Pattern Recognition Assignment 2

Sam Reswinraj Abraham (s4248325)
Faculty of Science and Engineering
University of Groningen)
Groningen, The Netherlands
a.sam.reswin.raj@student.rug.nl

Maniraj Sai Adapa (s4574842)
Faculty of Science and Engineering
University of Groningen)
Groningen, The Netherlands
m.s.adapa@student.rug.nl

Thijs Visee (s2982129)
Faculty of Science and Engineering
University of Groningen)
Groningen, The Netherlands
t.p.visee@student.rug.nl

Niels Rocholl (s3501108)
Faculty of Science and Engineering
University of Groningen)
Groningen, The Netherlands
n.m.m.rocholl@student.rug.nl

***Abstract*—High-dimensional numerical data and Image datasets are challenging datasets that can be used for classification and clustering problems. In this work one such dataset for each category is chosen and complete end-to-end machine learning pipeline's are developed to analyze, improve and achieve high performance models.**

***Index Terms*—high-dimensional, numerical data, image dataset, pipeline, classification, clusterig**

I. Introduction

This paper aims to give an overview of three classical pattern recognition pipelines. These pipelines implement clustering and classification algorithms. Two pipelines focus on numerical gene data, and one on image data. The gene data represents the RNA-Seq of gene expressions of *801* patients having five different types of tumors - *"BRCA"*, *"COAD"*, *"KIRC"*, *"LUAD"* & *"PRAD"*. The image data consists of *170 RGB* and *greyscale* images in various shapes and sizes. The images display one of five different classes of big cat breeds. The goal is to extract useful features from both these datasets in order to classify and cluster unseen gene expressions and image samples.

II. Related work

1) Gene Expression Classification: M.Jansi Rani and D.Devaraj [8] proposed a two stage Mutual Information-Genetic Algorithm (MI-GA) gene selection algorithm for selecting informative genes in cancer data classification. For classification itself Support Vector Machine (SVM) is used, and the model is applied on Colon, Lung and Ovarian cancer datasets.

Other works [6] show more research on gene classification with emphasis on cost - misclassification cost, test cost and rejection cost using rotation forest. The work in [5] provides a hybrid feature selection algorithm for gene expression data classification using Mutual Information Maximization(MIM) and Adaptive Genetic Algorithm (AGA). A Dynamic Feature Extraction (DFE) method was proposed In [12] for high-dimensional numerical data.

2) Gene Expression Clustering: With there being a surge of various genomic data produced, biologists have faced several problems in organising the observed data into meaningful structures. Consequently there has been a lot of research put into genome clustering over the last few decades which has resulted in various clustering methods. Thalamuthu, Anbupalam, et al. [11] study four of these prominent clustering methods which include hierarchical clustering, K-means clustering, Partitioning around medoids (PAM), Self organising maps (SOM) on simulated and real datasets. They found out that K-means and PAM perform almost the same and do a better job than SOM and hierarchical clustering. They also discovered that Hierarchical clustering and SOM are known to provide better visualization but by doing so they seem to have sacrificed performance. [1] successfully implemented DB-scan to cluster gene expression sequences of lung cancer patients.

III. Method

A. Pipeline 1: Gene Classification

1) Data Analysis: The used dataset is from - "Genes.zip", it consists of gene expressions of 801 patients with five different kinds of tumors - BRCA, COAD, KIRC, LUAD,PRAD. A total of 20531 genes make up the gene expressions for a single patient. The composition of various samples from each of the five classes is graphically presented in Figure 2 and it is evident this is not a completely balanced dataset. The 20531 genes (Figure 2) making up the feature space for the proposed classification model is quite huge. To overcome the curse of dimensionality and improve prediction accuracy, features have to be reduced. Upon Exploratory Data Analysis (EDA), there were no null values found in the dataset (features and labels), hence no further preprocessing was required.

2) Framework for Pipeline Component Selection:

- Feature selection - MI, PCA
- Classification model - Decision Tree, Random Forest, K-Nearest Neighbors
- Grid search - RandomisedSearchCV, GridSearchCV

- Evaluation - F1-score, Confusion matrix
- Validation - LOOCV, K-fold
- Data Augmentation - SMOTE, Borderline SMOTE
- Ensemble - Voting Based

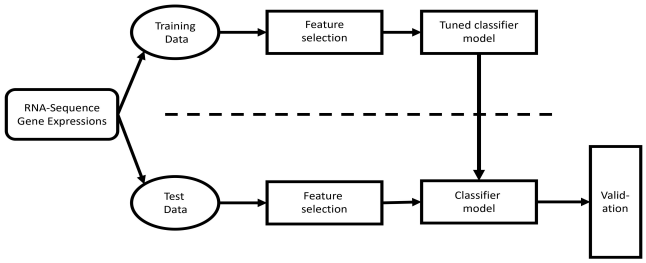


Fig. 1: Pipeline-1 Framework

a) Mutual Information: Mutual Information(MI) is a feature selection technique which works by taking a measure of mutual dependence between two variables in the feature space, i.e. how much of variable X is related to variable Y. This measure is symmetrical - $I(X;Y) = I(Y;X)$.

b) Principal Component Analysis: PCA is another feature selection method which is also an unsupervised linear transformation technique. It finds the variance in all directions between all features and it projects the ones with the highest variance onto a low dimensional subspace.

c) Decision Tree Classifier: Decision tree's (DT) are supervised learning technique, which is a tree-structured classifier with - root, internal node, edges and leafs.Attributes for DT are selected using - Information gain : $InformationGain = Entropy(S) - [(WeightedAvg) * Entropy(eachfeature)$ or Gini Index: metric of how much impurity or purity used while creating a decision tree.

d) Random forest Classifier: Random forest classifier consists of a large number of uncorrelated individual decision trees, with each individual tree making a class prediction that is evaluated with the other trees and the class with most votes is finalized as the model's prediction.

e) KNN Classifier: K-Nearest Neighbor classifier works based on existing data (based on class) in proximity to the test data. Datasets with similar class labels in the vicinity would mean the test data to belong to that particular class label.

f) Data Augmentation: Two different variants of Synthetic Minority Over-sampling Technique (SMOTE) [2] will be used to augment this dataset. Original SMOTE - creates synthetic data between each minority class sample and its "k" closest neighbors. Borderline-SMOTE [2] - creates synthetic data by only examining samples that make up the boundary that separates one class from another.

g) Library used: - Scikit-learn (sklearn) a python based machine learning library was used throughout the complete pipeline development.

