Exploring the Supervised Learning Models to Automatically Diagnose Colon Cancer Patients based on their SNP Profiles

Ali Reza Ibrahimzada, Huzeyfe Ayaz, and Ali Cakmak

August 19, 2019

The goal of this analysis is to explore the machine learning-based automatic diagnosis of colorectal patients based on the single nucleotide polymorphisms (SNP). Such a computational approach may be used complementary to other diagnosis tools, such as, biopsy, CT scan, and MRI. Moreover, it may be used as a low-cost screening for colorectal cancers to improve public health.

Dataset: The dataset includes SNPs observed in particular genomic loci that are located within DNA regions of 11 selected genes, namely, p16540, p16580, mdm2, GAL3, TIM1, trail, pd1duz, pdl1poly, CD28, cd27snp, and CD40. The dataset included 50 healthy individuals (control group) and 65 colon cancer patients. The dataset also includes additional information for patients only, such as, the age of the patient, the stage of the cancer (in terms of two different staging models, namely, T-model and n-model), the location of the tumor, peripheral invasiveness of the tumor, as well as tissue-differentiation around tumor regions. However, since these extra information is only available for cancer patients, we have not utilized them for the classification task that we focus in this study.

Methods: We employ several supervised classification algorithms, namely, Logistic Regression (LR), Random Forests (RF), and Support Vector Machine (SVM). We write scripts in Python programming language using the Scikit-learn library. We initially perform an exploratory analysis of the underlying data to determine the features for the classification models.

The following table shows the statistics for the ages of healthy individuals and cancer patients. Since the mean ages values of colon cancer patients (mean age = 60) are considerably older than the control group (mean age = 35), we remove the age information from consideration to avoid the bias that the classifiers would learn such that older people would be mostly predicted as “colon cancer”, while younger people would be predicted as “healthy”. Such a bias would cause the classification models to overfit the current dataset, may prevent creating a generalized model that would work for unseen future patients.

Age Information	n	missing	unique	mean	.05	.10	.25	.50	.75	.90	.95
All subjects	115	16	48	49.3	26.9	28	35	47.0	66.0	74.2	79
colon cancer	65	9	32	59.0	35.0	38	52	61.5	69.3	79	80
control	50	7	20	35.4	26.1	27	29	35.0	37.0	41	44.8

We next explored the SNP data for individual genes. The SNP data for some genes were not complete, that is, some of the individuals do not have any information on their DNA sequence at the studied loci. What is more, the individuals with missing SNP data mostly belonged to a particular category. For instance, the following table shows that for the trail gene, 53 individuals had no SNP information, and the majority (i.e., 39) of these individuals with no SNP data belong to the control group.

trail genotype frequencies (n=115)						
	All subjects		colon cancer		control	
Genotype	Count	Proportion	Count	Proportion	Count	Proportion
C/C	35	0.56	27	0.53	8	0.73
C/T	27	0.44	24	0.47	3	0.27
NA	53	---	14	---	39	---

Similarly, as shown in the following table, for CD28 gene, 37 individuals do not have any SNP information, and almost all (i.e., 36) of these patients with no CD28 SNP data belongs to the cancer group.

CD28 genotype frequencies (n=115)						
	All subjects		colon cancer		control	
Genotype	Count	Proportion	Count	Proportion	Count	Proportion
C/C	17	0.22	2	0.07	15	0.31
T/C	13	0.17	9	0.31	4	0.08
T/T	48	0.62	18	0.62	30	0.61
NA	37	---	36	---	1	---

Likewise, four other genes, namely, PD1DUZ, PDL1POLY, CD27, and CD40 have unbalanced missing data between control and cancer group.

Baseline Approach: We omitted SNP data for genes that many individuals do not have SNP genotype to avoid the unrealistic bias that individuals with no SNP information would be automatically predicted as “healthy” or “colon cancer” depending on the majority class that has most of the missing rows. To sum up, in the baseline approach, 6 genes, namely, Trail, CD28, PD1DUZ, PDL1POLY, CD27, and CD40 are omitted. Hence, the classification models are built on the following remaining 5 genes: MDM2, GAL3, TIM1, p16540, and p16580.

