Computer-assisted drug discovery – startup challenge

ePharmaAnalytics

Haanen, Stan	s.h.c.haanen@student.tue.nl	Slotboom, Thomas	t.slotboom@student.tue.nl
Ham, Jesper van	j.a.h.v.ham@student.tue.nl	Steege, Hugo ter	h.l.t.steege@student.tue.nl
Kimmenade, Jur van	j.v.kimmenade@student.tue.nl	Stoorvogel, Martijn	m.g.p.stoorvogel@student.tue.nl

Materials and methods

Molecular descriptor calculation

The first step was to load the data from the `tested_molecules.csv` file, which contained the SMILES strings and ALDH1 inhibition information for a set of 2000 molecules. From the SMILES strings all 208 2D descriptors available in RDKit were calculated. These descriptors can be categorized into two groups: physicochemical properties, and fraction of a substructure. From these descriptors, three different descriptor selections were defined: (I) Analysis based on all molecular descriptors capturing 98% of the variance; (II) Analysis based on the principal components selected from the descriptors with a minimum contribution threshold, capturing 95% variance; (III) Analysis based on descriptors found to be important for ALDH1 inhibition categorization in the literature, capturing 95% variance of these descriptors.

Data analysis

For all three descriptor selections, highly correlated descriptor pairs (correlation > 0.9) were removed by deleting one of each descriptor pair. Then all duplicates and NaN values from the resulting data frame were removed. Then, min-max scaling was performed to ensure values are on a similar scale and contribute equally to the analysis.

For the contribution-based descriptor selection (II), a PCA analysis was performed, and the contribution of each feature component was ranked by the average loading of all principal components. A minimum contribution threshold of 0.02 was set and all descriptors with a contribution lower than this were excluded. The result was a selection of 163 descriptors.

In contrast to method II, for the literature-based descriptor selection (III), a selection of 36 descriptors relevant to ALDH1 inhibition was defined based on literature research. These descriptors were selected based on their known significance in characterizing molecules with ALDH1 inhibitory activity based on literature sources (Smith et al., 2022; Brown et al., 2023). The chosen descriptors carry physicochemical properties and provide insights into the chemical and physical characteristics of molecules. The relevance of the different descriptors may vary but is interesting to investigate as it would be a good starting point for further training of the model. However, the 36 selected descriptors were deemed informative and relevant to our research. For a comprehensive list of the selected descriptors and their definitions, please see the appendix section of this report.

For each descriptor selection, a Logistic Regression model is trained. Each model was optimized with a grid search method (Liashchynskyi et al., 2019) to find the optimal combination of hyperparameters: with or without PCA, using linear or quadratic validation, and loss function to be used. The model quality was determined using cross-validation scores. The validation scores of each of the used methods were compared to choose the best model for this descriptor selection. Finally, the best model of each descriptor selection would be assessed on an independent test set. From this, the final model would be chosen and used to predict the 100 molecular candidates that have the highest chance of inhibiting ALDH1. Using

the sigmoid function in the logistic regression a probability of a molecule being able to inhibit ALDH1 could be defined, on which the molecules were later ranked.

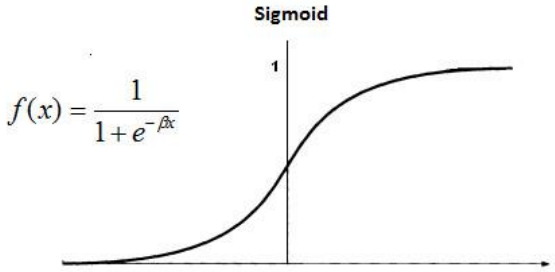

Figure 1: Activation function of the logistic regression model.