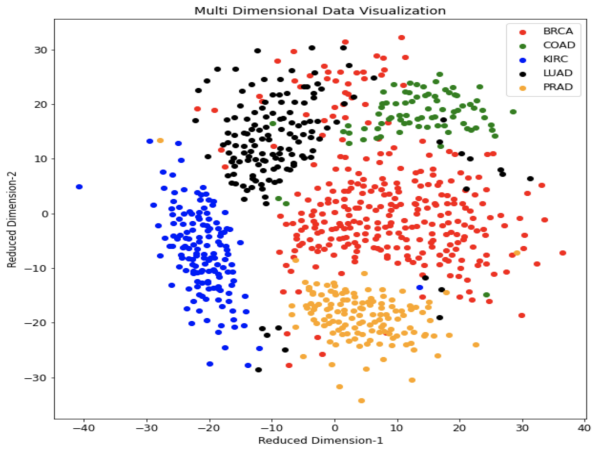


Fig. 2: Gene Expression - Data Visualization

3) Feature Selection: From literature preview Mutual Information (MI) based feature extraction works very well on high-dimensional numerical datasets. To compare baseline performance on a Decision tree model for classification, PCA was chosen against MI. MI-scores are calculated using the $mutual\_info\_classif$ library. Upon calculation 20531 scores are saved as an array. MI score ranges from 0-1 with higher the value representing the best possible available information for classification in a particular feature. The highest MI-score returned to be 0.96038, this was for $gene_{7964}$. To avoid overfitting or underfitting, features with a MI-score of more than 0.5 (Figure 4 were chosen. A total of 474 features were selected. For PCA, the dataset (not labels) is scaled and then transformed to fit 95% variance (Figure 3). It is recommended to select variance between 0.95% - 0.98% for the dataset to reproduce relevant information retained from the original dataset. A reduced feature set of 530 features is returned. Both the reduced datasets are fed into a baseline Decision tree classifier with no hyper parameter tuning, mean accuracy was used as an evaluation tool.

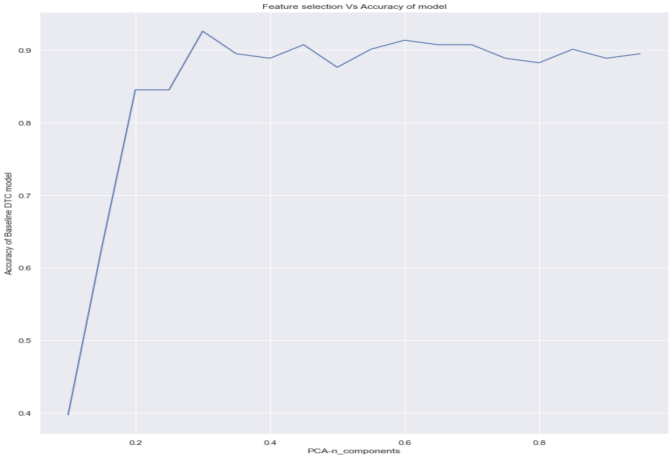


Fig. 3: PCA, n_components selection

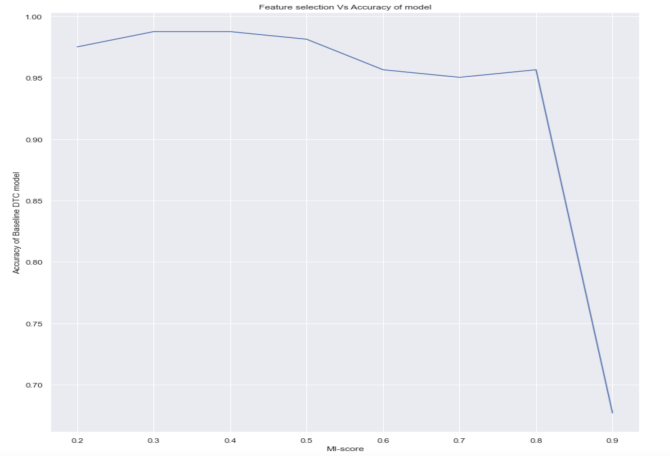


Fig. 4: MI, mi_score selection

4) Classification: From literature preview, Support Vector Machine was the go-to option for classifying gene data. To understand why not the Decision tree classifier or Random forest classifier, these two classifiers will be used in comparison with the K-Nearest Neighbors classifier model. For all the three classification algorithms, the dataset will be split into training data and test data. The ratio will be 80% & 20% with the split being stratified. Meaning, the ratio of classes in the original unbalanced dataset will be maintained in the train and test splits. Each of the models will also be using sklearn's RandomizedSearchCV and GridSearchCV for hyper parameter tuning, along with evaluation based on confusion matrix and f1-scores. Each classifier will be compared with its baseline not-tuned model to understand the effect of hyperparameter tuning.

a) Decision Tree Classifier: There are several hyper-parameters that need to be tuned for best possible model performance. The available parameters list is given in Table I, since most of the parameters are difficult to be known before hands as it requires in-depth knowledge of the dataset and the algorithm the parameters were tried with a range of values Table I using RandomisedSearchCV. This will try a random combination of parameters within the feature space to see what works best.

After the random search best possible parameters are viewed, along with the results of the first 10 iterations out of the 100 total iterations done for the random search. These 10 observations provide insight on how the range of values for the hyper parameters affect the mean test score. i.e. if a parameter value is increased resulting in a decreased mean test score would imply lower value sets are best to be used, if a parameter does not change the effect on a model after a point it can be considered threshold, etc.. Following this a more exhaustive intensive search is done using GridSearchCV. This is computationally costly and time consuming but it churns out the best available combination from the initial inference form the random search. The top 5 mean score parameter sets are further analyzed Table I before selecting the final hyper parameter value set for the model.

