**Developing machine learning models to predict the Hansen solubility parameters**

Supervised by Jiayun Pang

**By Alexander Pine**

**Table of Contents**

[Abstract 0](#_Toc199318427)

[Chapter 1. Introduction 1](#_Toc199318428)

[1.3 Aims of the project 12](#_Toc199318429)

[Chapter 2. Methodology 13](#_Toc199318430)

[2.6 The Python code 14](#_Toc199318431)

[Chapter 3. Results and Discussion 18](#_Toc199318432)

[Chapter 4. Conclusion, future work and general remarks 22](#_Toc199318433)

[References 23](#_Toc199318434)

# Abstract

Solubility parameters are important for pharmaceutical formulations, paint formulations and new material development.  There is a need to improve the accuracy of solubility calculations, and to be able to make rapid predictions of the solubility of new molecular structures. In this project, a range of Python plugins, and open-source codes have been used to develop a Lasso linear regression machine learning model to predict the Hansen solubility parameters (HSP) - δd, δp and δh, which represents dispersion forces, dipole-permanent dipole forces and hydrogen bonding respectively with the intention of making faster and more accurate prediction in solubility. A dataset of 87 and 193 solvent molecules was used respectively with the molecules, molecular fingerprints, charge density, electrostatics, and shape/size chosen as the features for the model. Overall, the model gives an adequate fitting, and the predicted Hansen parameters of many molecules are in fair agreement with experimentally determined HSP. To evaluate the accuracy of the model, the coefficient of determination R2 was calculated, which identifies how close the data is fitted relative to a given fitted regression line. It was observed that for the dataset of 87 molecules, the R2 values are 0.56, 0.67 and 0.86 for δd, δp and δh, respectively. In contrast, for the 193-molecule data set, the R2 values are 0.32, 0.54 and 0.77 for δd, δp and δh, respectively, with all three R2 values decreasing compared to the smaller dataset. Changes in the size of the dataset have a significant effect on the R2 values. It is also worth noting that the predicted δh performs better in comparison to δd and δp. Also, the model in the current study gives a better performance in terms of R2 when comparing the previous study carried out by Sanchez et al using the same dataset, although it is not straightforward as to why, as the details of the code used for their Lasso model was not published. For future work, the sizes of the datasets could be randomised, and larger datasets could be considered. Other machine learning models, such as KernelRidge and Regularised Greedy Forest (RGF) could be explored; as well as the Gaussian processes regression model which takes a probabilistic approach. Lastly, different ways to generate the current descriptors or the use of entirely different descriptors could be explored to see whether they will improve the model.

# Chapter 1. Introduction

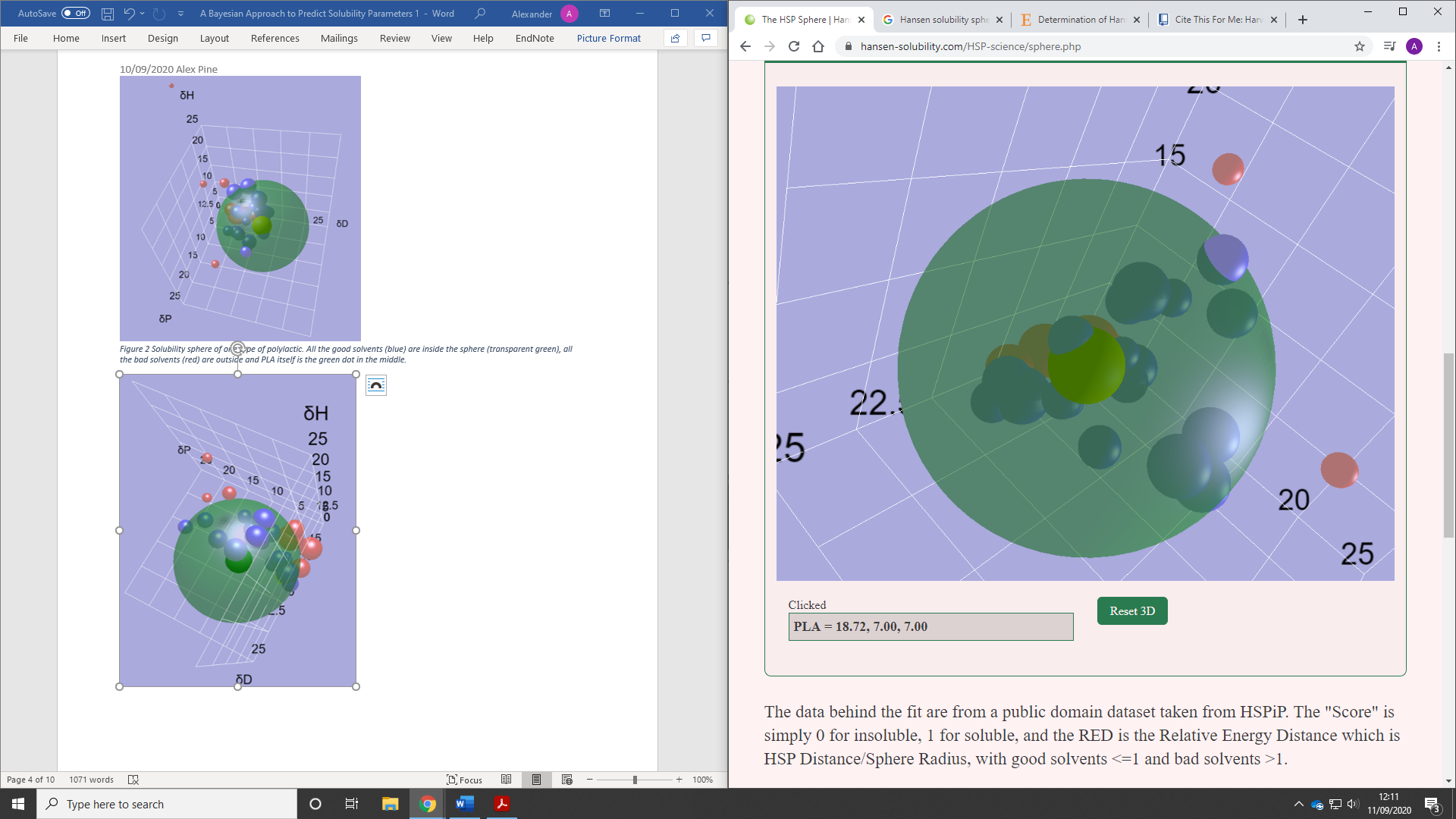
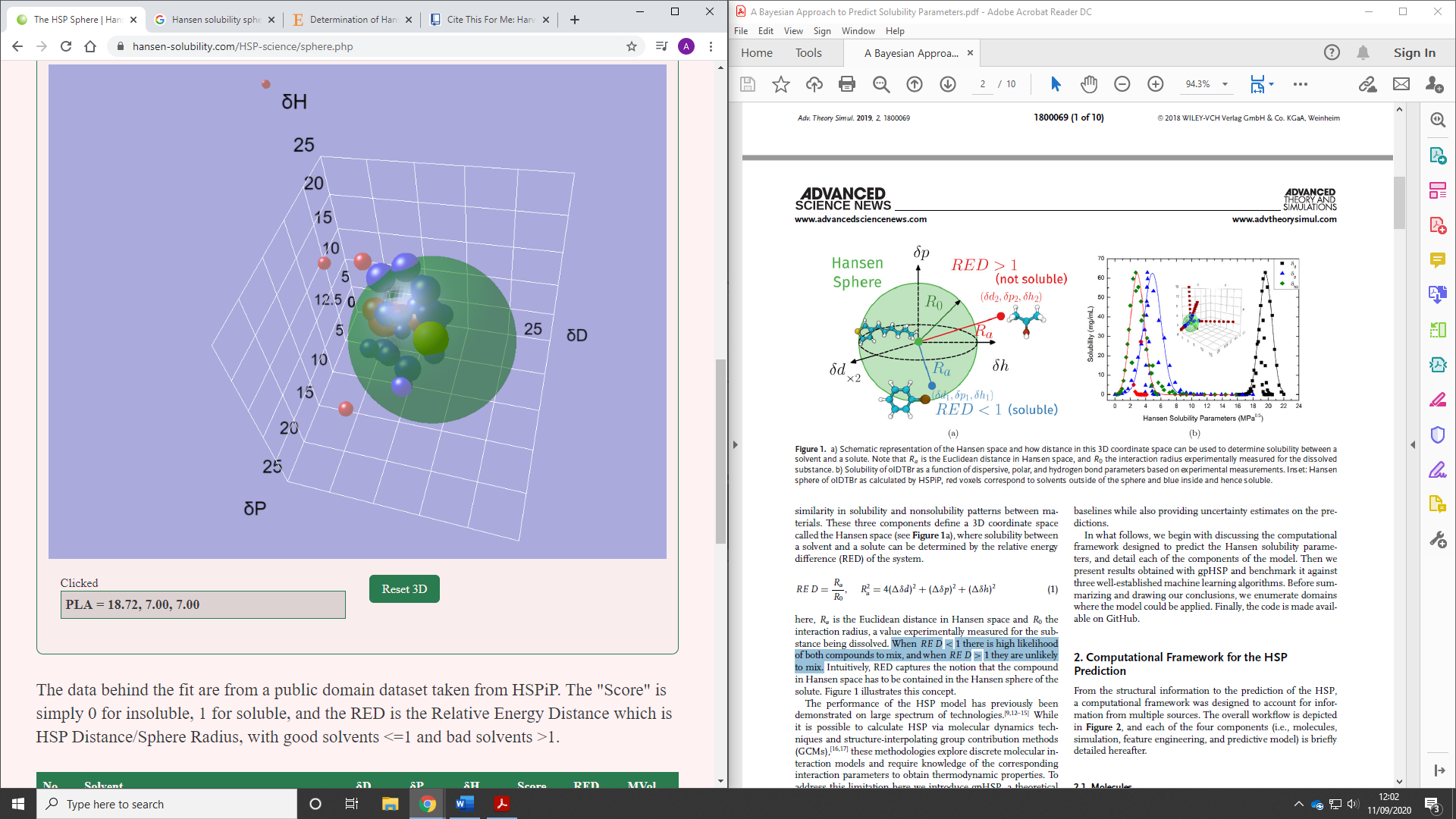
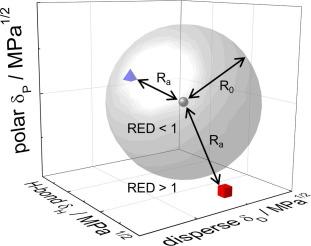
**1.1 Overview of the Hansen solubility parameters**

Solubility can be defined as the maximum quantity of a chemical that can be fully dissolved in a given volume of solution. [13] Predicting solubility can be a challenging task since it depends on various physicochemical factors. Some of the more important factors to be considered are: the interactions between solute and solvent, the nature of intrinsic intermolecular forces, and whether a molecule is polar or non-polar. [14] These principles were built upon by Hildebrand and Scott in 1949. [12] Then, in 1967, the pre-existing parameters were retouched by Charles M. Hansen, which became the Hansen solubility parameters (HSP). The Hansen solubility parameters were built on the foundation of “like dissolves like”, for example caffeine, a polar solute dissolving in water, a polar solvent. [10] There are three variables that make up the Hansen solubility parameters: δh, δp, δd, which account for hydrogen bonding, dipole-permanent dipole forces, and dispersion forces respectively. These parameters were made to better understand how the nature of intermolecular forces affect solubility, [11] thus having vast applications in pharmaceutical, paint and material science-related industries. [15]

When considering the HSP, the user will obtain a *R*a value (Equation 1). This is directly influenced by whether the solvent and solute have congruent solubility. Therefore, if the *R*a value is large, the solute will have low solubility in the solvent because the intermolecular forces between the solute and solvent are dissimilar. [16]

Equation 1. The Hansen solubility parameter and its dependency on the three variables δh*,*δp*,*δd. The subscripts 1 and 2 indicate the molecules 1 and 2, respectively.

The Hansen sphere given in Figure 1 below is used to consider the relative energy difference (RED) of the two materials in the system RED = Ra/R0. As seen in Figure 1, the Hansen space offers 3-D coordinates, in a Cartesian plane. The R0 value is the given radius of the Hansen sphere for a specific molecule, in this case, it is polylactic acid (PLA seen in Figure 1b). For a given R0 and Ra of two materials, the user can calculate the relative energy difference (RED) for that system. If the system is “well miscible”, then a RED value will be less than 1. However, if RED value is greater than 1, this implies the two materials will be immiscible, or the system cannot stabilise and therefore there is not a feasible way to find a solvent for the inquiring solute. [17]



(a)

(c)

(b)

*Figure 1. a) Visualisation of 3-D cartesian plane showing the three different contributing parameters δD, δP and δh [41]. The HSP sphere is indicated by the grey sphere and the calculated values are shown by Ra, R0 and RED. b) Solubility sphere for polylactic acid (PLA) showing its 3-D Hansen space given in (transparent green). Highlighting which solvent will be best for dissolving PLA. c) an alternative angle of b) [42].*

In Table 1 below, there lies the data set that was run with respect to PLA to generate Figure 1b. The dataset was taken from HSPiP, a package of eBook, software, and datasets for HSP. As shown in the dataset (Table 1), a plethora of molecules are presented and the three variables that make up the Hansen solubility parameters are given for each specific compound. This dataset demonstrates the efficacy that certain solvents will have when being used to dissolve PLA. As seen at the top of the table, Pyridine has a RED value of 0.176 making it a most favourable solvent for PLA and the least favourable being water having a RED value of 3.96.