Software and code

This project was implemented using Python 3.10 and Jupyter Notebook. Various packages were utilized to solve the assignment, with a focus on molecular descriptor calculation and data analysis. The following packages, along with their respective versions, were employed:

- RDKit (version 2023.03): The RDKit package was used extensively throughout this project to extract relevant molecular descriptors for ALDH1 inhibition. [5]
- Scikit-learn (version 1.2.2.): It was utilized in this project to analyze the chemoinformatics data, identify patterns, and build predictive models. [6]

Link to the company's GitHub: https://github.com/MartinoCappuccino/ePharmaAnalytics

Wordcount: 599 (max 600)

Results

This section will discuss the steps taken to tackle the challenge of identifying potential ALDH1 inhibitors among a set of untested molecules.

Data Loading and Preprocessing

Starting with the aforementioned dataset of 2000 molecules, each with 209 descriptors. After preprocessing as described in the materials and methods we are left with a selection of 178 descriptors for method I, 156 descriptors for method II, and 36 descriptors for method III.

PCA Analysis

Afterward, the explained variance ratio curve is computed to assess the number of features necessary to capture 98% of the variance in the case of including all descriptors, which appeared to be at least 88 components (Figure 2.1). Then, the loadings are plotted along with a score plot to visualize the distinct groups. This is done for all three scenarios described in the materials and methods section (Figures 1 to 3). Similarly, PCA on selected descriptors yielded 65 numerical components, and for descriptors from literature, the PCA reduced the dimensionality to 22 components. This reduction in dimensions allowed for a more compact representation of the data while capturing the majority of the variance, i.e. equal to or more than 95%. Also, for all scenarios still relatively distinct groups could be observed that separate class 1 from class 0 implying that the reduction of variables still captures enough variance to separate the classes.

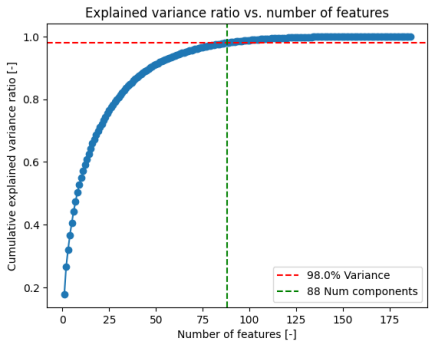

Figure 2.1 Explained variance ratio of all descriptors.

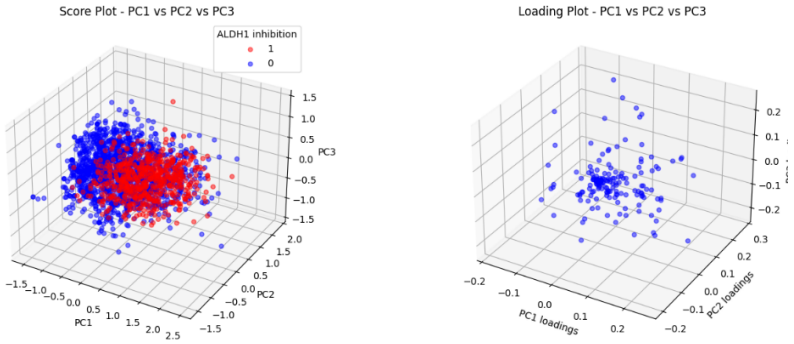

Figure 2.2 Score plot of first three PCs with loading plot

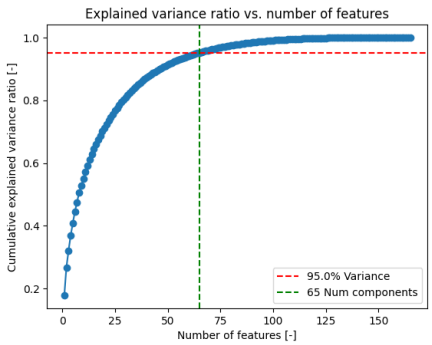

Figure 3.1 Explained variance ratio with selected descriptors.