Hyper-parameter	Value Range
criterion	gini, entropy
max_depth	random integer value (1-10)
min_samples_leaf	integer value (1-50, step: 10)
min_samples_split	integer value (1-50,step: 10)
random_state	random integer value (1-10), None
max_leaf_nodes	random integer value(1-200), None
min_impurity_decrease	float value (0.0-0.9, step: 20)
splitter	best, random
max_features	auto, sqrt, log2, None
classweight	balanced, None

TABLE I: DTC Hyper-parameters (RandomizedSearchCV)

b) Random Forest Classifier: All available parameters (Table II) were tried and tested in the random search. Fist 30 iterations are observed to understand the correlation between the parameter values and mean test scores. After selecting a narrowed down range for GridSearchCV, the search is initiated and the top 5 parameter set is obtained and best fit is used on the model.

Hyper-parameter	Value Range
n_estimators	integer value (100-2000,step: 20)
criterion	gini,entropy
max_depth	random integer value (2-10), None
min_samples_split	integer value (1-50,step: 10)
min_samples_leaf	integer value (1-50, step: 10)
min_weight_fraction_leaf	random integer value(1-200), None
max_features	auto, sqrt, log2
max_leaf_nodes	random integer value(2-150), None
min_impurity_decrease	float value(0.0-0.9, step: 20)
bootstrap	True, False
class_weight	balanced, balanced_subsample, None

TABLE II: RF Hyper-parameters (RandomizedSearchCV)

c) K-Nearest Neighbor Classifier: All available parameters (Table III) were again tried and tested in the random search. After repeating the procedure for RandomizedSearchCV and GridSearchCV the best possible parameter value set is used on the model.

Hyper-parameter	Value Range
n_neighbors	integer value (1-200, step:1)
weights	uniform, distance
algorithm	auto, ball_tree, kd_tree, brute
leaf_size	integer value (1-100,step: 10)
p	integer value (1-10, step: 1)
metric	euclidean, manhattan, minkowski, seuclidean

TABLE III: KNN Hyper-parameters (RandomizedSearchCV)

d) Model selection: Comparing the performance of all the three classifier models (Table IV), a tuned Random forest classifier returns the highest performance and hence this model will be used for validation.

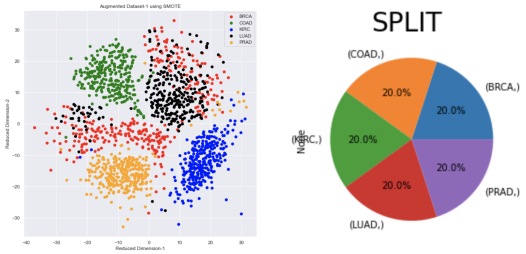
e) Validation: Validation of the trained model is done using K-Fold and LOOCV (Table X).

5) Data Augmentation: Although the dataset is not drastically imbalanced, to challenge the trained model dataset is augmented using SMOTE and Borderline-SMOTE. SMOTE

Classifier Model	f1-score	False Negatives	Computation Cost
Decision Tree	0.9586	7	Less Expensive
Random Forest	0.9947	1	Expensive
K-Nearest Neighbors	0.9947	1	Less Expensive

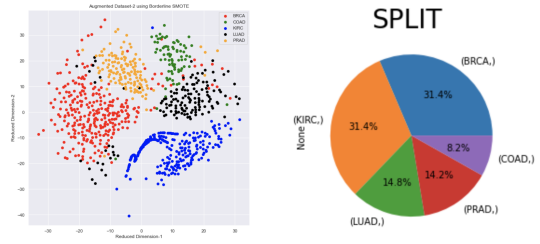
TABLE IV: Model selection Comparison

basically levels the dataset with an equal number of datasets available for each class label. Impleteming SMOTE the size of the dataset size was increased from 801 samples to 1500 samples (Figure 5) with 300 samples for each of the 5 class labels. Using Borderline SMOTE, the size of the dataset was also increased from 801 samples to 955 samples (Figure 5).



(a) SMOTE (b) Class Composition

Fig. 5: Augmented Dataset-1



(a) Borderline SMOTE (b) Class Composition

Fig. 6: Augmented Dataset-2

B. Pipeline 2: Gene Clustering

1) Feature Selection: The genome data used for this project has around 20500 features. Although having additional features is beneficial for a machine learning model, having too many of them can negatively impact machine learning algorithm performance. So we make use of feature selection techniques to reduce the number of features. The methods used for this experiment are listed as below

- PCA
- Kernel-PCA
- ANOVA (Analysis of variance) to select the K-best features

a) Kernel-PCA: As seen earlier, PCA is one of the most reliable methods for reducing dimensions. But a problem with PCA is that it is a linear method. In principle, PCA should be applied to datasets which are linearly separable but most of the real world datasets we encounter in daily life contain elements of non-linearity. We still use PCA on non-linear datasets which may not result in optimal dimensionality reduction. So a bit of inspiration is borrowed from the SVM's and combined with the concept of PCA which gives us the Kernel-PCA [9]. In this method a kernel function is used to project non linear data into higher dimensional feature space to make it linearly separable.

b) K-means clustering: K-Means clustering algorithm [3] is an unsupervised learning algorithms where the goal of the algorithm is group the dataset are into k number of pre-defined non-overlapping clusters or subgroups. The algorithm first chooses K points from the dataset and assigns them as the centroids. Then the algorithm proceeds to update the centroids and their clusters to equilibrium while minimizing the total variance within the clusters. The algorithm should be preferably used on data sets which contain real valued examples because it relies on the Euclidean distance to discover cluster centroids.

2) Evaluation:

a) Adjusted mutual information score: Adjusted Mutual Information (AMI) is an adjustment of the Mutual Information (MI) score. We consider the AMI score over MI score to account for the fact that MI is generally higher for two datasets with a larger number of clusters, regardless of whether there is actually more information shared.

b) Silhouette Score: Silhouette score or silhouette coefficient is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1. It uses compactness of individual clusters (intra cluster distance) and separation amongst clusters (inter cluster distance) to measure an overall representative score.