*Table 1. The data used in HSPiP to plot the Hansen sphere shown in figure 1 b) and c), where a “Score” 0 means insoluble, and 1 for soluble between PLA and the respective solvent.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Solvent** | **δD** | **δP** | **δH** | **Score** | **RED** | **MVol** |
| **598** | **Pyridine** | **19.00** | **8.80** | **5.90** | **1** | **0.176** | **80.9** |
| **521** | **N-Methyl-2-Pyrrolidone** | **18.00** | **12.30** | **7.20** | **1** | **0.516** | **96.6** |
| **524** | **Methylene Dichloride** | **17.00** | **7.30** | **7.10** | **1** | **0.374** | **64.4** |
| **307** | **1,3-Dioxolane** | **18.10** | **6.60** | **9.30** | **1** | **0.306** | **69.9** |
| **481** | **Methyl Ethyl Ketone** | **16.00** | **9.00** | **5.10** | **1** | **0.635** | **90.2** |
| **617** | **Tetrahydrofuran** | **16.80** | **5.70** | **8.00** | **1** | **0.480** | **81.9** |
| **156** | **Chloroform** | **17.80** | **3.10** | **5.70** | **1** | **0.554** | **80.5** |
| **7** | **Acetone** | **15.50** | **10.40** | **7.00** | **1** | **0.751** | **73.8** |
| **297** | **Dimethyl Formamide** | **17.40** | **13.70** | **11.30** | **1** | **0.841** | **77.4** |
| **303** | **Dimethyl Sulfoxide** | **18.40** | **16.40** | **10.20** | **1** | **0.996** | **71.3** |
| **306** | **1.4-Dioxane** | **17.50** | **1.80** | **9.00** | **1** | **0.722** | **85.7** |
| **328** | **Ethyl Acetate** | **15.80** | **5.30** | **7.20** | **1** | **0.681** | **98.6** |
| **403** | **Furan** | **17.00** | **1.80** | **5.30** | **1** | **0.760** | **73.1** |
| **169** | **m-Cresol** | **18.50** | **6.50** | **13.70** | **1** | **0.734** | **105** |
| **637** | **Toluene** | **18.00** | **1.40** | **2.00** | **1** | **0.881** | **106.6** |
| **697** | **Xylene** | **17.80** | **1.00** | **3.10** | **1** | **0.859** | **121.1** |
| **52** | **Benzene** | **18.40** | **0.00** | **2.00** | **1** | **0.992** | **89.5** |
| **414** | **Hexafluoro Isopropanol** | **17.20** | **4.50** | **14.70** | **1** | **0.956** | **105.3** |
| **754** | **Isoamyl Alcohol (3-Methyl-1-Butanol)** | **15.80** | **5.20** | **13.30** | **1** | **0.963** | **109.3** |
| **10** | **Acetonitrile** | **15.30** | **18.00** | **6.10** | **0** | **1.331** | **52.9** |
| **255** | **Diethyl Ether** | **14.50** | **2.90** | **4.60** | **0** | **1.078** | **104.7** |
| **181** | **Cyclohexane** | **16.80** | **0.00** | **0.20** | **0** | **1.182** | **108.9** |
| **443** | **Isopropyl Ether** | **15.10** | **3.20** | **3.20** | **0** | **1.006** | **141.8** |
| **417** | **Hexane** | **14.90** | **0.00** | **0.00** | **0** | **1.390** | **131.4** |
| **325** | **Ethanol** | **15.80** | **8.80** | **19.40** | **0** | **1.479** | **58.6** |
| **456** | **Methanol** | **14.70** | **12.30** | **22.30** | **0** | **1.923** | **40.6** |
| **696** | **Water** | **15.50** | **16.00** | **42.30** | **0** | **3.960** | **18** |

**1.2 Overview of computational approaches to predict solubility**

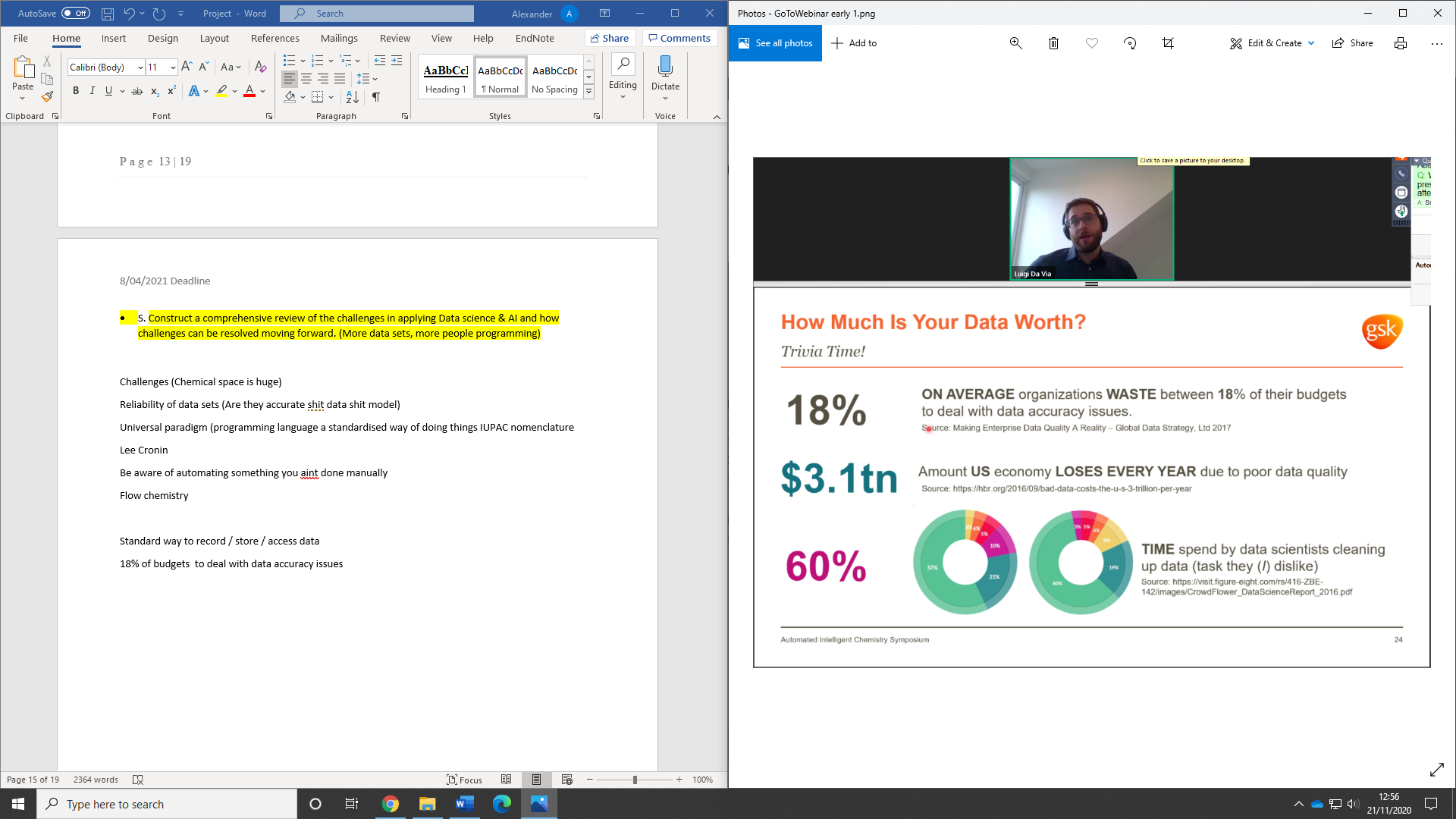
The MARSplines approach and QSARINS computational framework are considered during this brief overview. This is because they both complement each other well and are improvements of previous methods like QSARS. The MARSplines was chosen as it provides open-source molecular descriptors, which is important for improving software, and it can also be used in cluster computing, meaning that it can be used in industry to carry out larger scale analysis. In the QSARINS approach the focus is on the smaller scale and it uses automation to check for duplication of data.

For the fast and accurate prediction of solubilities, there are an array of computational methods used to complement experimentally determined HSP or even surpass them. Quantitative Structure-Property Relationship (QSPR) and similar methods have been used to successfully forecast solubility in “biorelevant dissolution media.” [28] Different methodologies for the successful predictions of HSP have been brought about by the combination of multivariable adaptive regression splines (MARSpline). [29] This involves the use of multivariable regression with the use of one and two-dimensional PaDEL molecular descriptors, where the PaDEL descriptor is open-source software that is used to calculate descriptors and fingerprints. Moreover, PaDEL has an advantage over separate molecular descriptor software, as it offers an open-source package, which is complemented by an integrated graphical user interface (GUI). This allows the user to have a programmable command-line to carry out large scale analysis, and operations that can performed on a software job scheduler. This has the purpose of running a cluster of computers simultaneously, for an industrial scale analysis. [30] However, like most of the Cheminformatic software, multiple core CPU computers must be used for timely analysis. [30] This can be costly for small and start-up companies but is required to make the analysis cost-effective and worth the user’s time. The efficacy of the MARSplines approach using QSPR and QSAR were validated using QSARINS computational framework, where the solubility classification for solutes and solvents were derived from SMILES strings. [29]

The QSAR models have the potential to be very useful for the prioritisation of chemicals and screening. This is only true when the models are worked on and meticulously tested. [31] Moreover, they can be used with insufficient data and even before the synthesis, which can aid the user when considering the safe chemical design approach or with respect to a retrosynthesis. [31] The QSAR approach can be summarised into three stages. The first stage is the preparation stage using inputted data that must be made ready and set up for an analysis. Next is stage two, where calculations can be made on specific parameters and descriptors. Lastly, in stage three the model will be further refined and validated with a specific selection of data. The most important factor to be considered is the relevance and reliability of the data set. [32] Lastly, if the data is inaccurate this will jeopardise the model, which will lead to unreliable results.

The QSARINS approach combats one of the disadvantages of the QSAR model; this software is used for the analysis application and validation of QSAR multiple linear regression (MLR) and ordinary least-squares regression (OLS). It is more favourable than that of conventional QSAR modelling, due to its ability to import and minimise unnecessary data with an implemented automation algorithm, which gives a resultant output as a graph or table allowing for visual interpretation of data trends and correlations by the user. [31] Lastly, this algorithm involves the mitigation of redundant descriptors by the means of removing descriptors that are either the same, similar, or intercorrelated. [31] To remove unneeded data, the user has a programmable feature that entails a specificity search that finds and eliminates duplication data within a specific percentage of similarity, deemed reasonable by the user. For example, if the pair of molecular descriptors are found to have over X percentage of similarity, then the descriptor will be eradicated. [31] This overcomes the limitation of QSAR as with the implementation of this algorithm there will be a smaller chance of duplicate data, therefore forming more valid and reliable models in smaller amounts of time, as the user does not have to manually check for duplicate features.

It is worth noting the importance of having accurate data; organisations must find or have accurate and reliable data available in an organised way. This was highlighted by Dr Luigi Da Via from GSK at the SCI Automated intelligent Chemistry seminar given on 26-27th of October 2020. As seen in Figure 4 Data scientists spend a lot of their allocated time cleaning up and refining data manually. This of course this is not the most joyous task as 60% of Data scientists disliked their clean up role. A way to overcome this long, arduous task is by using automation with applied machine learning models to clean up tasks automatically without the constant intervention of scientists.



*Figure 4. The importance of accurate data Automated Intelligent Chemistry [45].*

**1.2.2 The machine learning framework used in this project to predict HSP**

Machine learning (ML) is a specific learning algorithm, which gives out values based on multiple programmable parameters. These inputted parameters, outline the task and are processed by the given learning model, thereby giving outputted values that represent the hypothesised solution to the programmed data (final hypothesis). [1] Moreover, it is of importance to note whether a machine learning method would be beneficial to the user, as opposed to the use of traditional statistics. [2]

Traditional statistics consists of either Bayesian or frequentist methodologies, both offering their advantages and disadvantages. It is worth noting that an amalgamation of these two can lead to optimal results if a way is found to minimise the limitations of each method. [3] The focus of frequentist statistics is on the parameter, which is assumed to be a fixed constant, and confidence intervals are read in terms of repeated sampling. [4] In contrast, in the Bayesian approach, there is a focus on subjective probability, considering a priori prediction that can be changed by the user on successive iterations. Credible intervals are read in terms of subjective uncertainty, which are the equivalent of confidence intervals in the frequentist approach. [5] The application of Bayes theorem updates our prior intuitions about the parameter, with the information that the user can gather from the sampling set. [6] Furthermore, Bayes algorithms are easy to use and, when compared with other methods of training data, they perform extremely fast. [7]

For the successful optimisation of machine learning algorithms, three criteria must be met: (1) There must be enough specific data on the problem that is to be answered, (2) some form of relationship must be present between the outputted data points, and (3) it is observed that there is not a simple way of finding a mathematical relationship between the given inputted and outputted data. [8] Otherwise, statistical, and mathematical methods would be vastly superior, as they are less demanding on the user’s time and there is a lot more support available online to traditional maths than computational problems. When it comes to the application of ML, there are two methodologies that are deemed appropriate to be considered for understanding the premise of ML. These two methods are known as supervised and unsupervised learning. In supervised learning the system is given inputs that are programmed by the user and these give specific outputs. Then these outputs are put into a specific category “labelled” for the user acting as a visual aid for the analysis of the given system. [9] Moreover, in unsupervised learning, the “label is not available” and therefore, the output is less refined and harder to interpret. [9]

To implement a viable Machine learning methodology into a system, the user must have a framework to implement and build their model. The programming language used for this research project is Python. In recent times Python is one of the most common programming languages in the field of data analysis. [19] This is because Python is open-source and offers a vast number of available libraries, which again are free. These libraries enhance Python’s ability in data loading, visualisation, statistics, natural language processing, and image processing. The application of these frameworks is of much use for computational learning approaches to data analytics, and or in any situation where a user can apply programming to automate or make a process faster and more robust. Moreover, one of the main advantages of using Python is the ability to interact directly with the code, using a terminal or another tool like the Jupyter Notebook. Jupyter Notebook is utilised as the Interface development environment (IDE), for the scope of this project. There is also a lot of support available for Python online as it has been around for over 30 years.

