 2 and augment them to suit our own needs.

As mentioned in Subsections 3.2.2 and 3.2.3 we would make use of the *PubChem* (2022) database to retrieve the chemical descriptors for each drug and compound and the *SIDER* (2022) database to retrieve the side effects and indications.

We also aimed to further expand the size of the data set using straightforward text mining techniques on google searches about blood-brain permeability. Unfortunately, this was proven to be noisy and unreliable. However, the techniques and methods developed were easily repurposed on searches performed on medical APIs, like *PubMed's E-Utilities API* (2022) and *Springer Nature's API* (2022), which proved to be much more reliable.

## 4 | Design & Implementation

This chapter will explore how the data set, and various machine learning models were created, using the decisions already discussed in Chapter 3.

### 4.1 Data Set Creation Process

Taking into consideration all the data set decisions discussed in Section 3.2, the data set creation process was split up into sequential steps.

#### 4.1.1 Combining Publicly Available Data Sets

To create our data set, we started by combining the publicly available data sets provided by Singh et al. (2020); Zhang et al. (2008); Gao et al. (2017); Zhang et al. (2008). All columns were removed except the ones with the SMILES notation, drug or compound name, experimental logBB value and blood-brain barrier permeability. We felt that this was the most appropriate strategy as all data sets used a vast array of chemical descriptors from different sources. Furthermore, this would allow us to retain the most essential information that we could then build upon.

Each academic paper’s Digital Object Identifier (DOI) was provided as the source for compounds and drugs. When it was not available, either a link to *PubMed* (2022) or *PubMed Central* (2022) was provided as the source.

It should also be noted that when the experimental logBB was available, BBB permeability was recalculated using the thresholds suggested by Li et al. (2005):

$$BBB + \text{ if } LogBB \geq -1$$

$$BBB - \text{ if } LogBB < -1$$

Given that the vast majority of the data set was created using the process discussed above, its quality is as good or bad as those data sets it has built upon.

#### 4.1.2 Automated Google Searches

In an effort to expand our data set, we initially decided to perform some basic text mining using automated google searches, using *Google Package* (2022), querying about specific drugs and compounds not present in our data set and their relation to the blood-brain barrier.

Our strategy was to gather the first 10 URLs returned from our google search and their HTML contents, and then using regular expressions, try to find matches for our query, hopefully leading to an apparent relationship between our specific drug or compound and the blood-brain barrier. These matches and their associated URLs were then loaded onto excel files and manually verified or rejected. Unfortunately, this strategy was proven to be too targeted and ineffective, returning completely irrelevant results most of the time.

Learning from our experience and discovering a class imbalance heavily tilting towards BBB+ drugs and compounds in our data set, we decided to broaden our search to try and reduce this imbalance, querying about drugs and compounds unable to cross the blood-brain barrier.

We followed the same process as before, but this time collecting as many URLs as we could before getting a "429: Too many requests error" and using multiple variations of queries and regular expressions, as showcased in Listing 4.1. Unfortunately, this strategy was also proven ineffective, returning results from online forums and sites clearly having nothing to do with peer-reviewed medical or chemical information. Even though we had collected some usable results from these strategies, we decided not to use them as someone could argue that they are incredibly unreliable and noisy. Discussion on how to overcome these shortcomings led us to the discovery of the medical APIs explored in the next section.

```
queries_and_regular_expressions = [

    ["\"not able to cross the blood brain barrier\"", ".*was not able to cross the  
blood.brain barrier.*"],
    ["\"not able to cross the bbb\"", ".*was not able to cross the bbb.*"],
    ["\"not able to penetrate the blood brain barrier\"", ".*was not able to  
penetrate the blood.brain barrier.*"],
    ["\"not able to penetrate the bbb\"", ".*was not able to not penetrate the  
bbb.*"],
    ["\"not able to pass through the blood brain barrier\"", ".*was not able to pass  
through the blood.brain barrier.*"],
    ["\"not able to path through the bbb\"", ".*was not able tp pass through the  
bbb.*"],
    ["\"not able to get through the blood brain barrier\"", ".*was not able to get  
through the blood.brain barrier.*"],
    ["\"not able to get through the bbb\"", ".*was not able to get through the bbb.*"]

]
```

**Listing 4.1:** A small sample of the list of queries and regular expressions used in an effort to expand our data set using Google Searches.

### 4.1.3 Medical APIs

*PubMed's E-Utilities API* (2022) was used to get abstracts from *PubMed* (2022) and academic papers from *PubMed Central* (2022) that matched multiple queries about drugs and compounds unable to penetrate the blood-brain barrier.

The various paragraphs of the abstracts and academic papers were extracted using XML parsing, and then, just as before, regular expressions were used to find matches based on our query. Finally, the matches were again loaded into excel files and manually verified.

*PubMed* (2022) searches produced 15 usable drugs and compounds, 14 being BBB- and 1 surprisingly being BBB+, from 35 matches.

*PubMed Central* (2022) searches produced 91 usable drugs and compounds, with all being BBB-, from 361 matches.

*Springer Nature's API* (2022) was used to get abstracts, articles and journals and then the same process was used just as in the case above.

Springer Meta V2 searches allowed us to access the abstracts of articles and journals not open to the public and produced 42 usable drugs and compounds, with 41 being BBB- and 1 being

BBB+, from 108 matches.

Springer Open access searches allowed us to access the publicly available full-text content of articles and journals and produced 109 usable drugs and compounds, with 106 being BBB- and 3 being BBB+, from 491 matches.

As mentioned above, the matches returned by the API searches had to be manually verified, and it should be mentioned that any human validated data is bound to have at least a few errors.

#### 4.1.4 Retrieving Chemical Descriptors, Side Effects & Indications

Once we had gathered all the drugs and compounds from the publicly available data sets and our API searches, *PubChem's API* (2022) was used to retrieve the chemical descriptors, already mentioned in Subsection 3.2.2, as well as some other helpful information such as the PubChem CID, the unique identifier for each compound in the *PubChem* (2022) database, the synonyms associated for each compound and its most common name, which was essentially the first synonym available. Even though the name of the compound was supplied in most of the cases by the data sets we had combined, we felt that it would make things easier down the road, and for anyone in the future that might utilise this work, if we used replaced it with the one in the PubChem database.

We mainly used the SMILES notation of each drug or compound to search the PubChem database for a match to accomplish this task. If the SMILES notation was unavailable, we used the drug or compound name to search for a match. Naturally, we discovered that using the drug or compound name was less effective in getting a match from the database than the SMILES format, as the latter is unique for each drug or compound, whereas multiple drugs or compounds can use the same name, which can lead to confusion.