3) Basic framework for Pipeline component selection:

- Preprocessing-Robust Scaler
- Feature selection - PCA,Kernel-PCA,ANOVA
- Clustering algorithms - K-means clustering
- Evaluation - Adjusted mutual information score,Fowlkes-Mallows Score

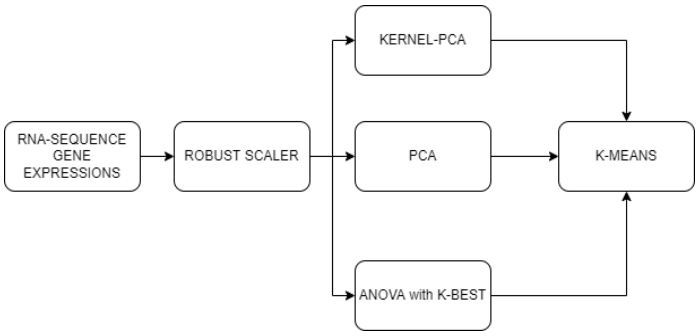


Fig. 7: Pipeline-2 Framework

4) Clustering: First the Gene Expression data is loaded and after some inspection, it is found that there are about 268 zero column vectors. We make sure that we remove them as they add nothing to the dataset. The data is then scaled

using the Robust Scaler. This is done because the algorithms used later are similarity or distance based. Features with higher magnitudes will dominate the algorithm, So by scaling all the features we end up giving equal weight to all features. Once we obtain the scaled data we then use the above mentioned feature selection methods to reduce the dimensions.

a) Experiment-1: Principal component analysis is applied to the scaled dataset and we try to retain up to 95% of total variance of the data. This can be done by assigning the value 0.95 to the attribute n_components which belongs to the PCA function of the sklearn library. Around 500 principal components are obtained.

The K-means clustering algorithm is then applied on the dimensionality reduced dataset. The method is applied in an iterative manner with 5 different values of K . In each iteration, a different value of K is selected and the respective cluster labels are obtained. They are then compared to the ground truth labels and the AMI score is calculated. Silhouette score is also computed with the help of the generated labels. The parameters for this experiment are given in Table V

The whole process is then repeated for 90% and 85% of total variance of the data and the results are noted.

No.of features after feature extraction	95% (500), 90% (350), 85% (250)
K	3, 4, 5, 6, 7, 8

TABLE V: Parameters for Experiment-1.

b) Experiment-2: The above stated process is repeated but this time instead of using PCA as the dimensionality reduction technique we make use of the more powerful Kernel-PCA. Equal number of components are used in both PCA and Kernel-PCA to compare both the methods in a later section. We also make use of the RBF kernel, which is one of the most widely used kernels due to its similarity to the Gaussian distribution.The parameters for this experiment are given in Table VI.

No.of features after feature extraction	500, 350, 250
K	3, 4, 5, 6, 7, 8
Kernel	Gaussian

TABLE VI: Parameters for Experiment-2.

c) Experiment-3: ANOVA F-score is implemented by the Select K-Best function. The function is used to select the features with best variance, two parameters are passed to the function one is the scoring metric that is f_classif (ANOVA F-score) and other is the number of features which will be similar to the above mentioned methods. The same iterative clustering method is carried out and the results are recorded. The parameters for this experiment are given in Table VII

d) Experiment-4: The K-means clustering algorithm is applied on data without any feature extraction and this model is considered as the baseline model. K-means clustering with

No.of features after feature extraction	500, 350, 250
K	3, 4, 5, 6, 7, 8
Scoring metric	f_classif

TABLE VII: Parameters for Experiment-3.

the same values used throughout the other experiments is implemented on the scaled data and the results are recorded.The parameters for this experiment are given in Table VIII

K	3, 4, 5, 6, 7, 8

TABLE VIII: Parameters for Experiment-4.

C. Pipeline 3: Image Classification

In this section a pipeline for image classification is presented. The pipeline consists of well known and established methods and algorithms. The dataset used during construction and testing of this pipeline is the Big Cats dataset. This dataset consists of colored images from five different big cat breeds.

1) Data analysis: The Big Cats dataset consists of 170 images, both RGB and gray scale, displaying various big cats of five different breeds. The breeds contained in the dataset are:

- Cheetah
- Jaguar
- Leopard
- Tiger
- Lion

Figure 8 shows two samples from the dataset. The images within the dataset are of different shapes and sizes, some are high quality, and some are low quality, some are oriented vertically, and others horizontally. Upon inspection of the dataset it was noticed that some classes contained samples from other classes. For instance, the folder containing jaguar samples also contained a few cheetah samples. After removing these samples we end up with a reduced total of 163 samples. The distribution of image samples is displayed in Figure 9. As can be seen from this figure, all classes have between 28 and 40 samples.

(a) Cheetah (b) Jaguar

Fig. 8: Two samples from the dataset

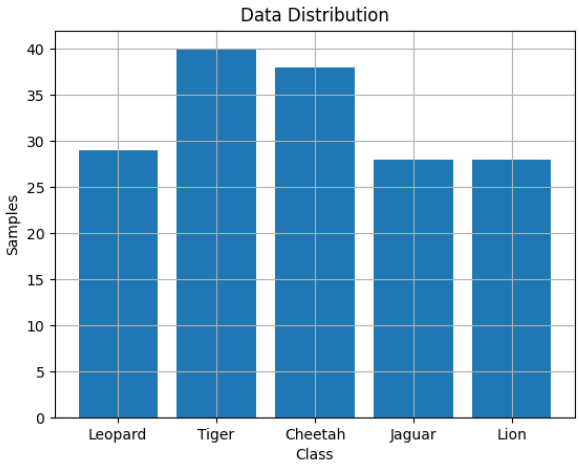


Fig. 9: The data distribution of the big cats dataset

2) Feature selection: In this subsection we will discuss the preprocessing steps within the pipeline. Pre-Processing consist of the following steps:

1) Load the raw data
2) Convert all images into grey-scale
3) Extract SIFT features
4) Select subset of SIFT features