This vast toolbox delivers data scientists with a large array of general and special-purpose functionalities. Machine learning and data analysis are based on iterative processes, which both rely on a guided data driven analysis. [19] These processes need to have tools that allow quick iterations and easy interaction with the programme. As a universal language, Python allows for the creation of complex graphical user interfaces (GUIs) and web services, alongside integration into existing systems. [20] Moreover, its framework can provide the user the ability to translate from different programming languages into a variant that is deemed appropriate for the situation.

The computational framework used for the prediction of the Hansen Solubility Parameters in this project is outlined below.

**Jupyter Notebook**

The Jupyter Notebook gives the user a more robust approach to the conventional notebook, as all the inputted information can be stored and viewed as an online interactive notebook. This allows an entirely interactive environment where code can be programmed and debugged just within the local network’s browser of the user. Moreover, the user does not have to have a deep background in programming to get the notebook up and running. The installation is easy, making it more enticing for user to share and explore their raw data. Furthermore, not only is it being utilised by beginner scientists, but it is also a robust tool for exploratory data analysis and is widely used by professional data scientists. Moreover, the versatility of the Jupyter Notebook is commended as it has many different kernels to support various programming languages; however, only Python will be used for the scope of this project. The successful incorporation of code, text, and images occurs with ease in the notebook’s virtual environment. Lastly, with the outcry for the standardisation of data, it is imperative that users have a place to share ideas and improve their code. Websites like GitHub are pioneers in this and provide open access resources for all users. This allows fellow scientists to manipulate, interpret and analyse data. [19] But more importantly this website provides an online platform for reproducibility and for user to learn the industry standards and conventions.

**NumPy**

NumPy is considered the industry standard for using applied mathematics in computing. It has a lot of use in several production systems and is known for its array computing functionality. It is also used for multidimensional arrays and high-level mathematical functions. It has been used to express linear algebra, statistical operations, and the Fourier transforms. [19] NumPy also can add, remove, and sort specific elements, which can help in the analysis of data, by making them more legible for the user. For the scope of the project, NumPy is used as a Python library that works with arrays therefore allowing the successful withdrawal of information from the CSV file upon request.

**Matplotlib**

Matplotlib is a graphing software that provides functions for applying a combination of mathematics principles that are offered by various other frameworks. For example, it allows NumPy to create and publish formal and high quality visualisations. It is worth noting that NumPy does not have the ability to execute these visualizations of data by itself, as it needs a plotting library. Further examples of Matplotlib’s plotting abilities are the generation of line charts, histograms, parabolic functions, and scatter plots. The purpose of this library is to provide the user a way of visualising data and different aspects of the data, which enables the user a much more depth in their analysis. It offers a MATLAB-like (MATLAB is another numeric graphing and computing environment) interface but has no cost due to being publicly available and free. Moreover, its use inside the Jupyter Notebook framework, offers the user figures that can be directly shown with specific inline commands and colors that can be used to highlight trends in data. [19] More importantly promoting reproducibility of results as it allows people from all around the world to graph the results from the user’s original data.

With respect to the project, Matplotlib is an object orientated API used to embed the plots onto the Jupyter notebook and is used in the code to plot data showing the trends between solubility parameters and R2 values.

**Pandas**

Pandas is arguably an extension of NumPy as it relies heavily on NumPy’s use of objects. More specifically, Pandas is a Python library with the intention of data manipulation and analysis. It was built around and modelled after a predecessor framework R. Pandas is the programmers equivalent of excel expressing its data frame in a tabulated mode. It is robust and provides a great range of operations that the user can use to modify and work in the given tables, which is much like that of SQL, with the application of queries and joining of tables. Pandas is different from NumPy as Pandas allows all the columns to have a different type, for example, strings, integers, dates, and floats; whereas NumPy is rigid and only allows entries to be in the same given type. Pandas also offers the user great versatility as it enables the user to work with a countless variety of varying file formats and databases. These include Excel spread sheets, SQL formats and comma-separated values (CSV). [19]

Pandas was used as the framework required for reading the CSV files and for generating user friendly tables showing the training dataset.

**Scikit-learn**

Scikit-learn (also known as Sklearn) is an open-source, supervised ML library for Python, which allows for the constant reiteration and improvement of code. Moreover, being free to use and distribute, anyone can easily obtain the source code to see what causes the specific output, with the use of GitHub and Stack Overflow. The scikit-learn project is constantly evolving, and user are all actively developing and improving its functionality. Scikit-learn is one of the most popular tools for the application of data science, A.I. and machine learning. It has a huge application in a variety of industries due to its successful integration with numerous Python tools that aids data science. Moreover, it offers a wide range of tutorials and there are many lectures available on YouTube and or other online platforms, delivered by industry experts and academics. [19]

With respect to the framework in this project, it is used to create a linear fitting ML model.

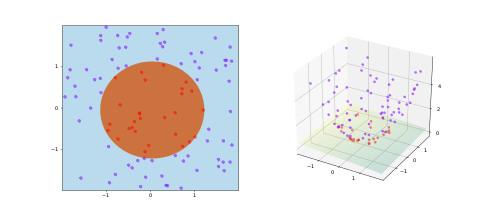
Below is a summary of the ML models embedded in SciKit-learn discussed in the project.

**Lasso**

Lasso improves both prediction accuracy and model interpretability by combining the good qualities of ridge regression and subset selection. If there is a high correlation in the group of predictors, LASSO chooses only one among them and shrinks the others to zero. It reduces the variability of the estimates by shrinking some of the coefficients exactly to zero producing easily interpretable models. [23]

**Kernel**

The Kernel method is a function in which non-separable variables of a specific dimension in “input space” are then placed into a higher dimension “feature space” so that the variables are now visually separable. [37] It is categorised as an algorithm as a part of pattern analysis and its use is to find trends and correlations in data for optimal analysis.



*Figure 5. A visualisation of the Kernel trick [46].*

**Ridge regression**

The least square fit method provides a direct and unbiased regression in which a statistical form of analysis can be used to check correlations and trends in the data. One of the downfalls in this method is that least-squares regression can give the user a high variance in which a minute change within the given data set can give intolerable results. [22] It is observed that a such a problematic result usually occurs when a small dataset is used. In most cases, the methodology is responsible for collinearity that leads to an over-fitted model. This hindrance is not resolved by the standard least square fit models; therefore, the user must consider a different methodology. It should be noted that the Ridge regression methodology has the goal of reducing the given sensitivity of the data set. It does this by implementing a bias to the data set. As observed in the figure below the regression does this by adding a value to the total amount of squared residuals, intending to reduce the overall variance. [22] This functionality is called a “penalty function” and is considered to be helpful when using training sets that are not large enough. This type of regression can be used to offer a result with the incorporation of cross-validation and the regression penalties. [22]

Equation 2 illustrating the difference between the regression calculations [21]

**RD-Kit**

RD-Kit houses a software suite for cheminformatics, computational chemistry, and predictive modelling. Its main use is for general molecular functionality and is based on SMILES. Moreover, constant improvements are being made to the software and it can be ran alongside other 3D visualisation software’s, such as PyMOL. [21]

**SMILES (Simplified Molecular Input Line Entry System)**

A SMILES string was created by David Weininger during the 1980s. It represents a molecule by describing its molecular graph based on the premises of graph theory. This is done by showing the atoms and bonds in a connection table, however no chiral or isotopic information is provided. There is an abundance of valid SMILES strings; however often they can misrepresent a structure of a given molecule. An example of this ideology is, CCO, OCC and C(O)C which all give the intrinsic structure of ethanol. [39]

Moreover, in the SMILES language, there are two essential types of symbols which are used in applied graph theory. These are the atoms and bonds. Using these SMILES symbols, is a way to express to the user the molecular build of a molecule in space forming a molecular graph which is made of the "nodes" and "edges" and assigning "labels" to the components of the graph. Therefore, it is given that each variant of an atom is a node, and the edge is the type of bond that occurs in the given instance. [39]

**Structural key/Molecular fingerprint**

In structural keys, specific parts of molecules are encoded into a binary which is a fixed-length bit string. Every variant of functional groups is associated with a specific value and therefore creates a specific molecular pattern. When a structural key is generated for a given molecule, the binary that is from the bit string shows the user whether these specific molecular patterns are present or absent in the molecule using the Boolean logic of true (1 object being present) or false (0 object being absent). When revising a structural key, the user must pay attention to the specific fragments that are used as the efficacy of such keys is completely reliant on the chosen fragments. This is then ran in a molecular database where a statistical analysis will give an insight to the probability and appearance of the given searched molecule. [27]

# 1.3 Aims of the project

The aim of the project is to construct a Lasso linear regression ML-model. A data set of 193 solvent molecules with four different descriptors taken from a previous publication will be used to input into the model to predict HSP. The accuracy of the predicted HSP will be evaluated by statistical errors and the effectiveness of the Lasso model will be evaluated by comparing with other published ML-models.

# Chapter 2. Methodology

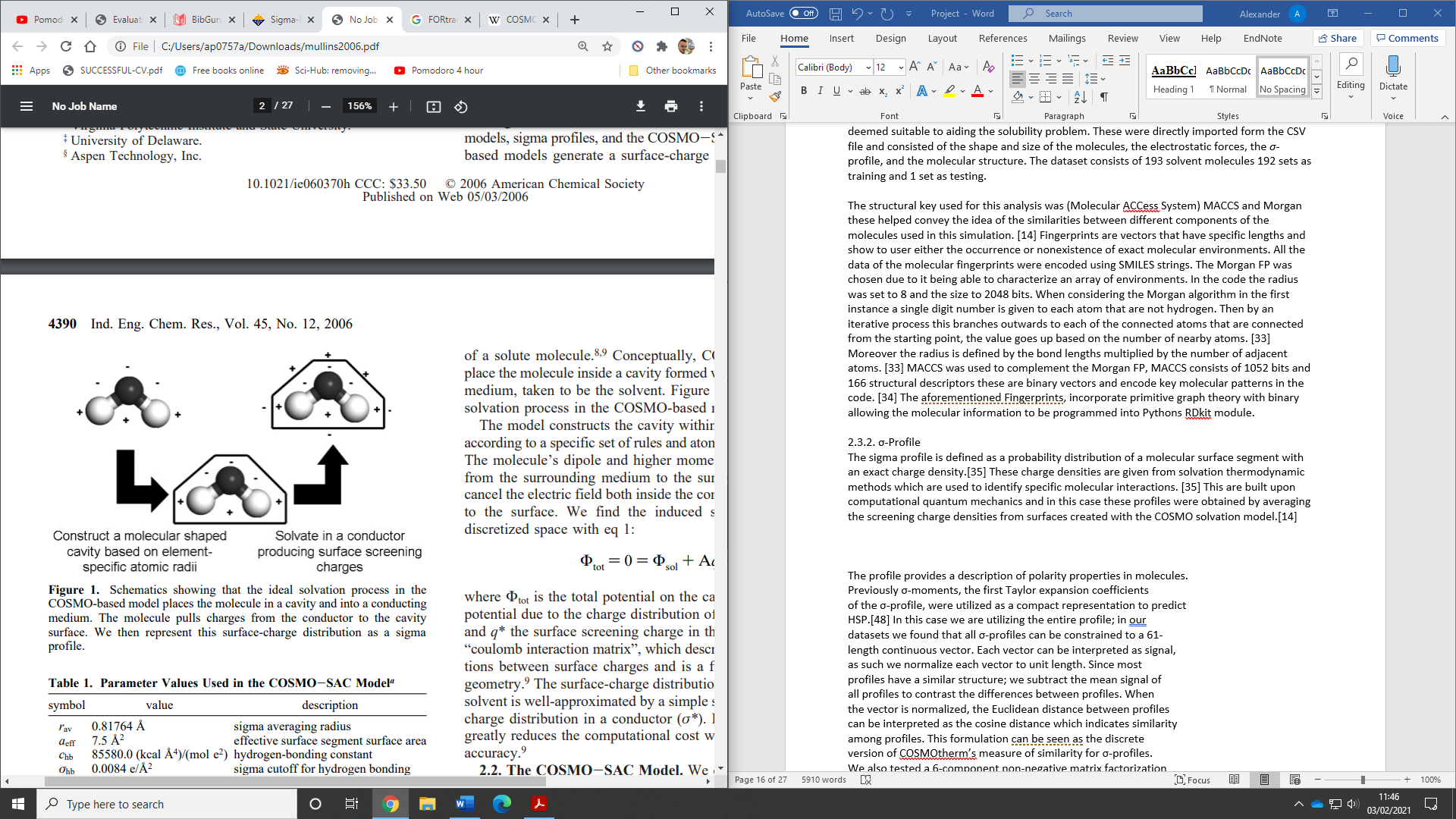
A dataset of 193 solvent molecules was taken from a previously published paper [14]. The molecules were made up of solvents and they contained 2 to 85 heavy atoms. The dataset was in the CSV file format and consisted of SMILES strings of the solvent molecules, the shape and size of the molecules, the electrostatic forces, the *σ*-profile, and the molecular structure, volume, and dipole moments of the molecules, which were completed using quantum mechanical calculations. Details of how the data was obtained in the work of Sanches-Lengeling et al [14] are given below. The data can be found on GitHub see the reference. [47]

**2.1 Molecular fingerprints**

The structural key used for this analysis was (Molecular ACCess System) MACCS and Morgan fingerprints, which help to identify similarities between different components of the molecules used. [14] Fingerprints are vectors that have specific lengths and show to the user either the occurrence or nonexistence of exact molecular environments. All the data of the molecular fingerprints were encoded from their SMILES strings using the RDKit program. The Morgan FP was chosen due to it being able to display a variety of different chemical environments and to simulate the Sanchez et al paper. In the code, the radius was set to 8 and the size to 1024 bits. When considering the Morgan algorithm, in the first instance a single-digit number is given to each of the atoms that are not hydrogen. Then by an iterative process this branches outwards to each of the connected atoms that are connected from the starting point; the value goes up based on the number of nearby atoms. [33] Moreover, the radius is defined by the bond lengths multiplied by the number of adjacent atoms. [33] MACCS was used to complement the Morgan FP. MACCS consists of 1052 bits and 166 structural descriptors; these are binary vectors and encode the key molecular patterns in the code. [34] The aforementioned fingerprints, incorporate primitive graph theory with binary allowing the molecular information to be interpreted by Python’s RDKit module.