It should also be noted that some drugs and compounds do not have a name or any synonyms associated with them.

Once the synonyms were retrieved for a specific drug or compound, they were looked up in the *SIDER* (2022) database. If a synonym was found in the *SIDER* database, we then retrieved the *SIDER* CID and the associated side effects and indications.

*SIDER* offered two types of labels for side effects and indications, Lowest Level Terms (LLTs), taken directly from the official description of drugs, and Preferred Terms (PTs), which simplify multiple LLTs. We decided to use the PT side effects and indications since they condensed multiple LLTs.

#### 4.1.5 Final Adjustments

Duplicates, drugs and compounds that could not be discovered and those without all chemical descriptors available were removed. This process decreased the size of the data set from 3748 drugs and compounds to 2396, with 1751 being BBB+ and 645 being BBB-. The data set was then finally sorted by drug name.

Table 4.1 showcases the first 10 rows of the final version of the data set.

## 4.2 Model Training & Testing Process

Our training and testing process was straightforward and used consistently for both classification and regression models.

We would first create a pipeline (*Scikit-Learn Pipeline* 2022) containing a model and a standard scaler (*Scikit-Learn StandardScaler* 2022). Pipelines simplify our workflow by stacking several

Table 4.1: The first 10 rows of our finalised data set.

SMILES	Name	PubChem_CID	SIDER_CID	Source	logBB	Class	Class_Verbose	MW	TPSA	XLogP	NHD	NHA	NRB	Synonyms	Side_Effects	Indications
CN1CCC2= (+)-Bicuculline		10237	-	https://doi.c		0	Does not Pass	367	66.5	2.6	0	7	1	["(+)-Bicuc	-	-
CCC[C@] (+)-Secobarbital		31143	-	https://doi.c	0.2	1	Passes	238	75.3	2	2	3	5	["(+)-Secol	-	-
CNCCC(Oc (+)-Duloxetine		122252	-	https://doi.c		1	Passes	297	49.5	4.3	1	3	6	["(+)-Dulo	-	-
C (11c)methane		297	-	https://doi.c		1	Passes	16	0	0.6	0	0	0	["(11c)met	-	-
c1ccccc1C (2S,3S)-2-(Diphenyl		9931510	-	https://doi.c	0.37	1	Passes	413	24.5	5.4	1	3	7	["(2S,3S)-2	-	-
CCS(C)SC4 (2S,5R,6R)-6-(3-Amir		20056959	-	https://doi.c		0	Does not Pass	394	138	-0.9	3	6	3	["(2S,5R,6f	-	-
CC1OC1P((3-Methyloxiran-2-y		3417	-	https://doi.c		1	Passes	138	70.1	-1.4	2	4	1	["(3-Methy	-	-
OCC(=O)C (8S,9S,10R,13S,14S,1		11245343	-	https://doi.c	-0.18	1	Passes	360	94.8	1.6	3	5	2	["(8S,9S,1C	-	-
CC(C)C(C)C (R)-Stiripentol		39524	-	https://doi.c		1	Passes	234	38.7	3.6	1	3	3	["(R)-Stirip	-	-
CN[C@H](R)-propylhexedrine		111165	-	https://doi.c	1.08	1	Passes	155	12	3.5	1	1	3	["(R)-prop	-	-

pre-processing steps that are applied to our data before they are passed to a chosen model as features. In our case, we are just using a standard scaler as a pre-processing step that normalises our features by "removing the mean and scaling to unit variance".

We would then pass this pipeline to GridSearch (*Scikit-Learn GridSearchCV* 2022) as the estimator along with the specific model parameters we wanted to tune, the metrics we wanted to be returned, the metric we wanted to optimise for and the number of cross-validation folds. This process is showcased by Figure 4.1 to make things clearer.

All models were optimised using a 10-fold cross-validation GridSearch except in the case of the dummy classifier, dummy regressor and linear regression models.

Once optimised, the models were then tested for robustness, using permutation testing as already discussed in 3.1.7, and used to make predictions on their respective test set.

```

0.1s
pipe = Pipeline(
    [
        ('scale', StandardScaler()),
        ('model', RandomForestClassifier(random_state=0))
    ]
)
pipe.get_params()

model = GridSearchCV(estimator=pipe,
                      param_grid={
                          'model__n_estimators': [100, 200, 400, 600, 800],
                          'model__criterion': ['gini', 'entropy'],
                          'model__max_features': ['auto', 'sqrt', 'log2'],
                          'model__class_weight': [None, 'balanced', 'balanced_subsample'],
                      },
                      scoring=grid_search_scoring_dict,
                      refit='f1', # Optimise for F1 Score
                      return_train_score=False,
                      cv=10,
                      n_jobs=-1)

```

Figure 4.1: The optimisation process used for the majority of our models, showcased with a Random Forest Classifier.

### 4.2.1 Classification Models

Classification models were split up into two categories, one using just the chemical descriptors as features and the other using the chemical descriptors and a selection of one-hot encoded side effects and indications as features.

A selection was used instead of every single side effect and indication supplied by *SIDER* (2022) to not only improve the training times of our models but to also potentially discover which side effects and indications play the most critical role in deciding blood-brain barrier permeability. To achieve this, we decided to use scikit-learn's recursive feature elimination with cross-validation

(RFECV) function (*Scikit-Learn RFECV* 2022) which does exactly what it suggests, with a random forest classifier, optimising for F1 score, and a 10-fold cross-validation. As a result, RFECV managed to reduce our features from 4353 to 217, keeping all 6 chemical descriptors, 196 of the side effects and 15 of the indications.

We decided to split our classification models into two separate categories because we were interested in discovering whether the addition of side effects and indications to the chemical descriptors improved their predictive performance. Therefore, we needed a common holdout test set to achieve this.

This holdout test set was a 20% stratified split of the 345 drugs and compounds that had chemical descriptors, side effects and indications available (199 BBB+, 146 BBB-), produced with scikit-learn's Train Test Split function (*Scikit-Learn Train Test Split* 2022).

The training sets for both categories excluded the drugs and compounds present in the test set, with the first category making use of the whole data set, while the second one used the subset of drugs and compounds having chemical descriptors, side effects and indications available.

Finally, the classification models we decided to train for both categories were:

- Dummy Classifier
- Logistic Regression
- Support Vector Classifier
- K-Nearest Neighbour Classifier
- Random Forest Classifier
- Decision Tree Classifier
- Stochastic Gradient Descent Classifier

## 4.2.2 Regression Models

Regression models were solely trained using chemical descriptors as there were not enough drugs and compounds with logBB values available and chemical descriptors, side effects and indications to split them into two categories just as it was done in the case of the classification models.