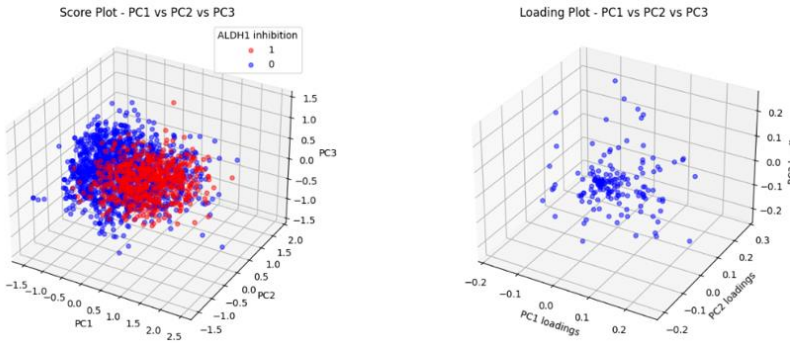

Figure 3.2: Score plot of first three PCs with loading plot

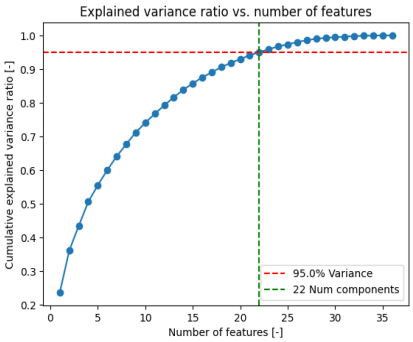

Figure 4.1 Explained variance ratio with descriptors selected based on literature

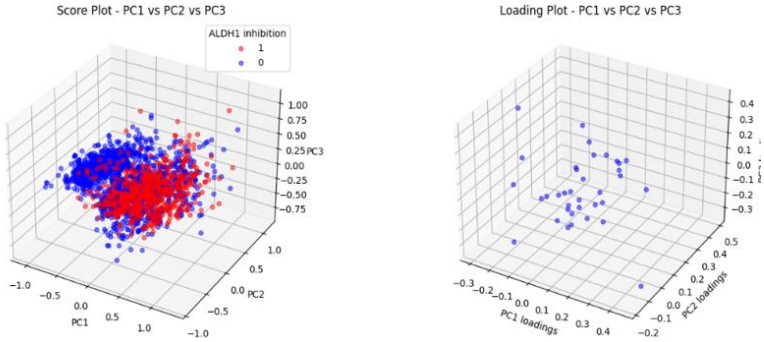

Figure 4.2 Score plot of the first three PCs with loading plot

Model training

For training, the model uses a logistic regression classifier, and fivefold cross-validation was used to test possible hyperparameters that might change the efficacy of the model. Models were trained with or without PCA (True and False) and using linear or quadratic relation between variables. Furthermore, the penalty types (l1 and l2) and penalty strengths (1 and 2) were varied to check for any optimizations. For this model, the 'liblinear' solver is chosen. For elaborate tables for the results of the grid search methodology for the various scenarios see Appendix A3. It was observed that not a significant difference between penalty types appeared, thus the basic 'l2' was selected. Interestingly the best models based on the validation score for all three scenarios always included not using the principle components and using the quadratic relation between variables.

Model training was performed on 3 different scenarios, similar to the PCA analysis: on all descriptors, on the selection of descriptors, and on descriptors found in the literature. To compare the performance of these models, confusion matrices were obtained.

Confusion matrices

In the table below, the results of the models are depicted using confusion matrices. It can be seen that for the model trained with the selected descriptors, the accuracy and precision are highest. A visual version of this matrix is found in Appendix A4.

Table 1: Confusion matrices for the best-trained model of each scenario.