**2.2 σ-Profile**

The sigma profile is defined as a probability distribution of a molecular surface segment with an exact charge density. [35] These charge densities are given from solvation thermodynamic methods, which are used to identify specific molecular interactions. [35] Moreover, these are built upon by computational quantum mechanics and in this case, these profiles were obtained by averaging the screening charge densities from surfaces created with the COSMO solvation model. [14] The premise of a COSMO model is to put a specific molecule into a vacant space that is generated in a homogenous solvent. As seen in Figure 6, a cavity is generated in molecular space showing how the solvent will interact with other species within a given proximity.



*Figure 6. Showing the ideal case for the application of the solvation in the COSMO-based model [44].*

**2.3 Electrostatic**

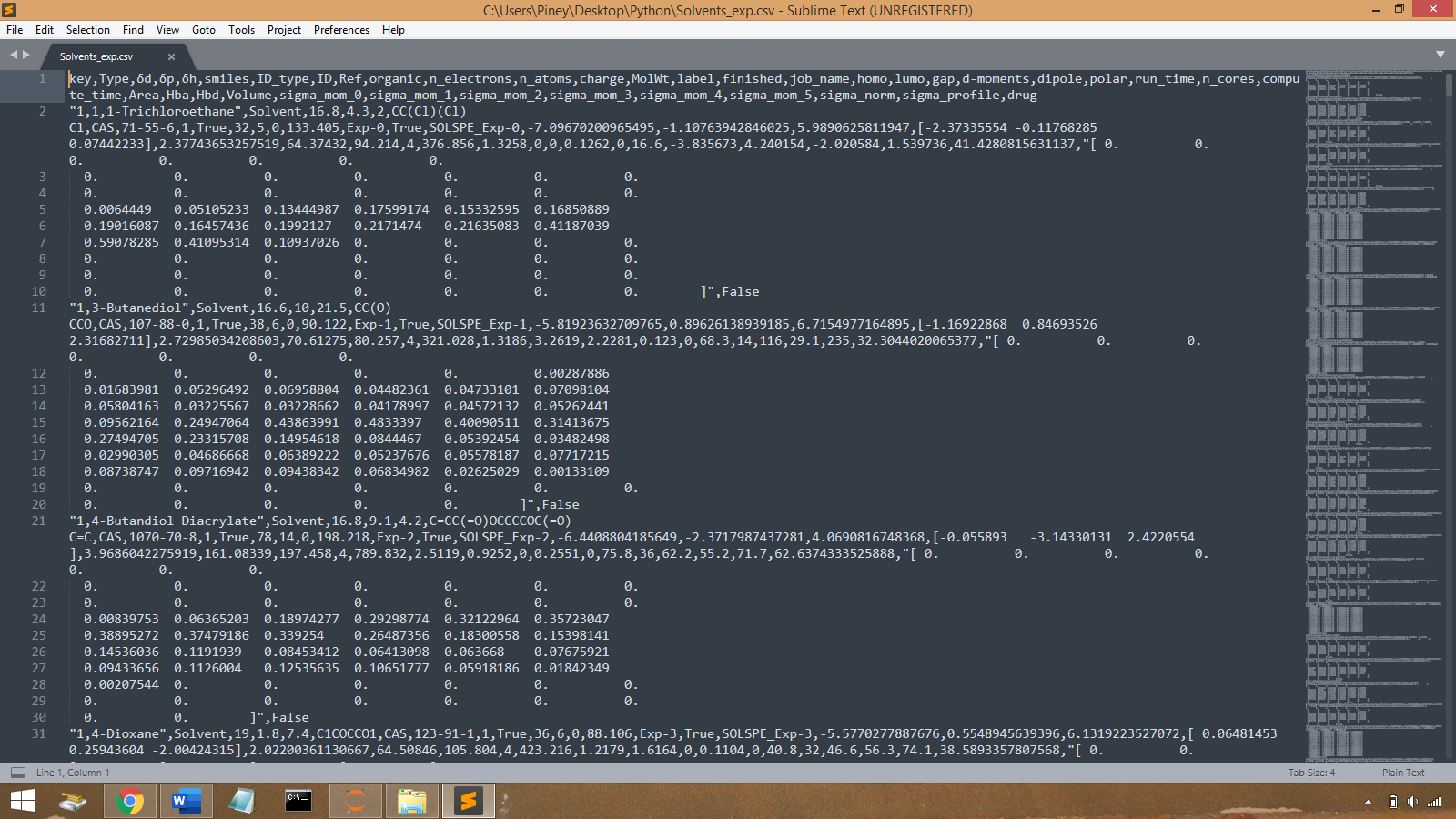
To get a deeper insight into the intrinsic interactions between specific solvent-solute systems, further electrostatic descriptors must be considered. [14] The intermolecular forces that were considered are polarisability and the overall direction of dipole moments, which were all derived from ORCA using a single point calculation. [14] Another method that was used to explore hydrogen bonding was the Conductor like screening model (COSMOtherm), with a specific emphasis on the electron-rich and electron-deficient centers, where the hydrogen bonding takes place. [14] The COSMO model is deemed to have improved efficacy over conventional models as the orientation in space of the system is considered for the molecules allowing for more precise calculations for chemical potentials in pure or mixed solvent environments. [36]

**2.4 Shape and Size**

When considering the size and shapes of the compounds, it is worth noting that there is a large assortment of different molecules which make up the solvent set. HSPiP consists of over 10,000 molecules, therefore, it is important to stay within a range of atoms as physicochemical properties will dramatically change in different systems. [14] The COSMO solvation model was used with the Volume and Surface Area calculations to obtain orientations in space. [14]

**2.5 The dataset format**

This outlines the format of the dataset used. The data is presented in the sequence as followed with the headings: Name of solvent molecules, Type (solvent), δd, δp, δh, smiles, ID\_type, ID, Ref, organic, n\_electrons, n\_atoms, charge, Molwt, label, finished, job\_name, homo, lumo, gap, d-moments, dipole, polar, run\_time, n\_cores, compute\_time, Area, Hba, Hbd, Volume, sigma\_mom\_0, sigma\_mom\_1, sigma\_mom\_2, sigma\_mom\_3, sigma\_mom\_4, sigma\_norm\_5, sigma\_profile, drug.

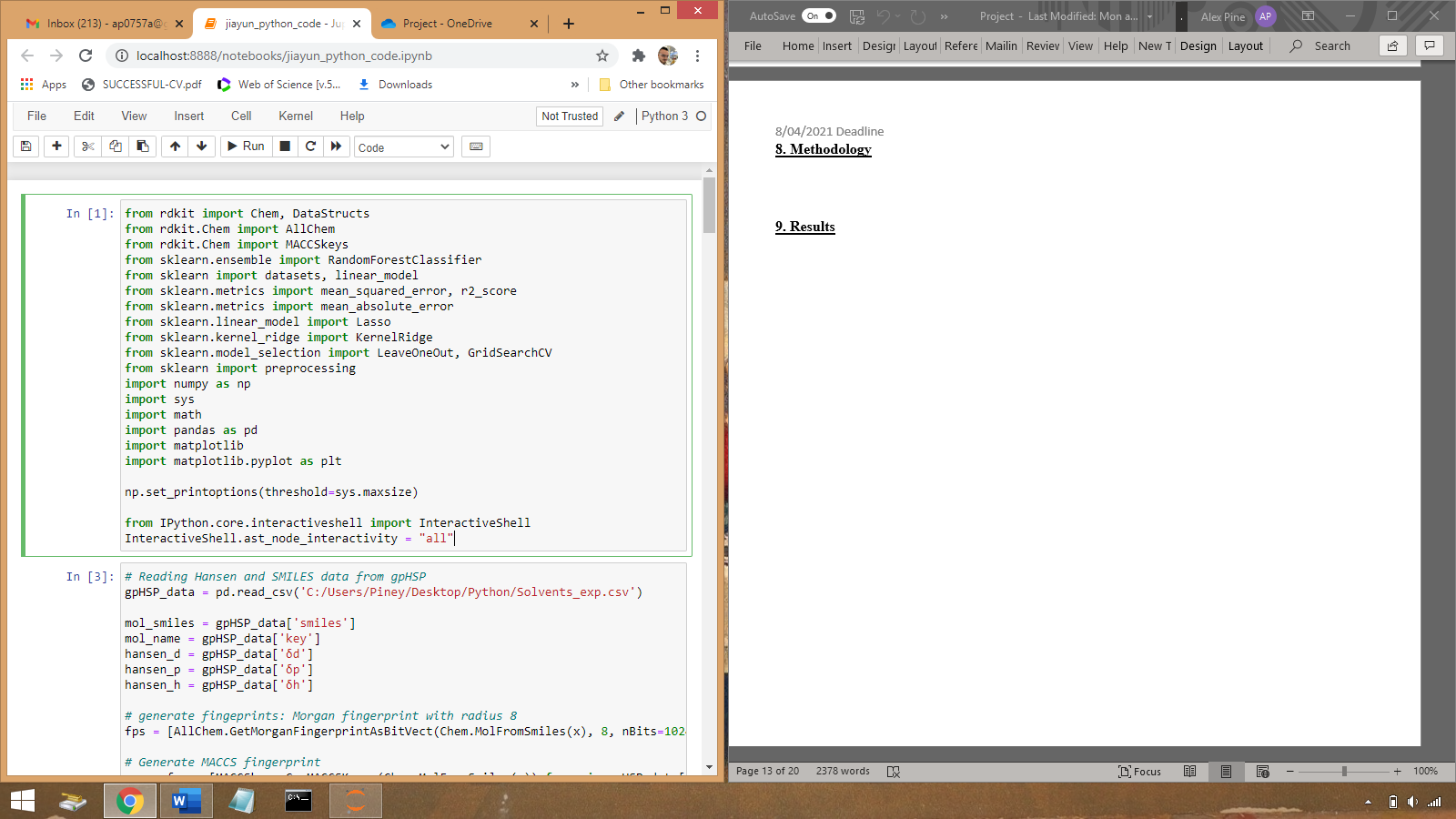


*Figure 7. Displaying the data format given by the CSV file.*

# 2.6 The Python code

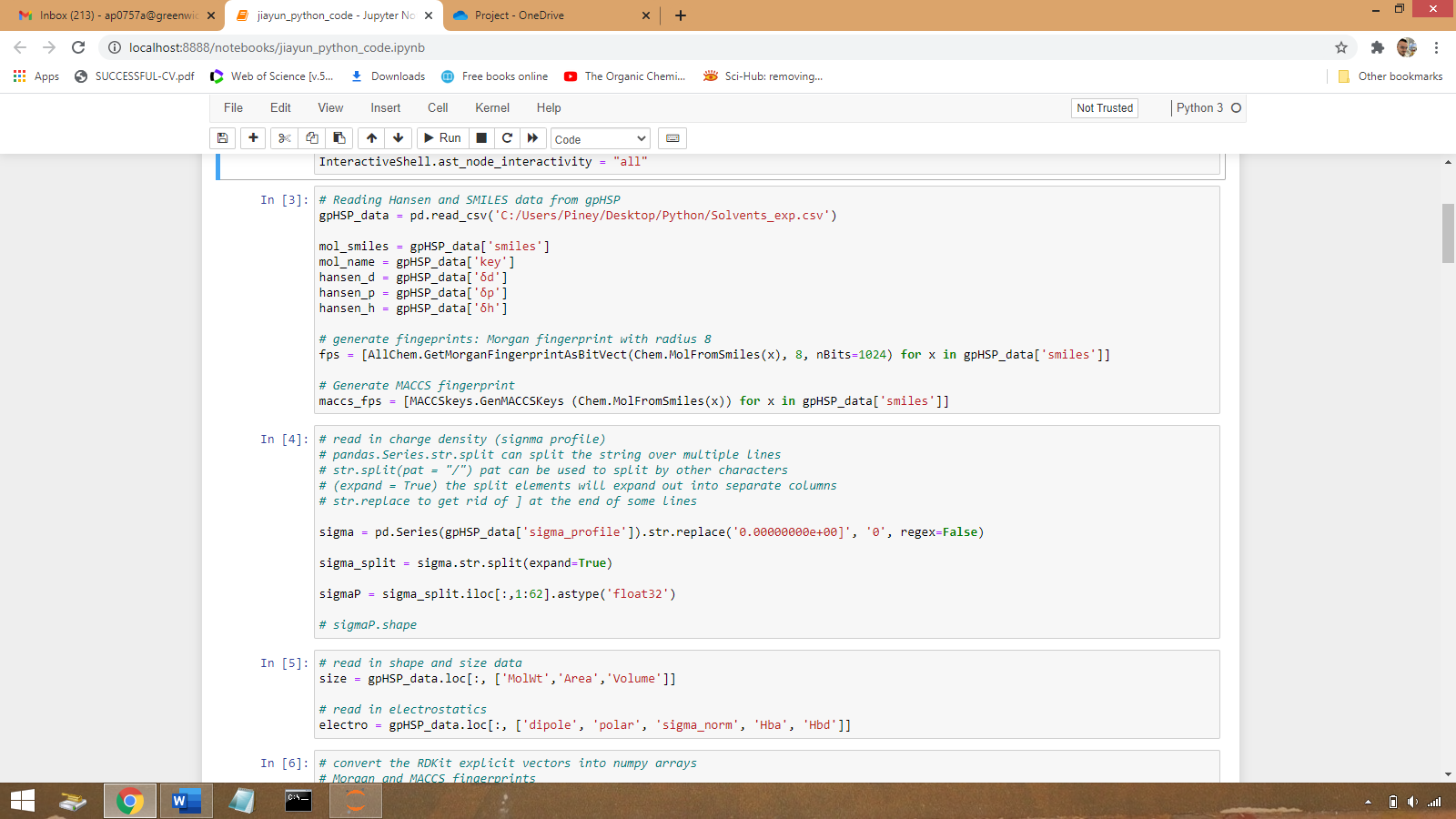
**2.6.1. Import libraries**

The required libraries are imported into the Jupyter notebook: RDKit, Sklearn, numpy, pandas and matpltlib.