The training and test sets were again produced using scikit-learn's Train Test Split function (*Scikit-Learn Train Test Split* 2022). The holdout test set was a 20% stratified split of the 401 drugs and compounds that had logBB values available (360 BBB+, 41 BBB-). The remaining 80% was used as the training set.

Finally the regression models we decided to train were:

- Dummy Regressor
- Linear Regression
- Support Vector Regression
- K-Nearest Neighbour Regressor
- Random Forest Regressor
- Decision Tree Regressor
- Stochastic Gradient Descent Regressor

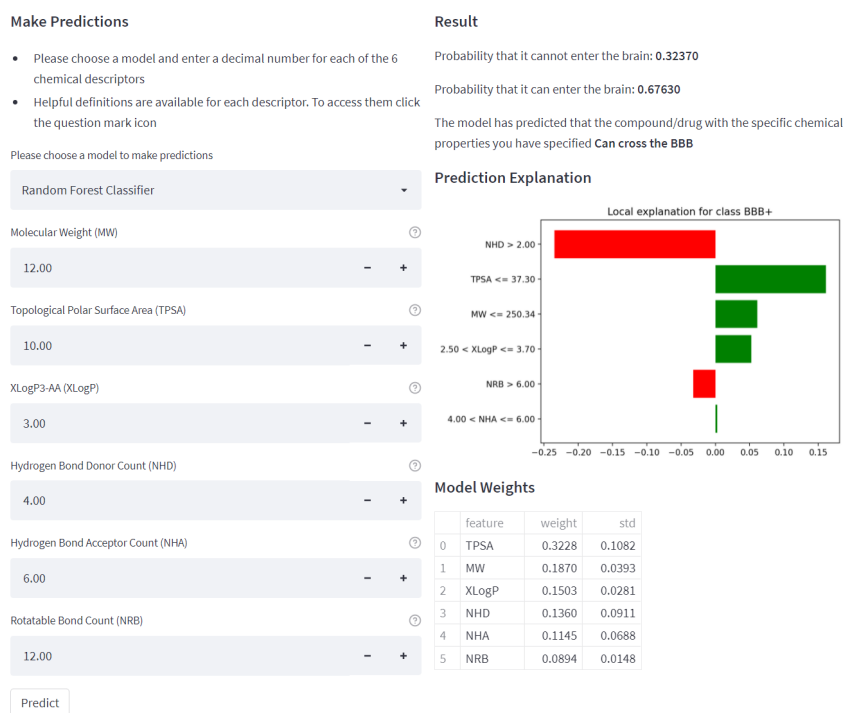
## 4.3 Streamlit Web App

The web app was created in the final weeks of the project to present a synopsis of our work and primarily showcase our models and to allow users to make predictions with them. A strong emphasis was also placed on model interpretability, helping users understand what led to a specific

prediction by a model, which was already briefly discussed in Subsection 3.1.8. However, it should be noted that these interpretability tools are not available for all models.

Figure 4.2 showcases a prediction made by our trained Random Forest Classifier using our web application, and the interpretability tools, which a user can use to try and understand the model's prediction.

You can access the web application using this link



**Figure 4.2:** Our trained Random Forest Classifier used to make a prediction in our Streamlit Web App.

## 5 | Results & Evaluation

This chapter will discuss data exploration findings and the predictive performances of our trained models.

### 5.1 Data Exploration

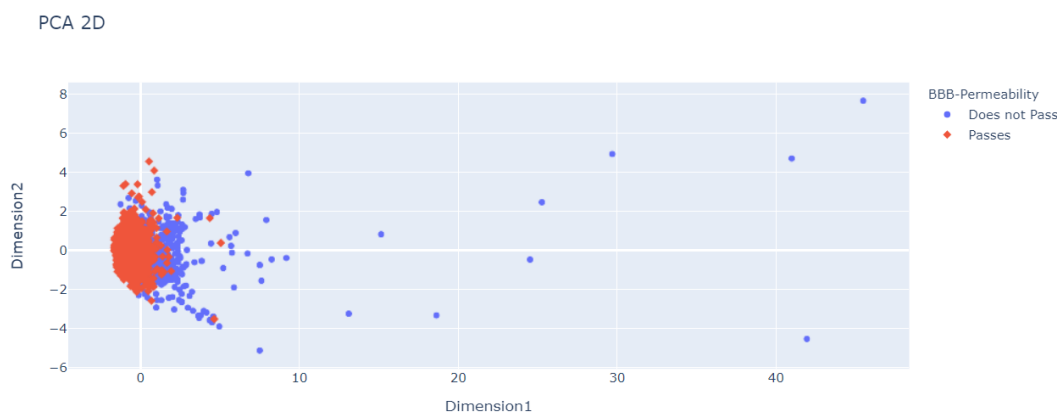
After creating our data set, we wanted to explore our classes' spread and the distributions of the chemical descriptors. This section also discusses the essential side effects and indications discovered.

#### 5.1.1 Principal Component Analysis (PCA)

Principal component analysis is a dimensionality reduction method used to project data in a lower-dimensional space (*Scikit-Learn PCA* 2022). We used it to project our chemical descriptor data for all drugs and compounds into a two-dimensional space in order to plot it and examine the spread of our two different classes.

Looking at Figure 5.1 we could see that BBB+ drugs and compounds were mostly closely packed together, with a few exceptions that appeared to be almost identical in some cases with drugs and compounds that are labelled as BBB-. These could be mislabelled drugs and compounds, but given the nature of the problem, we could not identify these with any confidence given our lack of medical and chemical knowledge. It could also be the case that these drugs and compounds could be using different methods to penetrate the blood-brain barrier, mentioned in Section 1.1, that are not accurately described just through their chemical properties.

BBB- drugs and compounds appeared to be more spread out with some very notable outliers that are later confirmed in Subsection 5.1.2



*Figure 5.1: Principal component analysis plot after projecting the data set to a two-dimensional space.*

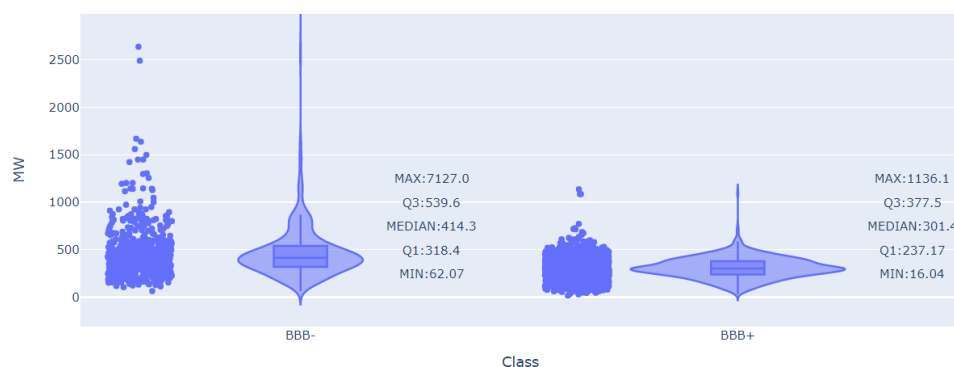


### 5.1.2 Chemical Descriptors Ranges

To explore our chemical descriptor data ranges, we decided to create violin plots combined with box plots.