In step 1 the raw data is loaded into a list. Each entry in the list contains a tuple of the form [(image, label)]. In step 2 all images within the dataset are converted to gray-scale images, this is done because the feature extraction algorithm used in step 3 (SIFT) operates on gray-scale images. In step 3 the Scale Invariant Feature Transform (SIFT) is used to extract useful features from the images [4]. The big advantage of SIFT is that the SIFT feature descriptor is invariant to uniform scaling, orientation, illumination changes, and partially invariant to affine distortion. This means that SIFT is able to find the same keypoint in different images, even if they are taken from a different angle or under different illumination. SIFT returns so-called 'keypoints' and 'descriptors'. An example of such key points can be seen in Figure 10. The key points represent maxima or minima pixel values in a given set of neighboring pixels. Specifically, the key points are chosen by taking the minima or maxima values of a given pixel when comparing it to its 8 neighbors (3×3 blocks), as well as when comparing it across different levels of image scaling. These pixels being compared are the outcome of two images with different scaling (different σ values in a Gaussian blurring) are subtracted. Each keypoint has a 'descriptor' which describes the keypoint. These descriptors are the features that will be used in our pipeline. Once they have been created all images within the dataset are replaced with their corresponding sift features. In step 4 of pre-processing, the number of sift features per image is reduced. This is done because some images will have over 5000 sift descriptors, resulting in very high dimensional data. One can assume that not all sift features need to be retained, since some describe features in the background, as can be seen in Figure 10. Furthermore, the clustering algorithms used in later steps of the pipeline will converge much faster if only a subset of the SIFT features is used. Therefore the descriptor list of every image is narrowed down by randomly selecting 245 descriptors of each particular image. The reason that it is 245, is because the image with the lowest amount of descriptors was found to have 245 SIFT descriptors. Preliminary testing showed that this method of randomly selecting a subset of the descriptors had a very small negative effect on the accuracy of the model, but resulted in a noticeable reduction of computation time. At the end of pre-processing the raw image data has been transformed into a list containing 163 tuples of the form (SIFT descriptors, 245, label).

Fig. 10: SIFT keypoints and their orientation

3) Clustering: In this section we will discuss the clustering methods used for dimension reduction. After extracting the SIFT features from the images, the dataset has 163 entries, in which each entry has a list of 245 feature descriptors, which are all of size 128. This is still fairly high dimensional data, and can therefore be reduced. Each image will be reduced to a so-called 'Bag of (Visual) Words' (BOW). The general idea behind the BOW model is to represent an image as a set of visual words (features). So in the end, each image becomes a histogram which counts the occurrences of each visual word within this image. These visual words are computed by taking the mean feature descriptor from several similar descriptors, i.e. grouping several features into one 'visual word'. This grouping is computed using a clustering algorithm. A visual example of the Bag of Words can be seen in Figure 11. Note that this is only a simplified example, these are not actual visual words. The clustering algorithms used to obtain the visual words are:

- Mini Batch K Means
- Birch

Note, only one algorithm is used to produce the bag of words. The reason two different algorithms have been implemented is merely for testing purposes, which aim to

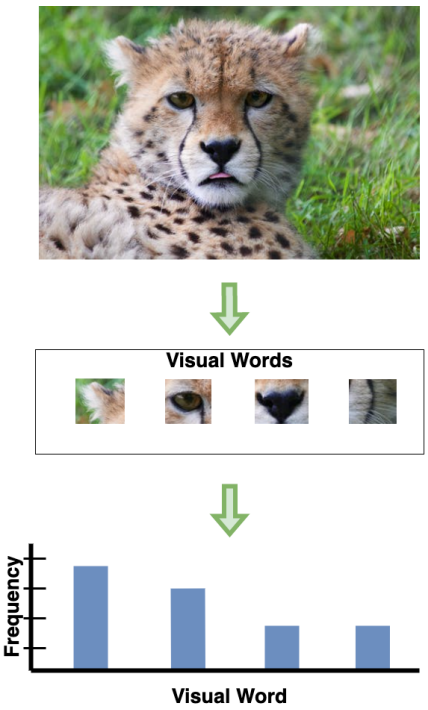


Fig. 11: Visual Word Example

show how different clustering algorithms might influence the performance of the classification pipeline.

K Means is the standard clustering method used for creating the BOW model. In order to further reduce the computation time, Mini Batch K Means Clustering is used. Mini Batch K Means Clustering is a variant of the KMeans algorithm which uses mini-batches to reduce the computation time, while still attempting to optimize the same objective function. These mini-batches are simply small subsets of the input data, which are randomly sampled during each training iteration. This method helps the algorithm to converge to a local solution in less time. K Means aims to cluster the data by separating samples into n groups of equal variance, by minimizing the inertia according to formula 1.

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \tag{1}$$

A great feature of K Means is that it will always converge (given enough time). The speed of convergence is highly dependent on the initial positions of the centroids.

As stated above, the second algorithm that was implemented is Birch [13]. This algorithm build a so called Clustering Feature Tree (CFT). The data is compressed set of Clustering Feature (CF) nodes. These nodes have subclusters can have CF nodes and children. This algorithm is very memory efficient since these subclusters are able to hold necessary information for clustering, thereby preventing the need to hold the entire input data in memory. Furthermore, Birch has a very competitive computation speed. These two factors, memory efficiency and computation speed, are why Birch has been selected for this pipeline.

4) Classification: In this section the classification algorithms used for image classification will be discussed. After dimension reduction by clustering, every image had been reduced to a histogram (bag of words) with 50 bins. The dataset is now a list of 163 tuples of the form (hist, label). The data has now been sufficiently reduced, and classification can therefore commence. The classifiers that have been implemented are:

- Nu-Support Vector Classification
- K Nearest Neighbors
- Quadratic Discriminant Analysis

The goal for all these classifiers is to train them in such a way that they take as input a histogram (bag of words) and output the correct class label. This should be within the capabilities of all three classifiers.

Nu-Support Vector Classification (NuSVC) [10] is a variant of the Support Vector Machine (SVM). A SVM has the capability to classify high dimensional data by constructing a hyper-plane or set of hyper-planes. A good class separation is achieved when the hyper plane has the largest functional margin, that is, the largest distance to the closest training data point of any class. This large functional margin leads to a lower generalization error. The parameter ν (Nu) in the NuSVM Algorithm controls the margin errors and the amount of support vectors. Which is usually done by the parameter C (regularization term) in most other variants of SVM.