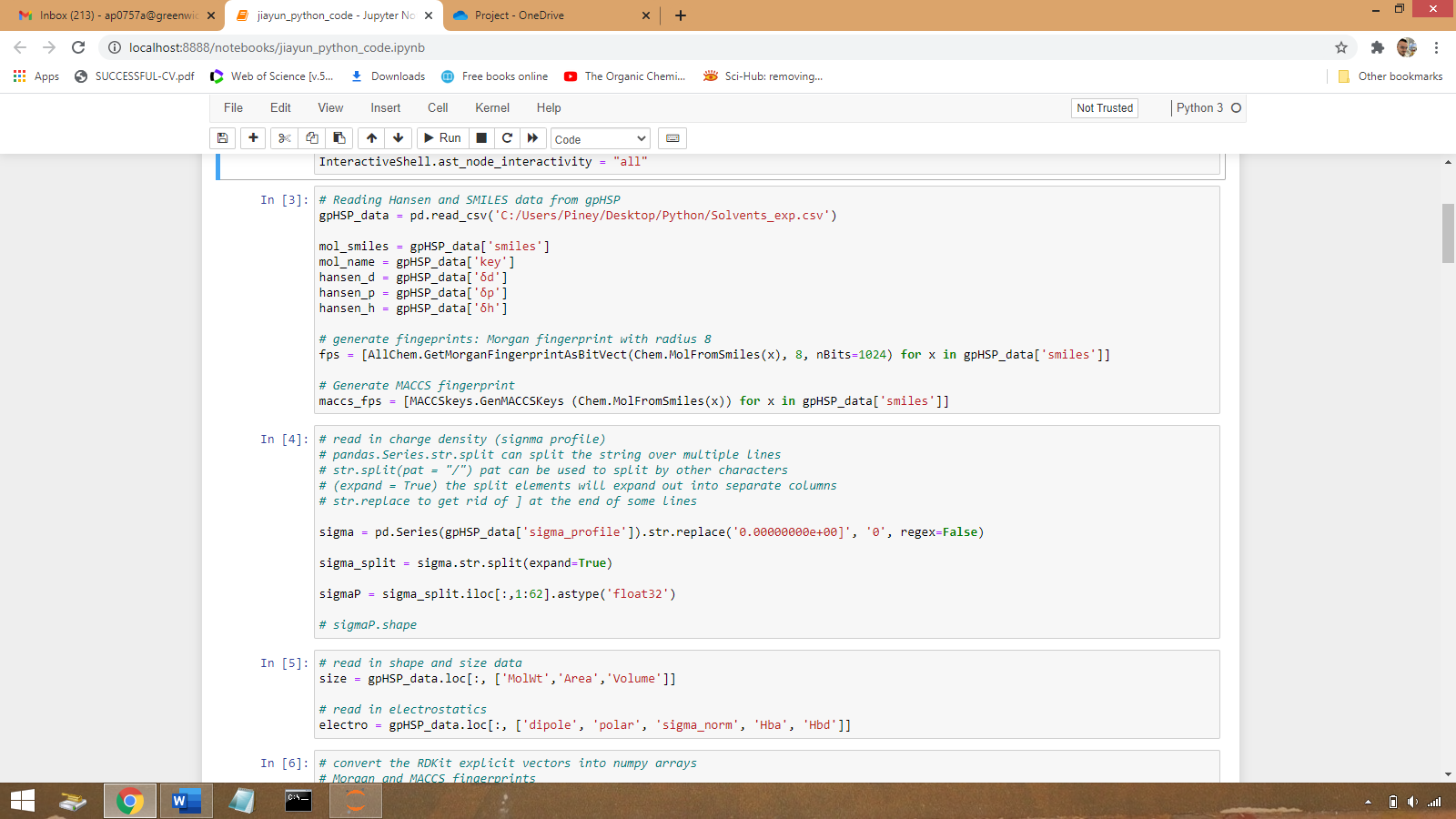


**2.6.2. Reading in and generating descriptors**

The variable gpHSP\_data reads the CSV file (Solvents\_exp.csv) which gives all the information about the Hansen Solubility Parameters and SMILES strings of the solvent molecules. The parameters used for the analysis are given their own variable names mol\_smiles, mol\_name, hansen\_d, hansen\_p and Hansen\_h. The Fps variable generates the fingerprints using Morgan (radius 8 and the size 1024) and MACCS fingerprints.



The sigma variable reads the charge density mapping, sigma profile and the series function as a one-dimensional array holding the gpHSP\_data as a Python object.



The variable size reads the shape and size data, and the electro variable reads the electrostatic information.

Text

Description automatically generated

Np\_fps converts the previously generated MORGAN and MACCS fingerprints from explicit vectors into numpy arrays.

Graphical user interface, text, email

Description automatically generated

All of the data arrays are concatenated, standardised and scaled accordingly.

Graphical user interface, text, email

Description automatically generated

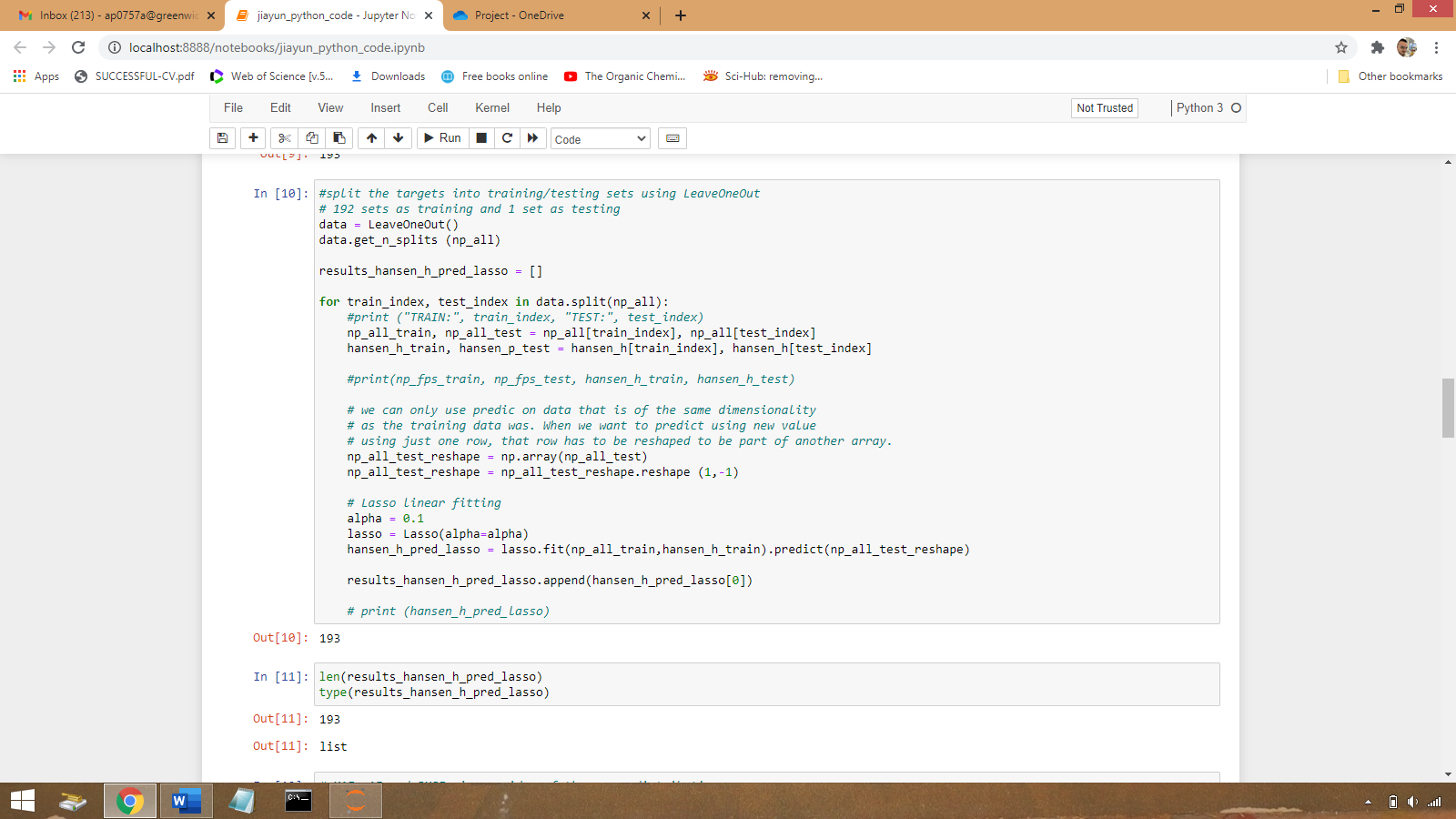
**2.6.3. Fitting data into Lasso linear regression using leave-one out cross validation**

The leave one out cross validation approach provides train/test indices to split data into train/test sets. Each sample is used once as a test set (singleton) while the remaining samples form the training set. [25] This is then continued giving 193 testing sets. The code then reshapes the data and prediction can only occur on code with the same dimensionality as the training set. This new prediction will use just a single row and is shaped to make another array.

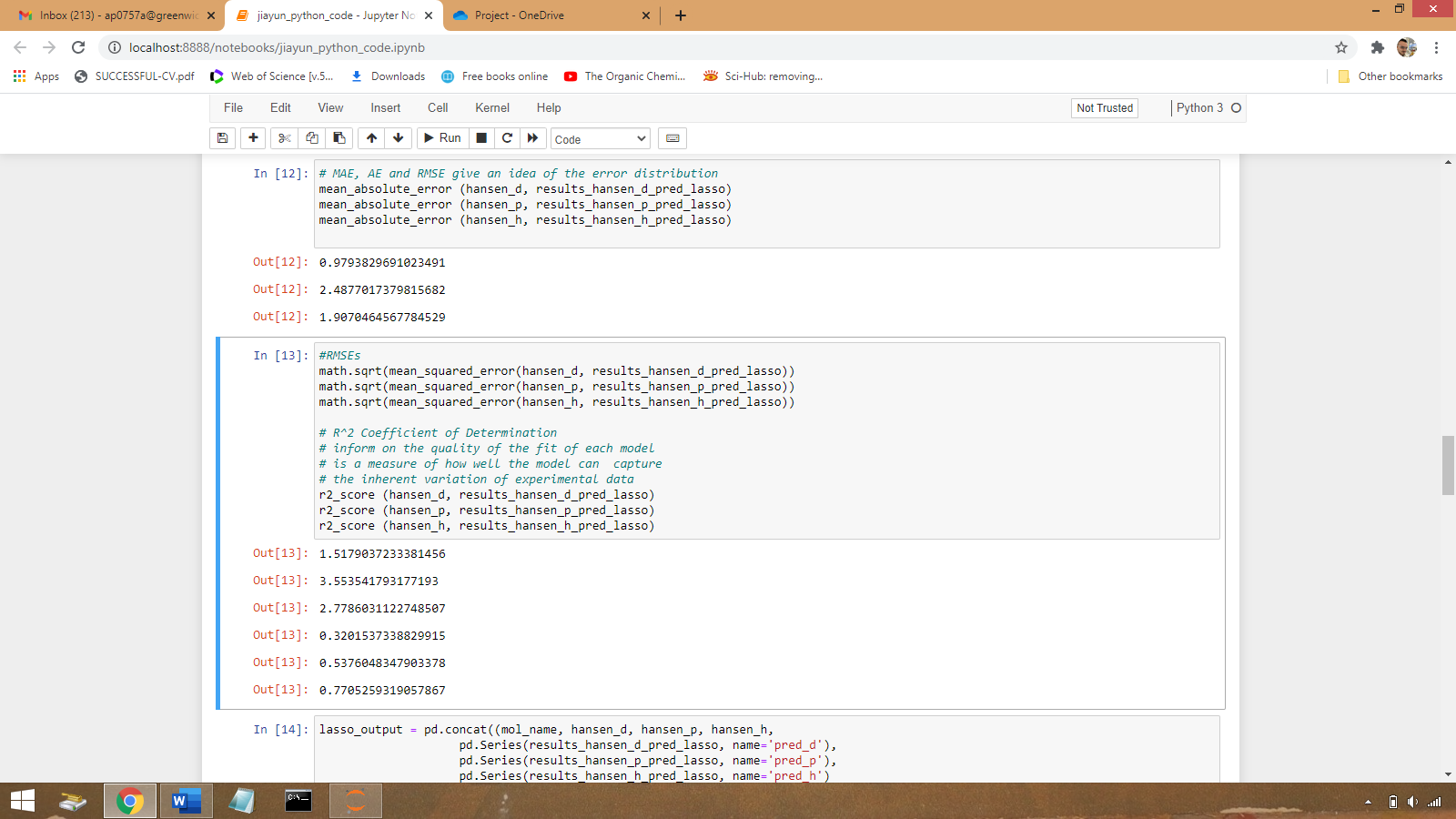
Graphical user interface, text, application, email

Description automatically generated

The len function will return the number of items in an object in this case 193, and the type of function which returns the class type.