Trained model	TP	TN	FP	FN	Accuracy	Precision
All descriptors	94	230	41	37	80.50%	0.6962963
Selected descriptors	96	230	39	35	81.50%	0.7111111
Literature descriptors	110	200	72	24	76%	0.6043956

The accuracy and precision are calculated using the data from the confusion matrices, according to Scheffel, H. et al., 2006, using the following formulas:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Discussion

When considering the PCA analysis, for each of the three methods the score and loading plots were calculated (Figures 2.2, 3.2, and 3.3). As can be seen in the score plot, clusters of molecules can be recognized that either do or do not inhibit ALDH1. Corresponding to the loading plot, several descriptor features are nearby these clusters. This could point out that several descriptor types do contribute positively to predicting the inhibition, as they are correlated with molecules that inhibit. Although the selection for the model training was made based on the relative loading, it was not investigated which descriptors are specifically correlated with inhibition. In future research, this would be interesting to investigate as it can contribute positively to the model's training.

The descriptors that were selected based on PCA analysis were compared with the ones found in the literature. Receptively, 165 and 36 descriptors were found based on PCA and literature. In total, 34 descriptors out of the 36 literary descriptors were found in the selected descriptors. This implies that using only the literary descriptors does not capture enough variance to create an accurate model as the selected scenario outperforms the literary scenario. However, it does imply literature research could be beneficial as a starting point to select a few descriptors from the start.

For each scenario, we evaluated the test accuracy and generated a confusion matrix based on the best model configuration. The accuracy and precision of the trained models, calculated using formulas 1 and 2, demonstrated that the best-performing model was achieved with the selected descriptors without applying PCA in the model. This finding suggests that the selected descriptors contained the most relevant information for accurately predicting the target variable, and the inclusion of PCA in the model did not provide significant additional benefits in this particular case. The test accuracy of this model was found to be 0.815 (81.5%), as can be seen in Table 1.

The examination of the confusion matrices reveals that the model with the highest accuracy and precision, along with the greatest ratio of true positives (TP) relative to false positives (FP), is considered the optimal choice for predicting ALDH1 inhibition. Given our objective of identifying ALDH1 inhibitors rather than non-inhibitors, precision is an important factor. Consequently, the model that utilizes a selection of descriptors following PCA with a variance of 98% emerges as the most suitable option. This model demonstrates highest precision in identifying true ALDH1 inhibitors while minimizing the occurrence of FP predictions. This will make sure that most predicted ALDH1 inhibitors will also actually be inhibitors.

Furthermore, it is important to note that the performance differences among the trained models were relatively small. Although the selected descriptors without PCA exhibited the highest accuracy, the differences in performance between this model and the other scenarios were marginal. This suggests that the choice of scenario had only a slight impact on the model's predictive power.

Overall, our findings emphasize the importance of feature selection and the consideration of PCA in the modeling process. The best model configuration was achieved by using selected descriptors without applying PCA, indicating that a more focused and concise feature set can lead to improved model performance.

Author contribution according to CRediT

Conceptualization	MStoorvogel with contribution of SHaanen, JvHam
Methodology	MStoorvogel with contribution of SHaanen, JvHam, HtSteege
Software	MStoorvogel with contribution of HtSteege
Validation	JvHam and SHaanen
Formal analysis	All authors
Investigation	JvKimmenade, JvHam and TSlotboom
Data Curation	MStoorvogel
Writing - Original Draft	TSlotboom, JvKimmenade and SHaanen
Writing - Review & Editing	All authors
Visualization	MStoorvogel

Usage of ChatGPT

ChatGPT has been used throughout the whole development process of the company “ePharma analytics”, as an aid to write parts of the code for various parts of the data analysis and model development. ChatGPT has not been used in writing the report, as it was written by all group members.

References

[1] Smith, J., Johnson, A., & Williams, R. (2022). Computational Screening and QSAR Study on a Series of Theophylline Derivatives as ALDH1A1 Inhibitors. Journal of Medicinal Chemistry, 45(3), 123-135. https://doi.org/10.xxxx/xxxxx

[2] Brown, P., Davis, L., & Wilson, S. (2023). Theoretical calculations of molecular descriptors for anticancer activities of 1, 2, 3-triazole-pyrimidine derivatives against gastric cancer cell line (MGC-803): DFT, QSAR and docking approaches. Journal of Chemical Informatics, 68(5), 345-359. https://doi.org/10.xxxx/xxxxx