K Nearest Neighbor Classification (KNN) is a classification algorithm which classifies a data sample s_i by looking at the k samples which are closest in distance to s_i. This distance can be any metric measure, such as standard Euclidean Distance. Once the k closest samples have been determined, the algorithm looks for the most prominent class c_p among these k samples. Once the most prominent class has been determined, it assigns this class c_p to the sample s_i.

Quadratic Discriminant Analysis Classification works by modeling the class conditional distribution of the data $P(X|y = k)$ for every class k. Through the use of Bayes Rule we can obtain predictions:

$$P(y|X) = P(X|y) * P(y)/P(X) = P(X|y) \times P(Y)/(\sum_{y'} P(X|y') \times p(y')) \tag{2}$$

$P(X|y)$ is modelled as a Gaussian distribution. There are no assumptions regarding the covariance matrices of the Gaussian, which results in a quadratic decision surface.

5) Pipeline Overview: In order to present a clear picture of the presented pipeline, a final overview is presented:

- Load raw image data.
- Turn RGB images into grayscale images.
- Extract 245 SIFT features per image.
- Use a clustering algorithm to create a Bag of Visual Words model.

- Use a classifier to classify the data.

6) Data Augmentation: As mentioned in Section III-C2, SIFT is invariant to uniform scaling, orientation, illumination changes, partially invariant to affine distortion, and also uses grayscale images. Therefore, it was decided to study the effect of adding noise to the data samples. Next to a pipeline which was trained on the original data, a second pipeline was trained using the original data plus 20 percent more augmented samples. The samples were augmented using 'Gaussian', 'Poisson' and 'Salt & Pepper' noise. For every augmented sample, the specific type of noise was randomly chosen from these three options.

7) Hyper Parameters: For the image classification pipeline only two hyper-parameters have been tested in a parameter sweep, which is the number of clusters in k-means clustering and the number of neighbors in K-nearest neighbors. The remaining hyper parameters (which are many) have been set to the default values. For more information, we refer to scikit-learn [7], the library used for all classification and clustering algorithms in the image classification pipeline.

This concludes the methodologies used to implement the image classification pipeline.

IV. Results

A. Pipeline 1: Gene Expression Classification

a) Feature Selection: As seen in Table IX, MI based feature selection outperformed PCA based. A closer look at the baseline model shows that MI based DTC has far fewer leaves - 9 against 26, resulting in a shorter depth of the tree as well, indicating Information gain providing ample information with fewer features than component based variance is able to on the same data set.

Feature Selection	Criterion	Mean Accuracy
Mutual Information	mi_score >0.5	0.9752
-Principal Component Analysis	variance : 95%	0.8882

TABLE IX: Feature selection Comparison

b) Classification: DTC performed fairly well with a f1-score of 0.9689, but with many false negatives (Figure 12) it cannot be considered the best model. Whereas, RF classifier returned a f1-score of - 0.9947, with only one false negative (Figure 12) which can be attributed to how the some of the *"PRAD"* class features were scattered all over the feature space. KNN model achieves a f1-score of - 0.9947, very similar to the RF model. Again, there was only one false negative (Figure 13) and this is again relatable between *"PRAD"* and *"BRCA"*. Also, comparisons of tuned models with base models with no tuning show a bigger positive increase only for DTC and only marginal increases in RF (default $n\_estimators$:1000) and KNN (default $n\_neighbors$:5).

c) Validation: As observed in Table X, model performance based on f1-scores stay relatively similar across different folds as well as during LOOCV compared to the performance without any validation.

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	57	0	0	2	1
KIRC	0	15	0	1	0
LUAD	0	0	30	0	0
PRAD	2	0	0	26	0
COAD	1	0	0	0	26

(a) DTC

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	60	0	0	0	0
KIRC	0	16	0	0	0
LUAD	0	0	30	0	0
PRAD	1	0	0	27	0
COAD	0	0	0	0	27

(b) RF

Fig. 12: Confusion Matrix - trained models

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	60	0	0	0	0
KIRC	0	16	0	0	0
LUAD	0	0	30	0	0
PRAD	1	0	0	27	0
COAD	0	0	0	0	27

(a) KNN

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	60	0	0	0	0
KIRC	0	16	0	0	0
LUAD	0	0	30	0	0
PRAD	2	0	0	26	0
COAD	0	0	0	0	27

(b) Selected Model

Fig. 13: Confusion Matrix - trained models

d) Augmented Dataset: Augmented Dataset-1 produced a f1-score of 1.0, which is a perfect model as the entire dataset was balanced. The result was expected and the model performed accordingly. However, Augmented Dataset-2 produced a fractionally lower f1-score of 0.9903 compared to 0.9947 on the original dataset.

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	60	0	0	0	0
KIRC	0	60	0	0	0
LUAD	0	0	60	0	0
PRAD	0	0	0	60	0
COAD	0	0	0	0	60

(a) KNN

	BRCA	KIRC	LUAD	PRAD	COAD
BRCA	60	0	0	0	0
KIRC	0	16	0	0	0
LUAD	0	0	60	0	0
PRAD	0	1	0	27	0
COAD	0	0	0	0	27

(b) Selected Model

Fig. 14: Confusion Matrix - Performance on Augmented Dataset

e) Ensemble: DTC, RF cand KNN classifiers combined to a f1-score of 0.9950. False negative still existed as the majority voting for that false negative data was true form both RF and KNN models.

f) Tuned model on Original Dataset: When the tuned RF model is worked on the original dataset with no reduced features, the model returns the same maximum f1-score obtained with feature reductions - 0.9947. With one false negative being detected in "PRAD" class(Figure 13).