The mean absolute errors (MAE) are calculated for the predicted HSP.

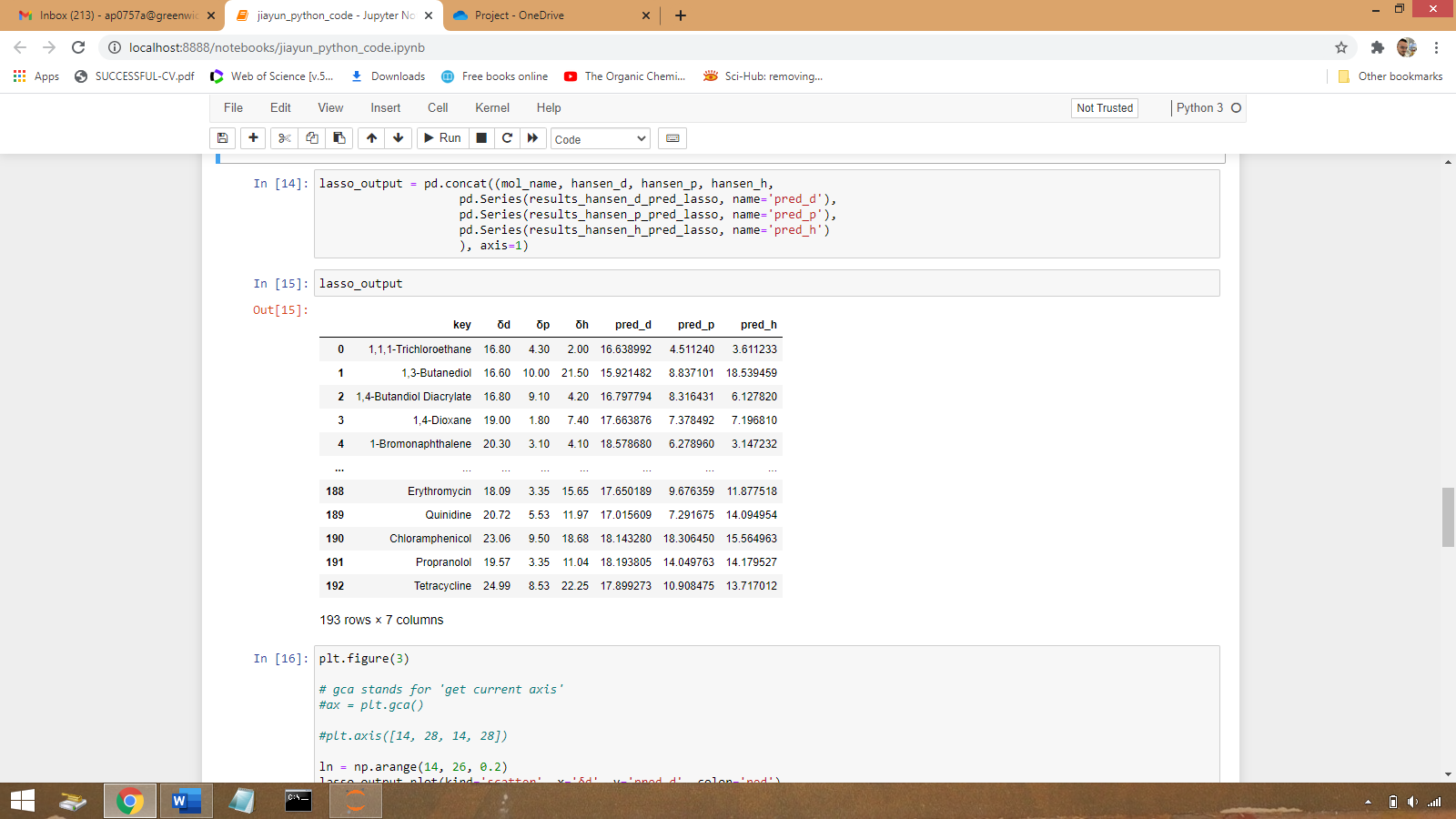


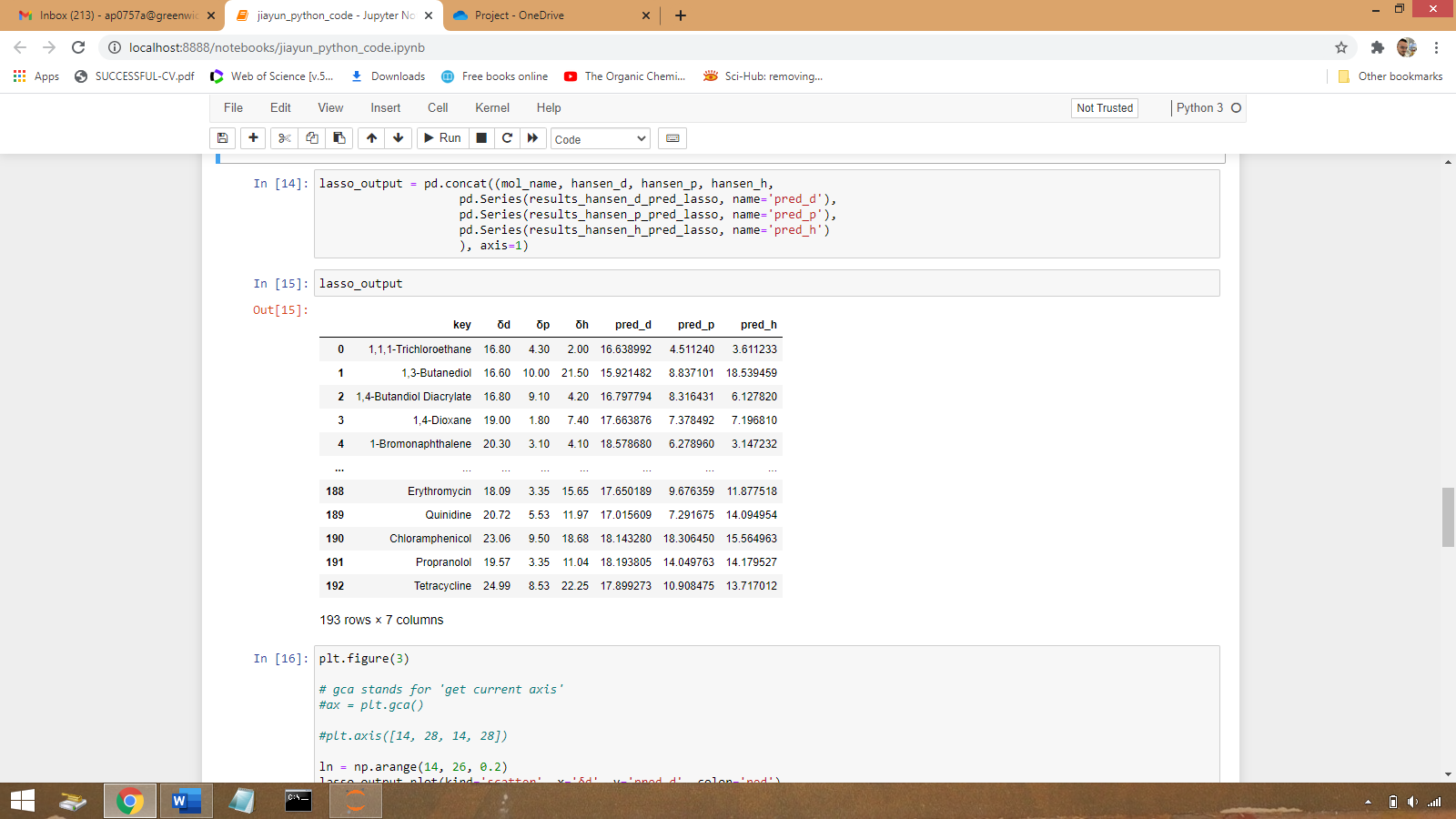
The root mean squared error (RMSEs) and the R2 coefficient.

Text

Description automatically generated

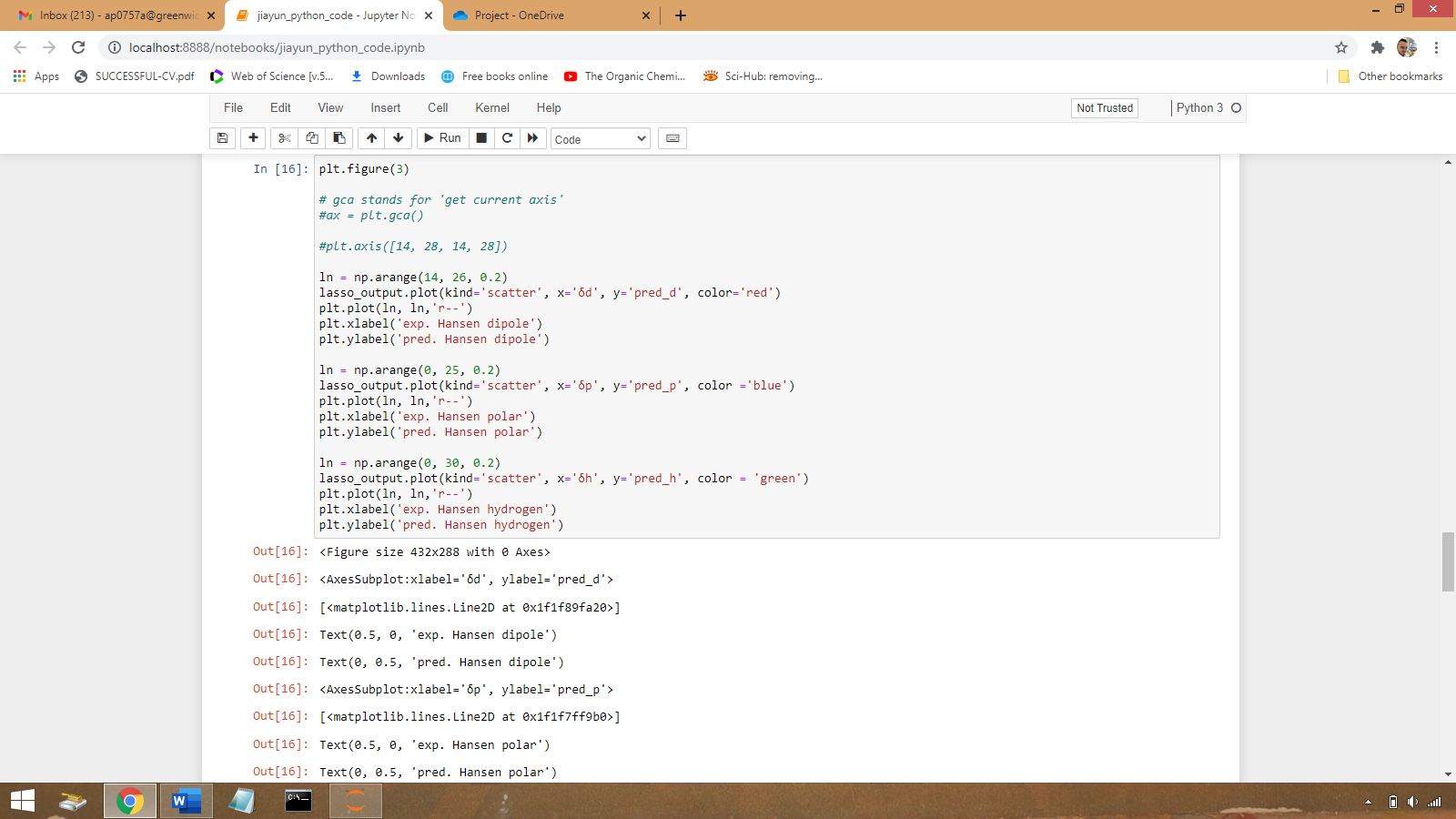
The predicted HSP along with the original HSP are concatenated into one table using Pandas and displayed in Jupyter notebook.