It should be noted that the following discussion will explore the distributions found in our specific data set, a sample of drugs and compounds which may or may not be representative of the whole population of drugs and compounds that can penetrate the blood-brain barrier and of those that cannot.

Molecular Weight (MW) distribution, showcased by Figure 5.2, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had a MW in the range of 237.2–377.5. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had a MW in the range 318.4–539.6. A note should be made of the very large outliers appearing in the BBB- drugs and compounds, with a notable example having a MW of 7127. It appears that a smaller MW is preferable for BBB penetration. However, this is not always the case, as we can see BBB- drugs and compounds with a very low MW.



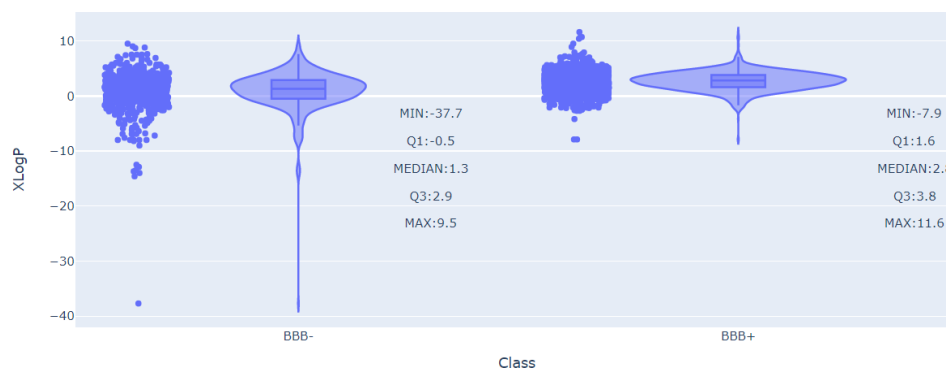
**Figure 5.2:** Violin plots for the distribution of molecular weight (MW) values of each class

Topological polar surface area (TPSA) distribution, showcased by Figure 5.3, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had a TPSA in the range of 32.3–74.6. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had a TPSA in the range 79.3–197. Again we can see some very large outliers in the BBB- drugs and compounds, with a notable example having a TPSA of 2860. Just as in the case of MW, it appears that a smaller TPSA is preferable.



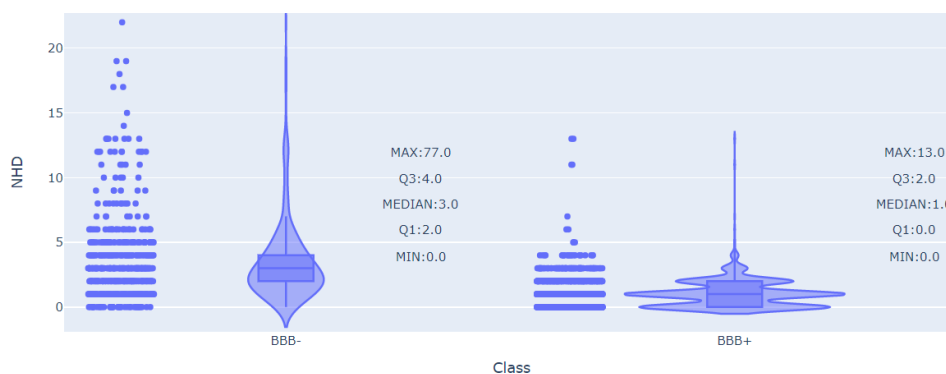
**Figure 5.3:** Violin plots for the distribution of topological polar surface area (TPSA) values of each class

Octanol-water partition coefficient (XLogP) distribution, showcased by Figure 5.4, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had an XLogP in the range of 1.6–3.8. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had an XLogP in the range -0.5–2.9. Again we can see some significant outliers in the BBB- drugs and compounds, with a notable example having an XLogP of -37.7. It seems that a small but positive XLogP is preferable.



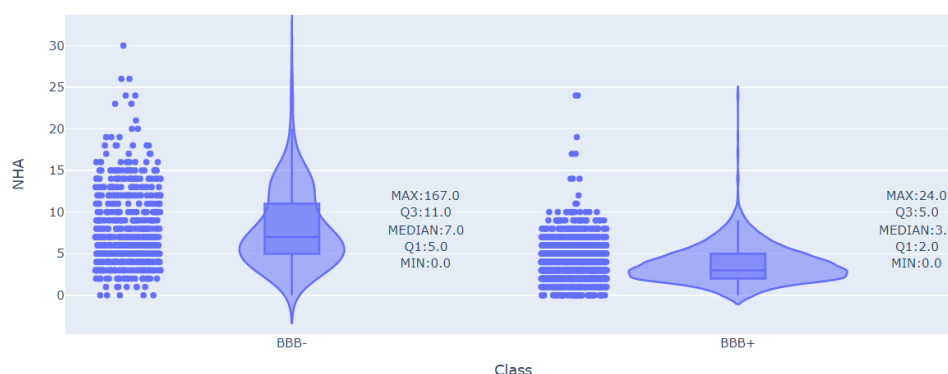
**Figure 5.4:** Violin plots for the distribution of octanol-water partition coefficient (XLogP) values of each class

Hydrogen-bond donors (NHD) distribution, showcased by Figure 5.5, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had an NHD in the range of 0–2. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had an NHD in the range 2–4. Again we can see some significant outliers in the BBB- drugs and compounds, with a notable example having an NHD of 77. It seems that a smaller NHD is preferable.