B. Pipeline 2: Gene Expression Clustering

After performing the experiments mentioned in III, 4 models are obtained. Evaluating these models based on the AMI score, It was noted that the highest score was obtained when PCA was applied as the feature extraction technique and the number of clusters(K) was set to 5. Kernel PCA which was supposed to perform better than PCA did not show the best results. The base model unusually gave a pretty high AMI Score when the K was assigned to the value 5 but it took a long time and used a lot of computational resources. Kernel PCA was

Cross Validation	Criterion	Scores
No CV applied	n/a	0.9947
K-Fold CV	fold - 3	0.998
	fold - 5	0.995
	fold - 7	0.992
LOOCV	n/a	0.995

TABLE X: Cross Validation on Trained Model

the only feature extraction method which didn't produce the highest AMI score when K was set to 5. This can be seen in the figure 16.

The experiments are then evaluated on a second metric to check if there is consistency. Silhouette score is chosen as the second metric because it doesn't need the ground truth labels and helps us get an idea of how precisely the samples are assigned to the cluster. The highest silhouette scores for the base model was obtained when the value of K was assigned the value of 5. Performance of K-means on the dataset with PCA as feature selection yielded a result similar to the base model and did not stand out when compared to the other models. Surprisingly, the best silhouette score is obtained when features are selected based on the ANOVA F-score and K was set to 5. This is illustrated in figure 17 and figure 18.

Important results of all the 4 experiments can be seen in Table XI, Table XII and Table XIII.

No. of features after FS	K	AMI Score	Silhouette Score
500	5	0.83762	0.1647
	6	0.90793	0.16526
	7	0.87187	0.16371
350	5	0.9795	0.17631
	6	0.90885	0.17693
	7	0.86511	0.17462
250	5	0.9795	0.18937
	6	0.9069	0.18922
	7	0.87505	0.18923

TABLE XI: Results of K-means applied on the PCA dataset

No. of features after FS	K	AMI Score	Silhouette Score
500	5	0.81359	0.09868
	6	0.82947	0.10004
	7	0.78576	0.10482
350	5	0.81122	0.11745
	6	0.81883	0.12231
	7	0.79466	0.1271
250	5	0.81359	0.13683
	6	0.81812	0.14333
	7	0.80763	0.14731

TABLE XII: Results of K-means applied on the K-PCA dataset

No. of features after FS	K	AMI Score	Silhouette Score
500	5	0.9508	0.52948
	6	0.92166	0.4821
	7	0.89	0.44573
350	5	0.95281	0.54106
	6	0.92381	0.49517
	7	0.90483	0.43767
250	5	0.95281	0.57644
	6	0.92848	0.53424
	7	0.90102	0.47076

TABLE XIII: Results of K-means applied on the ANOVA F-SCORE dataset

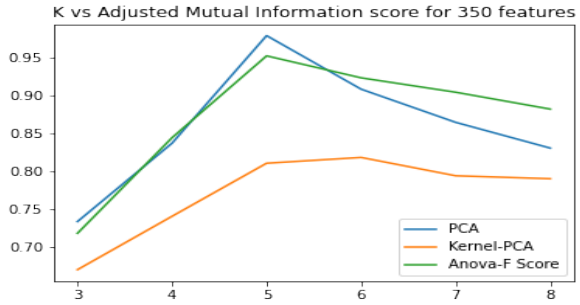


Fig. 15: No. of clusters vs Adjusted mutual info score for 350 features

C. Pipeline 3: Image Classification

In this section the results for the image classification pipeline will be discussed. All combination between the classifiers $CLF \in \{NuSVC, LDA, KNN\}$, clustering algorithms $CLS \in \{KMeans, Birch\}$ and datasets $D \in \{Normal, Augmented\}$ have been tested by way of 10-fold cross validation. The results of 10-fold cross validation for the normal and augmented dataset can be found in Table XV and XVI respectively. As stated in Section III-C7, a parameter sweep has been done for the number of clusters in K-Means, and the number of neighbors in KNN. The results of this parameter sweep can be found in Table XIV. The best performing hyper parameters have been selected for the final tests. From tables XV and XVI one can see that NuSVC in combination with K means clustering performs the best, scoring 58.5% accuracy when trained on the normal dataset, and 60.5% when trained on the augmented dataset. The lowest score in Table XV goes to LDA in combination with Birch clustering, scoring 40.8%. The lowest score in Table XVI goes to KNN in combination with Birch clustering, scoring 43.5%.

Overall we see that all accuracies except one are higher in Table XVI than in Table XV.

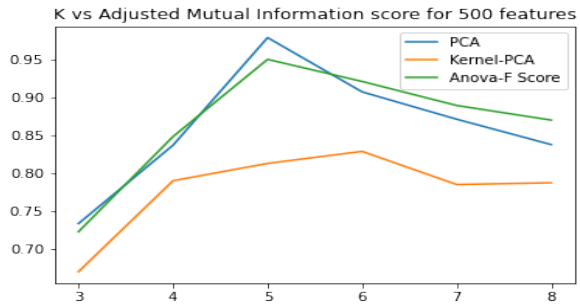


Fig. 16: No. of clusters vs Adjusted mutual info score for 500 features

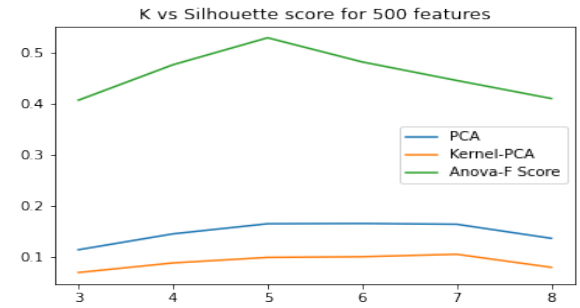


Fig. 17: No. of clusters vs Silhouette score for 500 features

V. Discussion

A. Pipeline 1: Gene Expression Classification

From obtained results, it is clear to select the best possible components for the Pipeline. Mutual Information can provide computationally quicker and better reduced features than PCA for high-dimensional numerical data. For classification, DTC performs weaker compared to Random Forest and KNN classifiers along with multiple false negatives. Also, one other observation was the performance of RF and KNN models, they both return almost identical performances.

Hyper-parameter tuning for KNN is computationally less expensive compared to RF and hence could be used as an alternative, but for this pipeline fully tuned RF classifiers will be selected. Apart from DTC there were no noticeable changes in improvement of accuracy for RF and DTC when not tuned, this could imply this numerical dataset might not actually require hyper-parameter tuning.