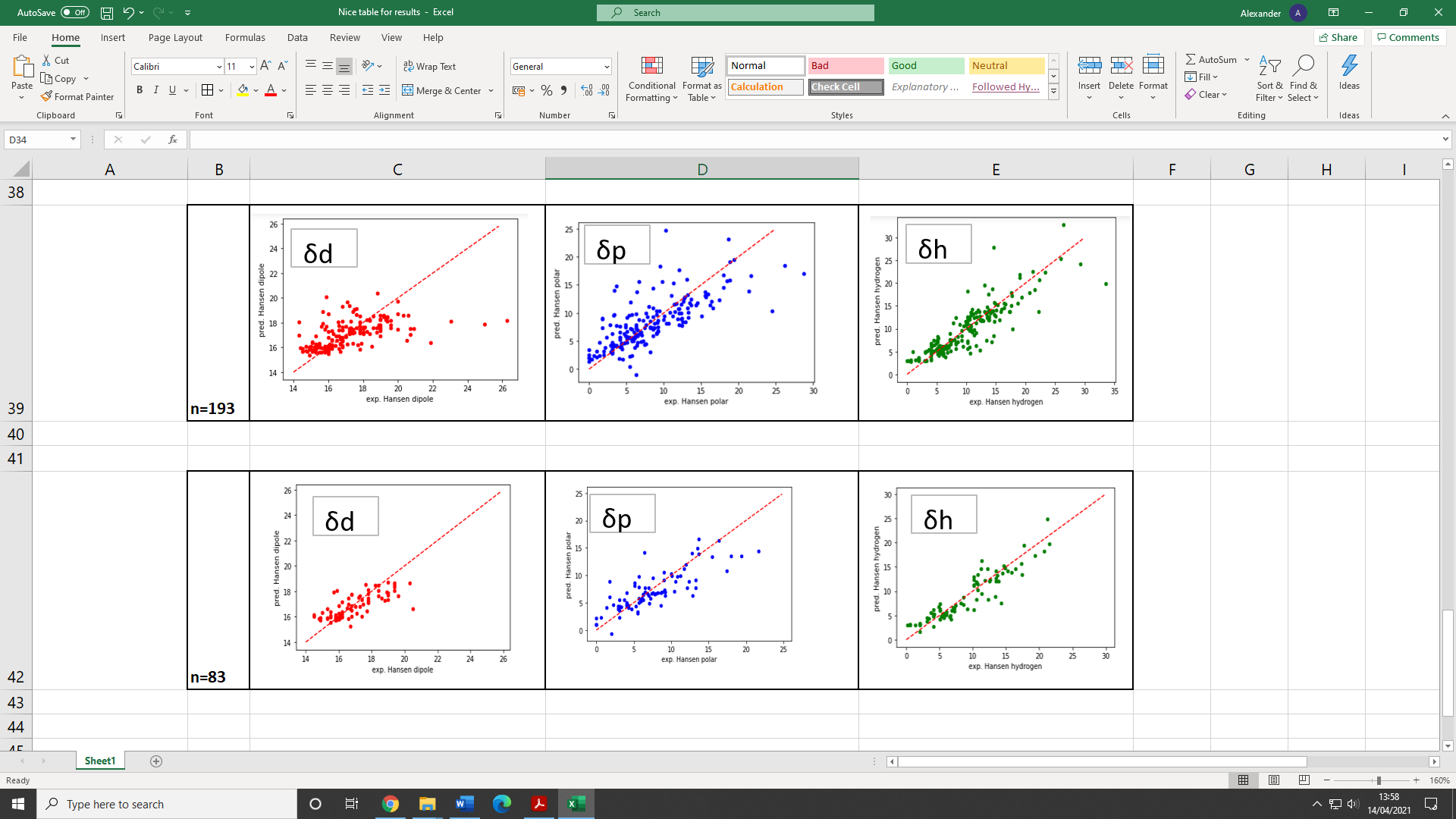




**2.6.4 Visualisation of predicted HSP**

Matplotlib is used to visualise the predicted HSP values in comparison to the original experimental data. Further details are given in Chapter 3, results, and discussion.





**2.7 Data variation and additional model attempted**

In addition to the Lasso linear regression, a Kernel ridge regression model has also been attempted during the project. However, this method was not successful due to the difficulty of deciding the fitting parameters and could not be completed within the timeframe of the project.

Additionally, a second Lasso linear regression fitting was made using half of the number of molecules (87 molecules) and will be discussed in Chapter 3, results, and discussions.

# Chapter 3. Results and Discussion

**3.1 Interpretation of accuracy of the predicted HSP**

Two sets of data were used in the Lasso linear fitting model, with the first set containing 87 solvent molecules and the second, larger data set containing 193 molecules. The first set of 87 molecules is taken randomly from the full set of 193 molecules. The predicted Hansen parameters were plotted (Y-axis) against the expected Hansen parameter (X-axis) (Figure X). When considering both data sets, it can be seen that for many molecules, predicted values are in fair agreement with experimentally determined HSP, although in some cases, the deviations in the two values can be quite large. It is also interesting that the predicted δh seems to perform better in comparison to δd and δp.

To evaluate the accuracy of the model, mean absolute error (MAE), room-mean square error (RMSE) and the coefficient of determination (R2) were calculated (Table 2). MAE and RMSE indicate the error distribution while the R2 value identifies how close the data is fitted relative to a given fitted regression line. When considering the 193-molecule data set, it is observed that the R2 values are 0.32, 0.54 and 0.77 for δd, δp and δh, respectively, are suggestive of an increasingly stronger correlation. It is also observed that for the n=193 molecules system, there is a much larger scattering of data points than that occurred for the n=87 molecule system (R2 of 0.56, 0.67 and 0.86 for δd, δp and δh, respectively). The increased data points make it hard to identify a strong correlation. For δh with both 87 and 193 molecules, it is worth noting that the predicted δh performs better in comparison to δd and δp, consistent with the visualization of the data in Figure X. It is also observed that the MAE and RMSE is greater for the δp n=193 graph. This implies that the average magnitude of error is greater than in the other graphs given below. Moreover, when the data set is reduced to 87 molecules there is some observable changes in the error values and R2 values. All the R2 values improved meaning their values increased and the mean squared values were also reduced. Moreover, the applied lasso regression was chosen to inform the quality of the fit of each model and its efficacy in measuring how well the model can capture the variation in the experimental data.

(A)

Graphical user interface, chart

Description automatically generated

(B)

Graphical user interface, chart

Description automatically generated

*Figure 8. Predicted HSPs in scatter plots against experimental HSP for (A) the 87 molecules and (B) the 193 molecules. The dashed lines indicate the ideal fit.*

*Table 2 The measures of regression and error taken from a paper [14] from Sanchez et al in red and in black from separate experimentation.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | Target | Approach | MAE | RMSE | R2 |
| solvents n=193 from GitHub [1] | δd | KernelRidge | 0.80 | 1.22 | 0.56 |
|  | Lasso | 0.91 | 1.44 | 0.39 |
| solvents n=193 |  | Lasso | 0.68 | 0.94 | 0.32 |
| solvents n=87 |  | Lasso | 0.98 | 1.52 | 0.56 |
|  |  |  |  |  |  |
| solvents n=193 from literature [14] | δp | KernelRidge | 2.46 | 3.70 | 0.50 |
|  | Lasso | 2.80 | 4.34 | 0.31 |
| solvents n=193 |  | Lasso | 1.93 | 2.66 | 0.54 |
| solvents n=87 |  | Lasso | 2.49 | 3.55 | 0.67 |
|  |  |  |  |  |  |
| solvents n=193 from literature [14] | δh | KernelRidge | 2.25 | 3.16 | 0.70 |
|  | Lasso | 2.66 | 3.56 | 0.62 |
| solvents n=193 |  | Lasso | 1.55 | 1.96 | 0.77 |
| solvents n=87 |  | Lasso | 1.91 | 2.78 | 0.86 |

In their paper [14], Sanchez et al used various regression approaches, namely gpHSP, KernelRidge, Lasso and Regularised Greedy Forest (RGF), where the same CSV files were used for the generation of models. It is observed that the R2 values obtained by their work and by the current project are similar, except that there is larger difference in the R2 values for δp by 0.23. Furthermore, when considering the Lasso fitting that was done in the Sanchez et al paper, there was an increase in all the measures of errors for the three parameters. From their code given on GitHub it is not clear how they did their Lasso fitting, also they used the same CSV file, but it is unclear how they implemented the data into the model was it from molecule 1-193 or maybe backwards from 193-1. It is not clear why all of Sanchez et al errors increased. Also, Sanchez et al carried out a KernalRidge regression which, lead to observable improvements in all measures of error and when considering the R2 values. However, these errors are still significantly higher than that obtained from the experimentation done for this project.

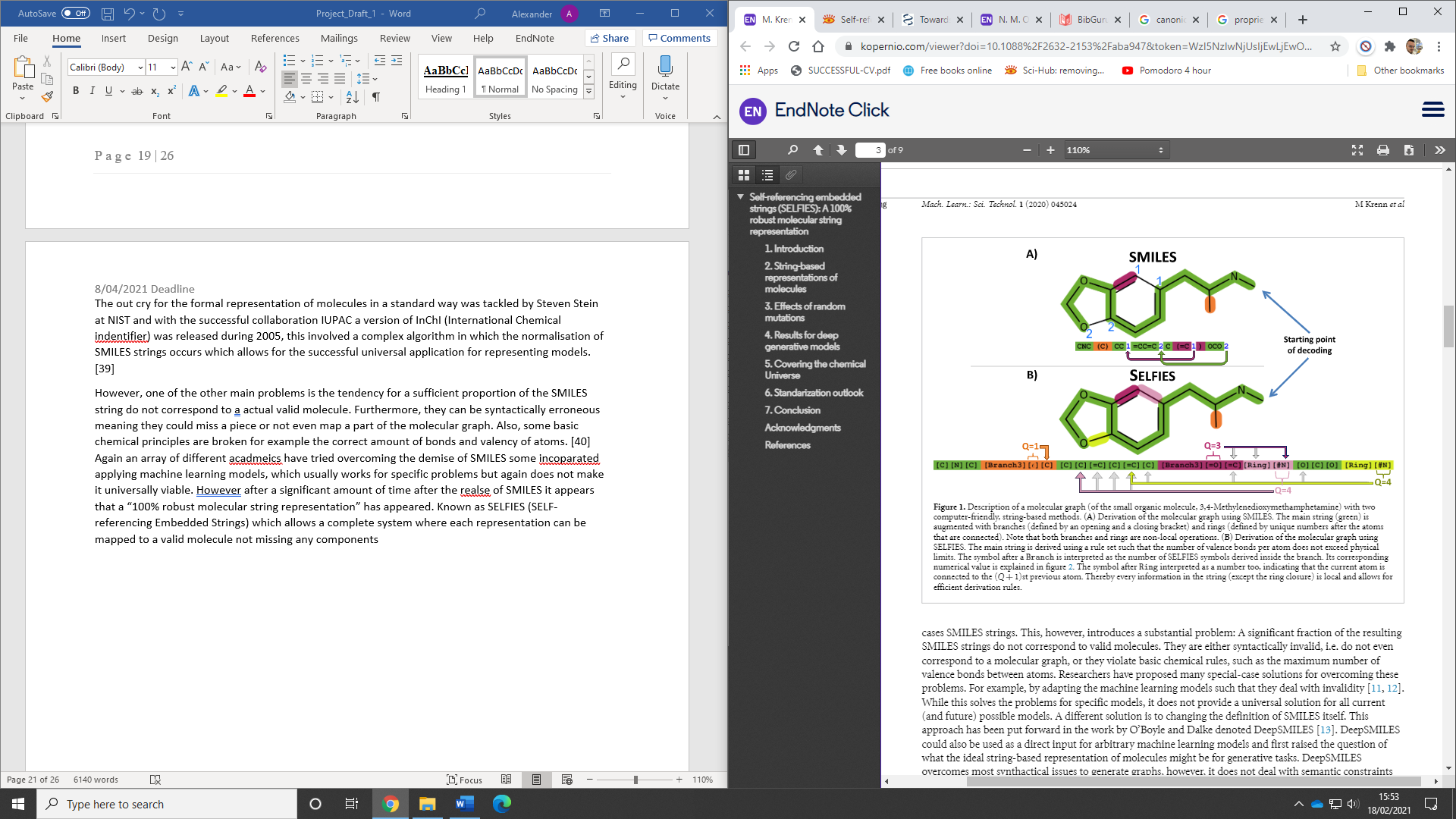
Another point of interest is the effect on the results obtained when the data set used from the CSV file was reduced from n=193 to n=87. It was observed in Table 2 that all the MAE, RMSE and R2 values in the n=87 data set increased. It is not straightforward to see why this occurs when the dataset is reduced. One possible explanation is as follows. The mean value for the molecular weight of 87 molecules is 49.22 g/mol whereas for the dataset of 193 molecules, the mean molecular weight is 82.16 g/mol. The importance of molecular weight and its effects on a solvent solute system is something that needs to be further investigated. As in the Sanchez et al paper [14] the maximum molecular weight of the molecules used in their data set are restricted to 1500 g/mol. Moreover, Sanchez et al [14] created a whole new model for polymers due to the different reaction in chemical space when small molecules are present, thus emphasising the importance of molecular weight on solubility. Also, it is worth noting how the physiochemical properties of a substance could change if a database was obtained for nanoparticles and how this would affect solvent and solute systems. For example, would a Non-polar substance exhibit a more polar character on the nanoscale?

**3.2 Limitation of SMILES**

One of the problems with the accuracy of the results could lie within the misrepresentation of molecular structure using SMILES. As of 2020, SMILES strings were used to represent molecules in 3D space, however, there have been some problems using the SMILES string system. The premise of the model is that a given molecule will fit the 2-electron valance model, therefore it does not represent stereochemical modalities well, and as of this moment in time there is no standardised way for the user to represent aromaticity. [39] There have been many attempts to standardise the system for the representations of molecules. Weininger et al., published an attempt to resolve the standardisation process however his system (CANGEN) excluded stereochemical descriptors. Moreover, Daylight created commerce for the generation of canonicalised SMILES system, but the algorithm generated was not open source and not available publicly. Furthermore, there have been lots of other attempts that have been made to standardise this system. Many of these open-source and publicly available endeavours have been created but none have been published or agreed upon. [39] This is of significance as with the standardisation a universal system can be used; for example, the nomenclature for naming compounds which is agreed upon be the IUPAC, which would allow users internationally to interact with each other and work together in a more concise collaborative fashion.