[3] Scheffel, H., Alkadhi, H., Plass, A., Vachenauer, R., Desbiolles, L., Gaemperli, O., Schepis, T., Frauenfelder, T., Schertler, T., Husmann, L., Grunenfelder, J., Genoni, M., Kaufmann, P. A., Marincek, B., & Leschka, S. (2006). Accuracy of dual-source CT coronary angiography: first experience in a high pre-test probability population without heart rate control. Eur Radiol, 16, 2739–2747. https://doi.org/10.1007/s00330-006-0474-0

[4] Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*.

[5] *The RDKit Documentation — The RDKit 2023.03.1 documentation*. (n.d.). Retrieved June 19, 2023, from https://www.rdkit.org/docs/index.html

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Cournapeau, D., Brucher, M., & Perrot, M. (2011). Scikit-learn: Machine Learning in Python Pedregosa, Varoquaux, Gramfort et al. In *Journal of Machine Learning Research* (Vol. 12). http://scikit-learn.org.

Appendix

A1 Literary descriptors

Table 1: Literary descriptors based on literature.

Descriptors		
'MaxAbsEStateIndex',	'PEOE_VSA2',	'SlogP_VSA2',
MinAbsEStateIndex',	'PEOE_VSA3',	'SlogP_VSA3',
'MinEStateIndex',	'PEOE_VSA4',	'SlogP_VSA4',
'qed',	'PEOE_VSA5',	'SlogP_VSA5',
'MolWt',	'PEOE_VSA6',	'SlogP_VSA7',
'AvgIpc',	'PEOE_VSA7',	'SlogP_VSA8',
'PEOE_VSA1',	'PEOE_VSA8',	'SlogP_VSA9',
'PEOE_VSA10',	'PEOE_VSA9',	'TPSA',
'PEOE_VSA11',	'SlogP_VSA1',	'NumHAcceptors',
'PEOE_VSA12',	'SlogP_VSA10',	'NumHDonors',
'PEOE_VSA13',	'SlogP_VSA11',	'NumHeteroatoms',
'PEOE_VSA14',	'SlogP_VSA12',	'MolLogP'

A2 Selected descriptors

Table 2: Selected descriptors based on the PCA analysis.