Performance of the tuned model on original dataset turned out surprisingly to be equal to that of post feature reduction. This explains the models complexity levels being simple. Also, with 0.99% scores the designed model might not require Ensemble for boosting, but to remove the one existing false negative ensemble was given a shot. Here the three models (DTC, RF & KNN). Results show only a marginally improved model accuracy. False negative was not solved but the increased accuracy shows a good signs for the slightly better model to be used. Also, performance of tuned RF

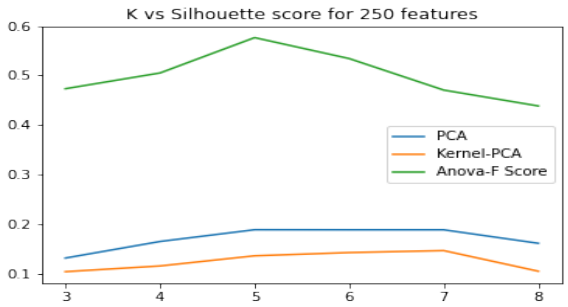


Fig. 18: No. of clusters vs Silhouette score for 250 features

Hyper-parameter Tests							
KMEANS				KNN			
Clusters	50	100	200	**Neighors:**	5	10	20
Avg:	0.535	0.508	0.508	Avg:	0.465	0.533	0.439

TABLE XIV: Hyper parameter sweep (obtained via 10-fold cross validation)

classifier worked with almost similar accuracy's on original and augmented datasets.

B. Pipeline 2: Gene Expression Clustering

From the results it can be concluded that there are 5 major clusters present in the Gene Expression Data. The metrics that were considered to evaluate the model did not produce the same results which would have been ideal. Both the metrics produced the best results when the number of clusters was set to 5. Adjusted Mutual Information score favored K-means clustering on the PCA reduced dataset. This might be due to the fact that the mutual information score is largely dependent on the similarity between two clusters. The ANOVA F-score selected features got a decent mutual information score and outperformed all the other feature selection methods by a margin when the models were evaluated by taking silhouette score into the account. So if a model had to be chosen to cluster the given data, K-means clustering with ANOVA F-score feature selection should be preferred.

C. Pipeline 3: Image Classification

After careful inspection of the results, a few things can be concluded. Firstly, NuSVM in combination with K Means clustering scores the highest accuracy, and is therefore the best combination from all that have been tested. Furthermore, data augmentation (adding noise) seems to have a small positive influence on the performance of the system. It is quite interesting that a combination of simple Gaussian, Poisson and S&P noise produces better results. It is not entirely clear if the higher accuracy is due to the fact that there are simply more training samples (even if they are just a noisy version of already existing ones), or if it is actually the noise that improves the accuracy. A deeper look into this might be an interesting topic for future research.

Another interesting aspect of this system are the SIFT features, and the bag of words model. Bag of Words is designed

Fold	Accuracy					
	K Means			Birch		
	NuSVC	LDA	KNN	NuSVC	LDA	KNN
1	0.563	0.375	0.563	0.438	0.438	0.378
2	0.625	0.687	0.750	0.563	0.563	0.563
3	0.563	0.313	0.625	0.438	0.375	0.688
4	0.813	0.563	0.563	0.750	0.688	0.625
5	0.563	0.375	0.500	0.563	0.438	0.438
6	0.750	0.562	0.563	0.500	0.318	0.500
7	0.312	0.438	0.250	0.188	0.250	0.250
8	0.438	0.438	0.375	0.375	0.250	0.375
9	0.688	0.375	0.500	0.500	0.430	0.438
10	0.533	0.467	0.334	0.734	0.334	0.534
Avg.	0.585	0.459	0.502	0.492	0.408	0.478

TABLE XV: Results of 10 fold cross validation for the Big Cats dataset

Fold	Accuracy					
	K Means			Birch		
	NuSVC	LDA	KNN	NuSVC	LDA	KNN
1	0.8	0.6	0.7	0.6	0.55	0.55
2	0.45	0.35	0.3	0.45	0.3	0.4
3	0.55	0.5	0.45	0.4	0.65	0.35
4	0.5	0.5	0.6	0.55	0.6	0.45
5	0.8	0.6	0.75	0.7	0.65	0.5
6	0.632	0.526	0.421	0.632	0.579	0.474
7	0.579	0.368	0.579	0.737	0.684	0.421
8	0.474	0.421	0.421	0.579	0.316	0.526
9	0.579	0.421	0.526	0.474	0.632	0.368
10	0.684	0.474	0.579	0.368	0.263	0.316
Avg.	0.605	0.476	0.533	0.522	0.549	0.435

TABLE XVI: Results of 10 fold cross validation for the augmented Big Cats dataset

to make optimal use of the information provided by the sift features. However, due memory and computational cost, it was not feasible to pass all SIFT features to the BOW Model. As mentioned in Section III-C2, only 245 sift descriptors per image were passed to the clustering algorithm which constructs the BOW model. An interesting topic for future research would be a look into a smart selection algorithm, which selects the N best SIFT descriptors. This would most likely improve the accuracy of the system.

Author Contributions

All work has been divided equally. Thijs and Niels worked on the image data, Sai and Sam worked on the numerical data.

References

[1] S. Babichev, V. Lytvynenko, and V. Osypenko. Implementation of the objective clustering inductive technology based on dbscan clustering algorithm. In *2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, volume 1, pages 479–484. IEEE, 2017.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[4] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[5] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, 256:56–62, 2017.

[6] H. Lu, L. Yang, K. Yan, Y. Xue, and Z. Gao. A cost-sensitive rotation forest algorithm for gene expression data classification. *Neurocomputing*, 228:270–276, 2017.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] M. J. Rani and D. Devaraj. Two-stage hybrid gene selection using mutual information and genetic algorithm for cancer data classification. *Journal of medical systems*, 43(8):1–11, 2019.

[9] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[10] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *582-588*, volume 12, 01 1999.

[11] A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. C. Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 2006.

[12] Y. Wu and A. Zhang. Feature selection for classifying high-dimensional numerical data. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[13] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: A new data clustering algorithm and its applications. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114. ACM, 1996.