The dilemma for the formal representation of molecules in a standard way was tackled by Steven Stein at NIST (National Institute of Standards and Technology) and with the successful collaboration with the IUPAC an iteration of SMILES strings was created and called InChI (International Chemical identifier). This was released in 2005 and involves a complex algorithm in which the normalisation of SMILES strings happens, thus allowing for the successful integration and universal application for representing models. [39]

As previously mentioned in the introduction one of the other main problems with the SMILES system is the tendency for a sufficient proportion of the SMILES strings to not correspond to an actual valid molecule. Furthermore, they can be syntactically erroneous meaning there could be a missing piece of the string that may not even be mapped as a part of the molecular graph. Moreover, another discrepancy is that in some cases the fundamental laws of Chemistry are broken, an example of this is a carbon atom having 5 bonds when it only has the valency to support 4 bonds. [40] Yet again an array of different academics have tried to overcome the demises of SMILES, and some have even incorporated applied machine learning models, which usually works for specific problems but again does not make it universally viable. However, after a significant amount of time after the release of original SMILES framework it appears that a “100% robust molecular string representation” has appeared. It is known as SELFIES (SELF-referencing Embedded Strings) which allows a comprehensive system where each depiction of a system can be mapped to a valid molecule not missing any components or breaking any rules of valency or Chemistry. [40] Therefore leading to more accurate results in optimising and running programmes. More importantly being open source on GitHub it can be easily installed via a pip installation which is much more user friendly than other frameworks like RD-Kit. Lastly it offers a selfies.encoder function which translates SMILES into an equivalent SELFIES string.



*Figure 9. The successful complete mapping of MDMA by an iterative process starting at the same point [43].*

**3.3 General discussion on the scientific software installation**

Another problem faced during this project was the installation of the Scientific software. A user can easily install the anaconda framework onto their computer with the data analysis package which gives the user all the frameworks required for basic data analysis. However, when installing the RD-Kit cheminformatic framework, it was not as simple as press a button and install. Trying to follow the website was useless and using websites like Stack Overflow was not helpful, one of the main problems faced for this project was the knowledge gap in programming. However, eventually it was resolved by learning more about computer architecture, consulting supervision, and endless You Tube videos.

# Chapter 4. Conclusion, future work and general remarks

In this project, a Lasso linear regression-ML algorithm was applied to predict the Hansen solubility parameters of 193 solvent molecules. Four features have been used to describe the molecules, including their molecular fingerprints, charge density, electrostatics, and shape/size. It becomes apparent that there are more positive observable trends in the data when changing the data set to 87 molecules. However, these correlations do not imply causation as the errors rise and there is no proof or proven reasoning to explain what brings about the changes in the correlation. It is suggested that it could be because of the observable difference in the relative molecular weights of the different systems (87 and 193 molecules). However, this is just a suggestion and more work must be done. A more in-depth analysis of how different datasets, physicochemical factors and different quantum calculations should be taken into account; to explore with the intention of giving more accurate and reliable predictions in solubility.

Another point to consider moving forward is when using a Cheminformatic framework the user should only use SELFIES strings, and SMILES should be a figment of the past enabling no misrepresentation in data in modelling. A healthy combination of the above will start to help the unravelling of the mysteries of Chemical space.

Also, there is a need to explore how molecular weight and adjacent molecules can affect the solubility of a system, and a much larger molecular data set should be used and explored with the integration of Quantum calculations. The reality is that chemical space is huge and there is no way to navigate and harness it to understand solubility and intermolecular interactions.

With Industry 4.0 becoming a reality for many sectors in the world; the need and use of applying machine learning and automation to chemistry will revolutionise the speed and accuracy at which predictions are made for solubility and other chemical systems. One of the main limitations of driving digital chemistry forward is the non-standardisation of data there is a need to have a standardised and organised way to record, store and access data that is free and open access to all users. Furthermore, an attempt should be made to make it easier to install RD-kit and other Cheminformatic software’s and moving forward programming and basic computer architecture should be integrated into the earlier curriculum (Pre-18), and at a minimum as a part of a Chemistry degree, as soon it will be difficult to find the two subjects separate.

# References

1. Gambella C, Ghaddar B, Naoum-Sawaya J. Optimization Problems for Machine Learning: A Survey. European Journal of Operational Research [Internet]. 2020 [cited 6 September 2020];. Available from: https://www.sciencedirect.com/science/article/pii/S037722172030758X

2. Jiang T, Gradus J, Rosellini A. Supervised Machine Learning: A Brief Primer. Behavior Therapy. 2020;51(5):675-687.

3. Fernández A, Correa-Álvarez C, Pericchi L. Balancing producer and consumer risks in optimal attribute testing: A unified Bayesian/Frequentist design. European Journal of Operational Research. 2020;286(2):576-587.

4. Philippe Rigollet. 18.650 Statistics for Applications . Fall 2016. Massachusetts Institute of Technology: MIT OpenCourseWare, [https://ocw.mit.edu](https://ocw.mit.edu/courses/mathematics/18-650-statistics-for-applications-fall-2016). License: [Creative Commons BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/).

5. Zeltzer J. [Internet]. 2020 [cited 7 September 2020]. Available from: https://www.youtube.com/watch?v=Pahyv9i\_X2k&ab\_channel=zedstatistics

6. [Internet]. Math.caltech.edu. 2020 [cited 7 September 2020]. Available from: <http://www.math.caltech.edu/~2016-17/2term/ma003/Notes/Lecture04.pdf>

7. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. [Internet]. 2019 [cited 7 September 2020];Volume 947. Available from: https://www.sciencedirect.com/science/article/pii/S0168900219312021

8. Lecture 1: The Learning Problem [Internet]. 2020 [cited 7 September 2020]. Available from: <http://work.caltech.edu/slides/slides01.pdf>

9. Hands-On Unsupervised Learning Using Python [Internet]. O’Reilly Online Learning. 2020 [cited 7 September 2020]. Available from: https://www.oreilly.com/library/view/hands-on-unsupervised-learning/9781492035633/ch01.html

10. Cao Z, Zhang R, Hu X, Sha J, Jiang G, Li Y. Thermodynamic modelling, Hansen solubility parameter and solvent effect of oxaprozin in thirteen pure solvents at different temperatures. The Journal of Chemical Thermodynamics. 2020;151:106239.

11. Hansen C. The three dimensional solubility parameter and solvent diffusion coefficient. Copenhagen: Danish Technical Press; 1967.

12. J. H. Hildebrand and R. L. Scott, Solubility of Non-Electrolytes, 3rd ed. (Reinhold, New York, 1950; Dover, New York, 1964).

13. Jouyban A. Handbook of solubility data for pharmaceuticals. Boca Raton: CRC Press; 2010.

14. Sanchez‐Lengeling B, Roch L, Perea J, Langner S, Brabec C, Aspuru‐Guzik A. A Bayesian Approach to Predict Solubility Parameters. Advanced Theory and Simulations. 2018;2(1):1800069.

15. Letcher T. Development and applications in solubility. Cambridge: RSC Pub.; 2007.

16. Agata Y, Yamamoto H. Determination of Hansen solubility parameters of ionic liquids using double-sphere type of Hansen solubility sphere method. Chemical Physics. 2018;513:165-173.

17. Sehlleier Y, Dobrowolny S, Xiao L, Heinzel A, Schulz C, Wiggers H. Micrometer-sized nano-structured silicon/carbon composites for lithium-ion battery anodes synthesized based on a three-step Hansen solubility parameter (HSP) concept. Journal of Industrial and Engineering Chemistry. 2017;52:305-313.

18. Machine learning-accelerated quantum mechanics-based atomistic simulations for industrial applications

19. Müller A, Guido S. Introduction to machine learning with Python.

20. Data analysis and machine learning tools in MATLAB and Python. Computational Learning Approaches to Data Analytics in Biomedical Applications,. 2020;:231–290.

21. [Internet]. Rdkit.org. 2020 [cited 2 December 2020]. Available from: https://www.rdkit.org/RDKit\_Overview.pdf

22. Xiaohong D, Huajiang C, Bagherzadeh SA, Shayan M, Akbari M. Statistical estimation the thermal conductivity of MWCNTs-SiO2/Water-EG nanofluid using the ridge regression method [Internet]. Physica A: Statistical Mechanics and its Applications. North-Holland; 2019 [cited 2020Dec2]. Available from: https://www.sciencedirect.com/science/article/pii/S037843711931578X

23. LASSO: A feature selection technique in predictive modeling for machine learning - IEEE Conference Publication [Internet]. Ieeexplore.ieee.org. 2020 [cited 2 December 2020]. Available from: https://ieeexplore.ieee.org/abstract/document/7887916

https://link.springer.com/article/10.1007/s10822-020-00346-6

Python for Data Analysis by Wes McKinney (O’Reilly, 2012)

24. [Internet]. 2020 [cited 7 December 2020]. Available from: https://www.youtube.com/watch?v=eGXw9n7AnV4&ab\_channel=Stemplicity

25. sklearn.model\_selection.LeaveOneOut — scikit-learn 0.23.2 documentation [Internet]. Scikit-learn.org. 2020 [cited 8 December 2020]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.LeaveOneOut.html

26. Glossary — Toolkits -- Python [Internet]. Docs.eyesopen.com. 2020 [cited 9 December 2020]. Available from: <https://docs.eyesopen.com/toolkits/python/oechemtk/glossary.html#term-smarts>

27. Daylight Theory: SMARTS - A Language for Describing Molecular Patterns [Internet]. Daylight.com. 2020 [cited 9 December 2020]. Available from: https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

|  |  |
| --- | --- |
| 28. | Bergström CAS, Larsson P. Computational prediction of drug solubility in water-based systems: Qualitative and quantitative approaches used in the current drug discovery and development setting. Int J Pharm. 2018;540(1–2):185–93. |

|  |  |
| --- | --- |
| 29. | Przybyłek M, Jeliński T, Cysewski P. Application of multivariate adaptive regression splines (MARSplines) for predicting Hansen solubility parameters based on 1D and 2D molecular descriptors computed from SMILES string. J Chem. 2019;2019:1–15. |
| 30. | Yap CW. PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem. 2011;32(7):1466–74. |

|  |  |
| --- | --- |
| 31. | Gramatica P, Chirico N, Papa E, Cassani S, Kovarich S. QSARINS: A new software for the development, analysis, and validation of QSAR MLR models. J Comput Chem. 2013;34(24):2121–32. |

|  |  |
| --- | --- |
| 32. | Tropsha A. Best practices for QSAR model development, validation, and exploitation. Mol Inform. 2010;29(6–7):476–88. |

|  |  |
| --- | --- |
| 33. | Kensert A, Alvarsson J, Norinder U, Spjuth O. Evaluating parameters for ligand-based modeling with random forest on sparse data sets. J Cheminform. 2018;10(1):49. |

|  |  |
| --- | --- |
| 34. | Hu Y, Lounkine E, Bajorath J. Improving the search performance of extended connectivity fingerprints through activity-oriented feature filtering and application of a bit-density-dependent similarity function. ChemMedChem. 2009;4(4):540–8. |

|  |  |
| --- | --- |
| 35. | Mullins E, Oldland R, Liu YA, Wang S, Sandler SI, Chen C-C. Sigma-profile database for using COSMO-based thermodynamic methods. Ind Eng Chem Res. 2006;45(12):4389–415. |
| 36. | Klamt A. Conductor-like screening model for real solvents: A new approach to the quantitative calculation of solvation phenomena. J Phys Chem. 1995;99(7):2224–35. |

|  |  |
| --- | --- |
| 37. | Wikipedia contributors. Kernel method [Internet]. Wikipedia, The Free Encyclopedia. 2020 [cited 2021 Feb 16]. Available from: https://en.wikipedia.org/w/index.php?title=Kernel\_method&oldid=991967840 |

|  |  |
| --- | --- |
| 38. | Apsemidis A, Psarakis S, Moguerza JM. A review of machine learning kernel methods in statistical process monitoring. Comput Ind Eng. 2020;142(106376):106376. |

|  |  |
| --- | --- |
| 39. | O’Boyle NM. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. J Cheminform. 2012;4(1):22. |

|  |  |
| --- | --- |
| 40. | Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. Mach Learn: Sci Technol. 2020;1(4):045024. |
| 41. | Süß S, Sobisch T, Peukert W, Lerche D, Segets D. Determination of Hansen parameters for particles: A standardized routine based on analytical centrifugation. Adv Powder Technol. 2018;29(7):1550–61. |
| 42. | www. Hansen-Solubility.com. The HSP Sphere [Internet]. Hansen-solubility.com. [cited 2021 Apr 28]. Available from: <https://www.hansen-solubility.com/HSP-science/sphere.php> |
| 43. | Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. Mach Learn: Sci Technol. 2020;1(4):045024. |

|  |  |
| --- | --- |
| 44. | Mullins E, Oldland R, Liu YA, Wang S, Sandler SI, Chen C-C, et al. Sigma-profile database for using COSMO-based thermodynamic methods. Ind Eng Chem Res. 2006;45(12):4389–415. |

|  |  |
| --- | --- |
| 45. | Automated Intelligent Chemistry [Internet]. Soci.org. [cited 2021 Apr 28]. Available from: <https://www.soci.org/events/automated-intelligent-chemistry> |
| 46. | Wikipedia contributors. Kernel method [Internet]. Wikipedia, The Free Encyclopedia. 2021 [cited 2021 Apr 28]. Available from: <https://en.wikipedia.org/w/index.php?title=Kernel_method&oldid=1012624126> |
| 47. | Code repository to build models and predict, along with datatsets can  be found at https://github.com/aspuru-guzik-group/gpHSP. |