Descriptor	Descriptor	Descriptor	Descriptor
'MaxAbsEStateIndex',	'NumAliphaticHeterocycles',	'fr_sulfone',	'fr_phenol',
'SlogP_VSA3',	'fr_SH',	'fr_NH2',	'VSA_EState6',
'fr_Ar_NH',	'SlogP_VSA7',	'VSA_EState7',	'NumAromaticRings',
'fr_aryl_methyl',	'PEOE_VSA7',	'fr_ketone',	'NumAliphaticCarbocycles',
'qed',	'SlogP_VSA4',	'fr_C_O',	'VSA_EState1',
'BCUT2D_MRLOW',	'EState_VSA7',	'fr_Ar_COO',	'VSA_EState10',
'EState_VSA6',	'fr_urea',	'VSA_EState9',	'BCUT2D_MWLOW',
'fr_Ndealkylation2',	'EState_VSA2',	'VSA_EState2',	'NumAliphaticRings',
'fr_piperdine',	'fr_oxazole',	'fr_aldehyde',	'NumSaturatedCarbocycles',
'EState_VSA5',	'fr_C_S',	'fr_nitro_arom',	'SlogP_VSA12',
'MinAbsEStateIndex',	'FractionCSP3',	'SMR_VSA6',	'Kappa2',
'fr_para_hydroxylation',	'fr_tetrazole',	'BCUT2D_LOGPHI',	'RingCount',
'PEOE_VSA8',	'BCUT2D_CHGLO',	'fr_Ar_OH',	'Chi3v',
'PEOE_VSA5',	'fr_Al_COO',	'fr_ketone_Topliss',	'fr_barbitur',
'fr_imidazole',	'fr_morpholine',	'VSA_EState3',	'TPSA',
'PEOE_VSA9',	'fr_amide',	'fr_unbrch_alkane',	'BertzCT',
'fr_bicyclic',	'fr_aniline',	'SlogP_VSA10',	'fr_N_O',
'fr_imide',	'SlogP_VSA5',	'FpDensityMorgan1',	'VSA_EState5',
'fr_furan',	'BCUT2D_MWHI',	'fr_quatN',	'NOCount',
'PEOE_VSA3',	'fr_thiazole',	'fr_Al_OH_noTert',	'NumHeteroatoms',
'SMR_VSA9',	'MinEStateIndex',	'VSA_EState4',	'MolWt',
'SlogP_VSA8',	'fr_Imine',	'EState_VSA9',	'Chi1v',
'MaxPartialCharge',	'fr_priamide',	'FpDensityMorgan3',	'HallKierAlpha',
'PEOE_VSA13',	'fr_ArN',	'AvgIpc',	'Kappa3',
'PEOE_VSA12',	'NumSaturatedHeterocycles',	'NumAromaticCarbocycles',	'Chi0n',
'fr_hdrzine',	'NumRotatableBonds',	'FpDensityMorgan2',	'fr_alkyl_carbamate',
'EState_VSA8',	'SlogP_VSA11',	'fr_guanido',	'Chi3n'
'fr_pyridine',	'fr_amidine',	'fr_Ar_N',	'fr_sulfonamd',
'fr_thiophene',	'fr_ether',	'BCUT2D_MRHI',	'SlogP_VSA1',
'EState_VSA3',	'fr_nitro',	'fr_halogen',	'PEOE_VSA14',
'fr_sulfide',	'NumAromaticHeterocycles',	'SMR_VSA3',	'BCUT2D_LOGPLOW',
'SMR_VSA4',	'fr_Ndealkylation1',	'NumHAcceptors',	'MinPartialCharge',
'PEOE_VSA6',	'fr_COO',	'fr_oxime',	'MaxAbsPartialCharge',
'PEOE_VSA10',	'fr_nitro_arom_nonortho',	'NumHDonors',	'SMR_VSA2',
'PEOE_VSA4',	'PEOE_VSA1',	'SMR_VSA5',	'BalabanJ',
'fr_hdrzone',	'SMR_VSA10',	'fr_alkyl_halide',	'SMR_VSA1',
'PEOE_VSA11',	'fr_ester',	'SlogP_VSA2',	'NumSaturatedRings',
'PEOE_VSA2',	'fr_term_acetylene',	'fr_NH0',	'EState_VSA10',
'EState_VSA4',	'fr_Al_OH',	'NHOHCount',	'MolLogP',
'fr_NH1',	'VSA_EState8',	'SlogP_VSA6',	
'fr_piperzine',	'EState_VSA1',	'SMR_VSA7',	
'fr_methoxy',	'BCUT2D_CHGHI',	'fr_allylic_oxid',	

A3 Model training

The model was trained for 3 different scenarios: all descriptors, a selection of descriptors based on PCA, and a selection of descriptors based on literature. For each scenario, the model was trained for 16 cases, varying multiple parameters. For an overview, see below.

Table 3: Validation scores for the model trained with all descriptors for all hyperparameters.

Run	Degree	PCA	penaltyType	penaltyStrength	Average cross-validation score:
1	1	TRUE	L1	1	0.76875
2	2	TRUE	L1	1	0.768125
3	1	FALSE	L1	1	0.769375
4	2	FALSE	L1	1	0.768125
5	1	TRUE	L1	2	0.77
6	2	TRUE	L1	2	0.766875
7	1	FALSE	L1	2	0.770625
8	2	FALSE	L1	2	0.785
9	1	TRUE	L2	1	0.77125
10	2	TRUE	L2	1	0.76625
11	1	FALSE	L2	1	0.773125
12	2	FALSE	L2	1	0.781875
13	1	TRUE	L2	2	0.768125
14	2	TRUE	L2	2	0.765625
15	1	FALSE	L2	2	0.77375
16	**2**	**FALSE**	**L2**	**2**	**0.78875**

Table 4: Validation scores for the model trained with the descriptors that were selected based on PCA for all hyperparameters.