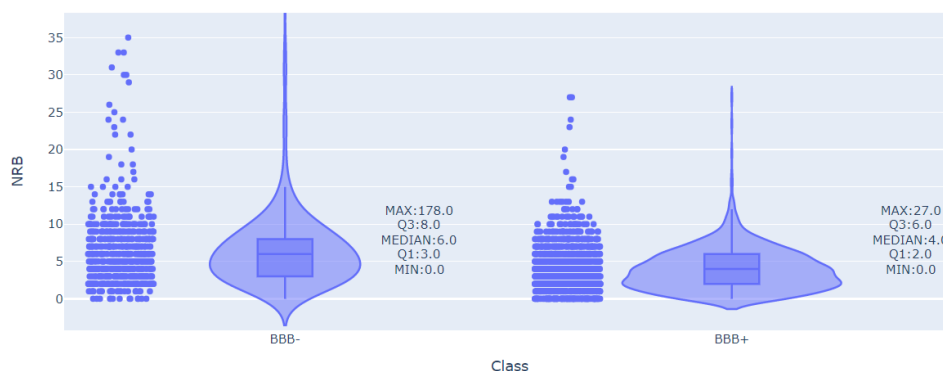
**Figure 5.5:** Violin plots for the distribution of the number of hydrogen-bond donors (NHD) of each class

Hydrogen-bond acceptors (NHA) distribution, showcased by Figure 5.6, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had an NHA in the range of 2–5. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had an NHA in the range 5–11. Again we can see some significant outliers in the BBB- drugs and compounds, with a notable example having an NHA of 167. Just as in the case of NHD, it appears that a smaller NHA is preferable.



**Figure 5.6:** Violin plots for the distribution of the number of hydrogen-bond acceptors (NHA) of each class

The number of rotatable bonds (NRB) distribution, showcased by Figure 5.7, showed that the greater number of drugs and compounds that could pass the blood-brain barrier had an NRB in the range of 2–6. Whereas the majority of drugs and compounds that could not pass the blood-brain barrier had an NRB in the range 3–8. Again we can see some significant outliers in the BBB- drugs and compounds, with a notable example having an NRB of 178. Just as in the case of NHD and NHA, it appears that a smaller NRB is preferable. However, it is a bit unclear due to a considerable overlap between the two classes.



**Figure 5.7:** Violin plots for the distribution of the number of rotatable bonds (NRB) of each class

### 5.1.3 Important Side Effects & Indications

As we have previously discussed in Subsection 4.2.1, RFECV was used to primarily reduce the large number of side effects and indications used by some classification models.

Table 5.1 showcases all the different side effects deemed highly important by RFECV. Some of them, such as depressed levels of consciousness or a coma, naturally seem sensible choices as they seem directly connected to the central nervous system. However, others, such as dry skin or influenza, seem less so.

Table 5.2 showcases all the different indications deemed highly important by RFECV. In our opinion, all indications returned seem sensible.

**Table 5.1:** Top side effects according to RFECV. \*Please ignore the <NA> entries, these were used as padding when creating the table.

	Side Effects_0	Side Effects_1	Side Effects_2	Side Effects_3	Side Effects_4	Side Effects_5	Side Effects_6	Side Effects_7
0	Abdominal discomfort	Abdominal distension	Abdominal pain	Abnormal dreams	Accommodation disorder	Acute coronary syndrome	Aggression	Agitation
1	Agranulocytosis	Albuminuria	Alopecia	Amenorrhoea	Amnesia	Anaemia	Anaphylactic shock	Anaphylactoid reaction
2	Angina pectoris	Angioedema	Aplastic anaemia	Apnoea	Arrhythmia	Arthralgia	Aspartate aminotransferas...	Asthenia
3	Ataxia	Atrophy	Balance disorder	Blindness	Blood and lymphatic syste...	Blood bilirubin increased	Body temperature increased	Bradycardia
4	Bronchospasm	Cardiac failure	Chest pain	Chills	Coma	Confusional state	Conjunctivitis	Constipation
5	Contusion	Convulsion	Coordination abnormal	Cough	Decreased appetite	Delusion	Depressed level of conscio...	Dermatitis
6	Dermatitis bullous	Dermatitis contact	Dermatitis exfoliative	Diabetes mellitus	Diarrhoea	Diplopia	Discomfort	Disorientation
7	Disturbance in sexual arou...	Dizziness	Drug eruption	Drug withdrawal syndrome	Dry mouth	Dry skin	Dysarthria	Dysgeusia
8	Dyspepsia	Dyspnoea	Dysuria	Eczema	Eosinophilia	Erectile dysfunction	Erythema	Erythema multiforme
9	Euphoric mood	Extrapyrmidal disorder	Extravasation	Eye pruritus	Fatigue	Feeling abnormal	Flatulence	Fluid retention
10	Flushing	Fungal infection	Galactorrhoea	Gastritis	Gastrointestinal disorder	Gastrointestinal haemorrh...	Gastrointestinal pain	Glossitis
11	Gynaecomastia	Haemolytic anaemia	Haemorrhage	Hallucination	Headache	Hepatic function abnormal	Hepatic necrosis	Hepatitis
12	Hepatobiliary disease	Hepatotoxicity	Hostility	Hyperhidrosis	Hyperkinesia	Hypersensitivity	Hypertension	Hypoesthesia
13	Hypoglycaemia	Hypokalaemia	Hypokinesia	Hypotension	Ill-defined disorder	Immune system disorder	Infection	Influenza
14	Insomnia	Irritability	Jaundice	Jaundice cholestatic	Lethargy	Leukopenia	Libido decreased	Loss of consciousness
15	Lymphadenopathy	Malaise	Malnutrition	Mediastinal disorder	Menopausal symptoms	Mental disability	Mental disorder	Mood swings
16	Muscle rigidity	Muscle spasms	Muscular weakness	Musculoskeletal discomfort	Musculoskeletal pain	Myalgia	Mydriasis	Myocardial infarction
17	Myocardial ischaemia	Nasal congestion	Nausea	Nervous system disorder	Nervousness	Neuroleptic malignant syn...	Neuropathy peripheral	Neutropenia
18	Oedema	Optic neuritis	Oropharyngeal pain	Orthostatic hypotension	Pain	Palpitations	Pancytopenia	Paraesthesia
19	Pharyngitis	Photosensitivity reaction	Pneumonia	Pollakiuria	Pruritus	Psychotic disorder	Pulmonary embolism	Pulmonary oedema
20	Purpura	Rash	Rash maculo-papular	Renal failure	Renal failure acute	Renal impairment	Respiratory depression	Respiratory failure
21	Sensory loss	Shock	Skin disorder	Skin hyperpigmentation	Somnolence	Stevens-Johnson syndrome	Stupor	Syncope
22	Systemic lupus erythemat...	Tachycardia	Tension	Thinking abnormal	Thrombocytopenia	Tinnitus	Tooth disorder	Toxic epidermal necrolysis
23	Tremor	Tubulointerstitial nephritis	Urinary retention	Urticaria	Vaginal infection	Vaginal inflammation	Vasculitis	Vertigo
24	Vision blurred	Vomiting	Weight decreased	Weight increased	<NA>	<NA>	<NA>	<NA>

**Table 5.2:** Top indications according to RFECV.

	Indications
0	Agitation
1	Anxiety
2	Asthma
3	Breast cancer
4	Hypersensitivity
5	Hypertension
6	Hypotension
7	Infection
8	Insomnia
9	Neoplasm malignant
10	Oedema
11	Prostatitis
12	Renal failure
13	Renal impairment
14	Schizophrenia

## 5.2 Models Performance

This section discusses the robustness and predictive performances of the different trained models and makes some comparisons.

### 5.2.1 Classification Models Utilising Chemical Descriptors

Tables 5.3 and 5.4 showcase the training and testing scores of all our classification models that only used chemical descriptors as their features. All of our models appear robust, with a permutation test p-value < 0.05, making them statistically significant.

Even though six models were produced, excluding the Dummy Classifier, which is only used as a baseline, we would not recommend the use of the Logistic Regression and Support Vector Classification models. These models are only slightly better than the Dummy Classifier and have a very low MCC score, making them just marginally better than a coin flip.

Our best model seems to be the Random Forest Classifier, outperforming all the other models in every metric. A very close second would be the K-Nearest Neighbour Classifier.

**Table 5.3:** Training performance of classification models with chemical descriptors used as features.

	Set	Model	Accuracy	Recall	Precision	F1	Matthews Correlation Coefficient	Permutation Testing P-Value
0	Training Set (2327 Compounds)	Dummy Classifier	0.7353	1.0000	0.7353	0.8475	0.0000	<NA>
1	Training Set (2327 Compounds)	Logistic Regression	0.8496	0.9620	0.8536	0.9041	0.5839	0.0099
2	Training Set (2327 Compounds)	Support Vector Classification	0.8470	0.9714	0.8451	0.9035	0.5742	0.0099
3	Training Set (2327 Compounds)	K-Nearest Neighbour Classifier	0.8719	0.9585	0.8792	0.9168	0.6533	0.0099
4	Training Set (2327 Compounds)	Random Forest Classifier	0.8702	0.9468	0.8854	0.9148	0.6520	0.0099
5	Training Set (2327 Compounds)	Decision Tree Classifier	0.8234	0.8901	0.8729	0.8810	0.5400	0.0099
6	Training Set (2327 Compounds)	Stochastic Gradient Descent Classifier	0.8539	0.9644	0.8567	0.9069	0.5968	0.0099

**Table 5.4:** Testing performance of classification models with chemical descriptors used as features.

	Set	Model	Accuracy	Recall	Precision	F1	Matthews Correlation Coefficient
0	Test Set (69 Compounds)	Dummy Classifier	0.5797	1.0000	0.5797	0.7339	0.0000
1	Test Set (69 Compounds)	Logistic Regression	0.6087	0.8750	0.6140	0.7217	0.1516
2	Test Set (69 Compounds)	Support Vector Classification	0.6087	0.8750	0.6140	0.7217	0.1516
3	Test Set (69 Compounds)	K-Nearest Neighbour Classifier	0.7826	0.9250	0.7551	0.8315	0.5562
4	Test Set (69 Compounds)	Random Forest Classifier	0.8116	0.9250	0.7872	0.8506	0.6145
5	Test Set (69 Compounds)	Decision Tree Classifier	0.7971	0.8500	0.8095	0.8293	0.5807
6	Test Set (69 Compounds)	Stochastic Gradient Descent Classifier	0.6812	0.8000	0.6957	0.7442	0.3322

### 5.2.2 Classification Models Utilising Chemical Descriptors, Side Effects & Indications

Tables 5.5 and 5.6 showcase the training and testing scores of all our classification models that used chemical descriptors, side effects and indications as their features. All of our models appear to be robust, with a permutation test p-value < 0.05, making them statistically significant.

The addition of side effects and indications to the chemical descriptors appears to have substantially improved the performance of almost all models, except in the case of the Decision Tree Classifier, where its performance decreased.

Again, our best model seems to be the Random Forest Classifier, outperforming all the other models in every metric. A very close second would be the K-Nearest Neighbour Classifier.

**Table 5.5:** Training performance of classification models with chemical descriptors, side effects and indication used as features.

	Set	Model	Accuracy	Recall	Precision	F1	Matthews Correlation Coefficient	Permutation Testing P-Value
0	Training Set (276 Compounds)	Dummy Classifier	0.5761	1.0000	0.5761	0.7310	0.0000	<NA>
1	Training Set (276 Compounds)	Logistic Regression	0.7939	0.8175	0.8245	0.8170	0.5870	0.0099
2	Training Set (276 Compounds)	Support Vector Classification	0.8115	0.8792	0.8147	0.8427	0.6181	0.0099
3	Training Set (276 Compounds)	K-Nearest Neighbour Classifier	0.7533	0.8413	0.7658	0.7974	0.4951	0.0099
4	Training Set (276 Compounds)	Random Forest Classifier	0.8181	0.8917	0.8267	0.8500	0.6450	0.0099
5	Training Set (276 Compounds)	Decision Tree Classifier	0.7284	0.7854	0.7621	0.7687	0.4424	0.0099
6	Training Set (276 Compounds)	Stochastic Gradient Descent Classifier	0.7825	0.8238	0.8071	0.8110	0.5593	0.0099

**Table 5.6:** Testing performance of classification models with chemical descriptors, side effects and indication used as features.

	Set	Model	Accuracy	Recall	Precision	F1	Matthews Correlation Coefficient
0	Testing Set (69 Compounds)	Dummy Classifier	0.5797	1.0000	0.5797	0.7339	0.0000
1	Testing Set (69 Compounds)	Logistic Regression	0.7101	0.7000	0.7778	0.7368	0.4191
2	Testing Set (69 Compounds)	Support Vector Classification	0.7681	0.7750	0.8158	0.7949	0.5295
3	Testing Set (69 Compounds)	K-Nearest Neighbour Classifier	0.8261	0.8500	0.8500	0.8500	0.6431
4	Testing Set (69 Compounds)	Random Forest Classifier	0.8406	0.8750	0.8537	0.8642	0.6716
5	Testing Set (69 Compounds)	Decision Tree Classifier	0.7391	0.7250	0.8056	0.7632	0.4779
6	Testing Set (69 Compounds)	Stochastic Gradient Descent Classifier	0.7391	0.7750	0.7750	0.7750	0.4647

### 5.2.3 Regression Models Utilising Chemical Descriptors

Tables 5.7 and 5.8 showcase the training and testing scores of all our regression models that only used chemical descriptors as their features. All of our models, except the Decision Tree Regressor, appear to be robust, with a permutation test p-value < 0.05, making them statistically significant.

Our best model seems to be the Support Vector Regression, outperforming all the other models in every metric.