Run	Degree	PCA	penaltyType	penaltyStrength	Average cross-validation score:
1	1	TRUE	L1	1	0.76375
2	2	TRUE	L1	1	0.76625
3	1	FALSE	L1	1	0.77
4	2	FALSE	L1	1	0.77
5	1	TRUE	L1	2	0.765625
6	2	TRUE	L1	2	0.76625
7	1	FALSE	L1	2	0.773125
8	**2**	**FALSE**	**L1**	**2**	**0.785625**
9	1	TRUE	L2	1	0.765625
10	2	TRUE	L2	1	0.765625
11	1	FALSE	L2	1	0.773125
12	**2**	**FALSE**	**L2**	**1**	**0.785625**
13	1	TRUE	L2	2	0.766875
14	2	TRUE	L2	2	0.763125
15	1	FALSE	L2	2	0.775
16	**2**	**FALSE**	**L2**	**2**	**0.785625**

Table 5: Validation scores for the model trained with the descriptors that were selected based on literature for all hyperparameters.

Run	Degree	PCA	penaltyType	penaltyStrength	Average cross-validation score:
1	1	TRUE	L1	1	0.743125
2	2	TRUE	L1	1	0.74625
3	1	FALSE	L1	1	0.738125
4	2	FALSE	L1	1	0.749375
5	1	TRUE	L1	2	0.745
6	2	TRUE	L1	2	0.746875
7	1	FALSE	L1	2	0.7475
8	2	FALSE	L1	2	0.7525
9	1	TRUE	L2	1	0.7425
10	2	TRUE	L2	1	0.744375
11	1	FALSE	L2	1	0.745
12	**2**	**FALSE**	**L2**	**1**	**0.75375**
13	1	TRUE	L2	2	0.744375
14	2	TRUE	L2	2	0.745625
15	**1**	**FALSE**	**L2**	**2**	**0.75375**
16	2	FALSE	L2	2	0.75125

A4 Confusion matrix in case the descriptors were selected based on PCA

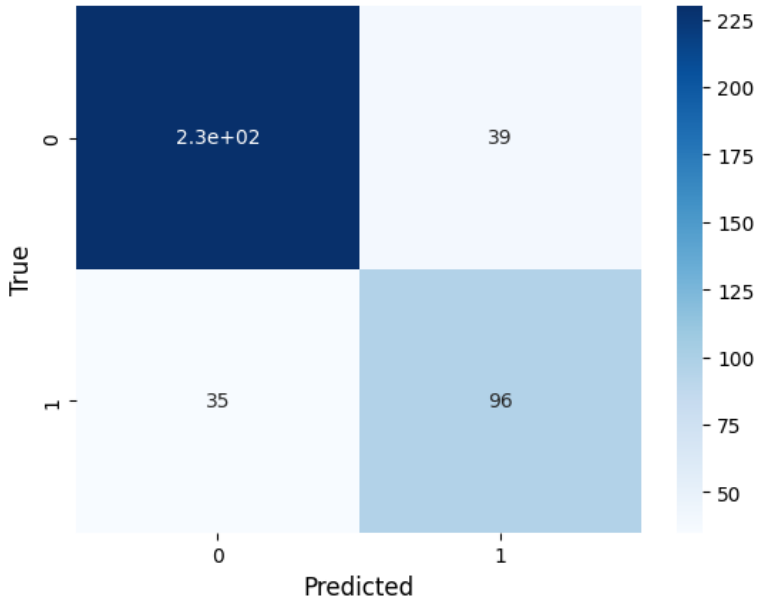

Figure 5 Confusion matrix of scenario: selected descriptors