**Table 5.7:** Training performance of regression models with chemical descriptors used as features.

	Set	Model	Negated Mean Absolute Error	R2 Score	Permutation Testing P-Value
0	Training Set (320 Compounds)	Dummy Regressor	-0.6208	0.0000	<NA>
1	Training Set (320 Compounds)	Linear Regression	-0.5010	0.1994	0.0099
2	Training Set (320 Compounds)	Support Vector Regression	-0.4666	0.2694	0.0099
3	Training Set (320 Compounds)	K-Nearest Neighbour Regressor	-0.5063	0.2272	0.0099
4	Training Set (320 Compounds)	Random Forest Regressor	-0.4989	0.1251	0.0099
5	Training Set (320 Compounds)	Decision Tree Regressor	-0.6212	-1.2564	0.1683
6	Training Set (320 Compounds)	Stochastic Gradient Descent Regressor	-0.4977	0.2620	0.0099

**Table 5.8:** Testing performance of regression models with chemical descriptors used as features.

	Set	Model	Negated Mean Absolute Error	R2 Score
0	Test Set (81 Compounds)	Dummy Regressor	-0.5276	-0.0205
1	Test Set (81 Compounds)	Linear Regression	-0.4258	0.3468
2	Test Set (81 Compounds)	Support Vector Regression	-0.3968	0.4746
3	Test Set (81 Compounds)	K-Nearest Neighbour Regressor	-0.4663	0.2541
4	Test Set (81 Compounds)	Random Forest Regressor	-0.4244	0.3023
5	Test Set (81 Compounds)	Decision Tree Regressor	-0.4450	0.2644
6	Test Set (81 Compounds)	Stochastic Gradient Descent Regressor	-0.4338	0.3284

## 5.3 Project Evaluation

All objectives specified in Section 1.2 were successfully achieved.

We managed to create a substantial data set that was used to create both classification and regression models using a very small number of chemical descriptors, further confirming the conclusions

reached by Zhao et al. (2007); Saber et al. (2020) that models predicting the blood-brain barrier permeability of drugs and compounds can be built using a minimal number of hydrogen-bonding chemical descriptors. All of our models, except for a single one, were statistically significant however some were only marginally better than a coin flip as discussed in Subsection 5.2.1

The project also managed to validate the conclusion reached by Gao et al. (2017) that the addition of side effects and indications to chemical descriptors substantially improved the predictive performance of models. All but one of our classification models' predictive performances improved by adding side effects and indications, even though we used a different technique and a smaller number of chemical descriptors. Our Random Forest Classifier even achieved somewhat similar performance to the Support Vector Machine trained by Gao et al. (2017), showcased by Table 2.2.

## 6 | Conclusion

This chapter summarises the project and discusses valuable lessons learned and any possible future work that could potentially improve upon our findings.

### 6.1 Summary

The brain is surrounded by a semi-permeable boundary that prevents many pathogens from getting in. However, it can also stop many useful drugs from entering the brain. This is especially important when trying to deliver critical therapeutics, such as chemotherapy, to brain tumours. Therefore, accurate prediction of whether a drug will easily cross the blood-brain barrier is a valuable tool for developing and testing new drugs for various diseases.

This project aimed to gather publicly available data on drugs known to cross into the brain and those that cannot and place them into a new data set and then, using that new data set, train machine learning models that use a drug's or compound's chemical descriptors to predict whether it can pass into the brain or not.

A data set of 2396 publicly available compounds and drugs was gathered from various academic papers and medical APIs, subsets of which were used to train both classification and regression models. Various models were trained for both types of models using a tiny number of chemical descriptors, checked for robustness and evaluated using test sets and dummy models. In the case of classification models, these were further improved by including the available side effects and indications of each drug and compound. Unfortunately, this could not be replicated for the regression models due to the small size of available drugs having all the necessary information we required.

A Streamlit web application was then created to present a synopsis of our work and primarily showcase our models, allowing users to use them to make predictions. A strong emphasis was also placed on model interpretability, potentially helping users understand what led to a specific prediction by a model. A strong chemical or medical knowledge is not necessarily needed, but it would definitely be a plus.

Our best classification model with just chemical descriptors used as features was the Random Forest Classifier which achieved an F1 score of 0.8506, an Accuracy of 0.8116, a Recall score of 0.9250, a Precision score of 0.7872 and a Matthews Correlation Coefficient of 0.6145.

Our best classification model with chemical descriptors and a selection of side effects and indications as features was again the Random Forest Classifier, which achieved an F1 score of 0.8642, an Accuracy of 0.8406, a Recall score of 0.8750, a Precision score of 0.8537 and a Matthews Correlation Coefficient of 0.6716.

Our best regression model with chemical descriptors used as features was the Support Vector Regression model, which achieved an R2 score of 0.4746, and a Negated Mean Absolute Error of -0.3968.

The created models can be used efficiently to predict the blood-brain permeability of thousands of already existing or new drugs and compounds. However, these predictions should be taken as



guidelines for further research, possibly even experimental trials in order to confirm them, and not as absolutes, as no model can be perfect. Given the important health consequences of a false positive or negative no one should trust a model completely, no matter how accurate it claims to be.

## 6.2 Reflection

This project allowed us to work in-depth with previously unfamiliar concepts, mainly bioinformatics and machine learning, and to learn multiple new skills, best practices, and techniques that we can build upon in the future.

Looking back at the project, we should have definitely used a common testing set with one of the background papers in Chapter 2 so we could directly compare our models' performance and to see if we achieved a better predictive performance or not, expand our data set even more, and spent more time and energy analysing the errors of our models. However, overall we believe the project to be a success, achieving all of its specified objectives in a professional and responsible manner.

## 6.3 Future work

Even though it could be argued that the project was reasonably successful, a few areas of improvement could be explored further in the future.

The data set could be expanded with the help of professionals with chemical and medical knowledge that could potentially point out any mislabelled entries, which could then be used to retrain the models or create new ones, even potentially utilising deep learning to produce even better models with greater predictive performances.

The already trained models could be improved by analysing their blind spots, the chemical areas of drugs and compounds that are consistently misclassified or produce a high error value. Some preliminary error analysis of the predictions made by our models found what appear to be groupings, suggesting that there is some pattern that could be looked at in more detail. These systematic weaknesses could be negated by further exploring the errors, but some in-depth chemical knowledge would be required, which we did not have during the project's life-cycle.

## Bibliography

- Dauria, E. (2022), ‘Medium – looking at r-squared’. Last accessed: 2022-02-19.  
**URL:** <https://medium.com/@erika.dauria/looking-at-r-squared-721252709098>
- Dragon (2022). Last accessed: 2022-02-13.  
**URL:** [http://www.taletе.mi.it/products/dragon\\_description.htm](http://www.taletе.mi.it/products/dragon_description.htm)
- ELI5 (2022). Last accessed: 2022-02-16.  
**URL:** <https://eli5.readthedocs.io/en/latest/overview.html>
- Gao, Z., Chen, Y., Cai, X. and Xu, R. (2017), ‘Predict drug permeability to blood–brain–barrier from clinical phenotypes: drug side effects and drug indications’, *Bioinformatics* **33**, 901–908.  
**URL:** <https://academic.oup.com/bioinformatics/article/33/6/901/2623044>
- Google Package (2022). Last accessed: 2022-02-26.  
**URL:** <https://pypi.org/project/google/>
- Li, H., Yap, C. W., Ung, C. Y., Xue, Y., Cao, Z. W. and Chen, Y. Z. (2005), ‘Effect of selection of molecular descriptors on the prediction of blood–brain barrier penetrating and nonpenetrating agents by statistical learning methods’, *Journal of Chemical Information and Modeling* **45**, 1376–1384.  
**URL:** <https://pubs.acs.org/doi/abs/10.1021/ci050135u>
- Local Interpretable Model-Agnostic Explanations (LIME) (2022). Last accessed: 2022-02-16.  
**URL:** <https://lime-ml.readthedocs.io/en/latest/>
- MOE (2022). Last accessed: 2022-02-13.  
**URL:** <https://www.chemcomp.com/>
- Mohajon, J. (2020), ‘Towards data science – confusion matrix for binary classification’. Last accessed: 2022-02-16.  
**URL:** <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>
- Molconn-Z (2022). Last accessed: 2022-02-13.  
**URL:** <http://www.edusoft-lc.com/molconn/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), ‘Scikit-learn: Machine learning in Python’, *Journal of Machine Learning Research* **12**, 2825–2830.
- PubChem (2022). Last accessed: 2022-02-16.  
**URL:** <https://pubchem.ncbi.nlm.nih.gov/>
- PubChem’s API (2022). Last accessed: 2022-02-23.  
**URL:** <https://pubchemdocs.ncbi.nlm.nih.gov/pug-rest>

- PubMed* (2022). Last accessed: 2022-02-25.  
**URL:** <https://pubmed.ncbi.nlm.nih.gov/>
- PubMed Central* (2022). Last accessed: 2022-02-25.  
**URL:** <https://www.ncbi.nlm.nih.gov/pmc/>
- PubMed's E-Utilities API* (2022). Last accessed: 2022-02-21.  
**URL:** <https://www.ncbi.nlm.nih.gov/books/NBK25501/>
- Saber, R., Mhanna, R. and Rihana, S. (2020), 'A machine learning model for the prediction of drug permeability across the blood-brain barrier: a comparative approach'.  
**URL:** <https://www.researchsquare.com> <https://www.researchsquare.com/article/rs-29117/v1>
- Scikit-Learn Accuracy Score* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html)
- Scikit-Learn Dummy Classifier* (2022). Last accessed: 2022-02-20.  
**URL:** <https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html>
- Scikit-Learn Dummy Regressor* (2022). Last accessed: 2022-02-20.  
**URL:** <https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyRegressor.html>
- Scikit-Learn F1 Score* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)
- Scikit-Learn GridSearchCV* (2022). Last accessed: 2022-02-16.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)
- Scikit-Learn - Matthews Correlation Coefficient* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews\\_corcoef.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corcoef.html)
- Scikit-Learn Mean Absolute Error* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/model\\_evaluation.html#mean-absolute-error](https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error)
- Scikit-Learn PCA* (2022). Last accessed: 2022-03-05.  
**URL:** <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- Scikit-Learn Permutation Test Score* (2022). Last accessed: 2022-02-16.  
**URL:** [scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.permutation\\_test\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.permutation_test_score.html)
- Scikit-Learn Permutation Test Score User Guide* (2022). Last accessed: 2022-02-20.  
**URL:** [https://scikit-learn.org/stable/modules/cross\\_validation.html#permutation-test-score](https://scikit-learn.org/stable/modules/cross_validation.html#permutation-test-score)
- Scikit-Learn Pipeline* (2022). Last accessed: 2022-02-27.  
**URL:** <https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>
- Scikit-Learn Precision Score* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html)
- Scikit-Learn R2 Score* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/model\\_evaluation.html#r2-score](https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score)
- Scikit-Learn Recall Score* (2022). Last accessed: 2022-02-19.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html)
- Scikit-Learn RFECV* (2022). Last accessed: 2022-02-27.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.RFECV.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html)
- Scikit-Learn StandardScaler* (2022). Last accessed: 2022-02-27.  
**URL:** <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

- Scikit-Learn Train Test Split* (2022). Last accessed: 2022-02-27.  
**URL:** [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)
- SIDER* (2022). Last accessed: 2022-02-13.  
**URL:** <http://sideeffects.embl.de/>
- Singh, M., Divakaran, R., Konda, L. S. K. and Kristam, R. (2020), 'A classification model for blood brain barrier penetration', *Journal of Molecular Graphics and Modelling* **96**.
- Springer Nature's API* (2022). Last accessed: 2022-02-21.  
**URL:** <https://dev.springernature.com/>
- Statology - Matthews Correlation Coefficient* (2022). Last accessed: 2022-02-19.  
**URL:** <https://www.statology.org/matthews-correlation-coefficient-python/>
- Streamlit* (2022). Last accessed: 2022-02-16.  
**URL:** <https://streamlit.io/>
- Woodruff, A. and Götz, J. (2017), 'What is the blood-brain barrier? - queensland brain institute'.  
**URL:** <https://qbi.uq.edu.au/brain/brain-anatomy/what-blood-brain-barrier>
- Yap, C. W. (2011), 'Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints', *Journal of Computational Chemistry* **32**, 1466–1474.  
**URL:** <https://onlinelibrary.wiley.com/doi/full/10.1002/jcc.21707>
- Zhang, L., Zhu, H., Oprea, T. I., Golbraikh, A. and Tropsha, A. (2008), 'Qsar modeling of the blood-brain barrier permeability for diverse organic compounds', *Pharmaceutical Research* **25**:8 **25**, 1902–1914.  
**URL:** <https://link.springer.com/article/10.1007/s11095-008-9609-0>
- Zhao, Y. H., Abraham, M. H., Ibrahim, A., Fish, P. V., Cole, S., Lewis, M. L., Groot, M. J. D. and Reynolds, D. P. (2007), 'Predicting penetration across the blood-brain barrier from simple descriptors and fragmentation schemes', *Journal of Chemical Information and Modeling* **47**, 170–175.