The SNP data is considered as categorical data. However, machine learning models work with number data. Thus, in baseline approach, we employ one-hot encoding transformation to convert the categorical SNP data into numbers. More specifically, one-hot encoding introduces a separate column into the data for each distinct value under each column. For instance, CD40 gene contains the following distinct genotypes: C/C, C/T, and T/T. One-hot encoding would create three columns for CD40 gene, one for each distinct genotype. Then, those individuals having a particular genotype will have value 1 for the corresponding column, and 0 for the remaining other genotype columns for CD40.

Completing the Missing Data with Data Imputation

The baseline approach may utilize less than half of the genes, and the partial information that is available for other genes are completely discarded. In order to address this problem, in the next set of

alternative approaches, we first employ data imputation to predict the missing values for above 6 omitted genes. Then, we include them in the classification models as well. In particular, we employ two imputation techniques, namely, KNN-based Imputation and Multivariate Imputation by Chained Equations (MISC).

Approach A (One-hot Encoding + Data Imputation): In this approach, we firstly applied the one-hot encoding via get_dummies method of Pandas module on our stock data-set. This method helps to convert missing values to 0 for each feature automatically and we got the binary type data for each value under our original values. Since we have no missing value after get_dummies method, we should find missing values in sparse data and convert them to NaN values to apply Imputation. K-nn imputation works based on using the nearest value to estimate missing data and it gives us float values for this situation. Now, we have lots of float values, and to obtain the binary representation of one-hot encoding, we converted each value to the nearest integer which 0 or 1.

Approach B (Custom decimal transformation + Data Imputation + One-hot Encoding): In this approach, we modify our dataset to decimal forms instead of using our categorical data. More specifically, we represent each possible 4 letters of DNA alphabet in two bit binary numbers. That is,

A: 00
T: 01
C: 10
G: 11

Since we have pairs of nucleotides in SNP genotypes, we will convert them by just replacing each character with its number form. For instance,

TG: 0111 = 7 (in decimal form)
CC: 1010 = 10 (in decimal form)
AC: 0010 = 2 (in decimal form)
etc.

Hence, for each column, we will be using the corresponding decimal value after this transformation. Then, we perform data imputation after this transformation.

Approach B2 (Data imputation on categorical data + One-hot Encoding): This approach is a slight variation of Approach B in that we directly applied imputation on the original categorical data without any transformation. We employ hamming distance as the distance metric. Then, we applied one-hot encoding.

Approach B3 (Custom decimal transformation + Data Imputation + Conversion to the Nearest Integer + One-hot Encoding): This approach is another slight variation of Approach B in that, after imputation, we converted floating point values that are filled for missing values to the nearest integer. This is because, as before, we observe that the imputation provides unique float values almost for every cell, and this may introduce some bias. The rest is the same as the original approach B.

Approach C (Custom decimal transformation + Data Imputation): For this approach differs from Approach B mainly in the last step. That is, it does not use one-hot encoding. Instead, it employs the decimal values of the genotype data that are obtained as explained above.

Approach C2 (Custom decimal transformation + Data Imputation + Conversion to the Nearest Integer): This approach is a slight variation of Approach C in that, after imputation, we converted floating point values that are filled for missing values to the nearest integer.

Evaluation & Metrics: We perform 10-fold cross validation to evaluate the performance of the employed classifiers. More specifically, we randomly divide the data into 10 parts. Then, we employ 9 parts for training the models, the remaining 1 part to test the trained model. Then, we repeat these steps 10 times, where each time, a new classification model is trained on a different combination of 9 parts, and tested on the remaining 1 part. Then, the average performance from this 10 iterations are reported as the performance of the classifiers. The splitting is done in a stratified manner keeping the same proportions of control and cancer groups in both the training and test set. This is important as the ratio of control group and cancer group is not balanced in the dataset. As the accuracy metric, we use f1-score, which is a commonly employed measure in machine learning tasks. We also generate Receiver Operating Characteristics (ROC) curves, and report the area under the curve (AUC) as an alternative performance metric for the classifiers.

Results: Figure 1 reports the f1-scores for the three employed classifiers for the baseline approach. The best classification performance was obtained with SVM with f1-score around 70%. This performance figure is not perfect, given that we had to omit 6 genes out of 11, and the number of individuals in the dataset is very small to build generic classification models.

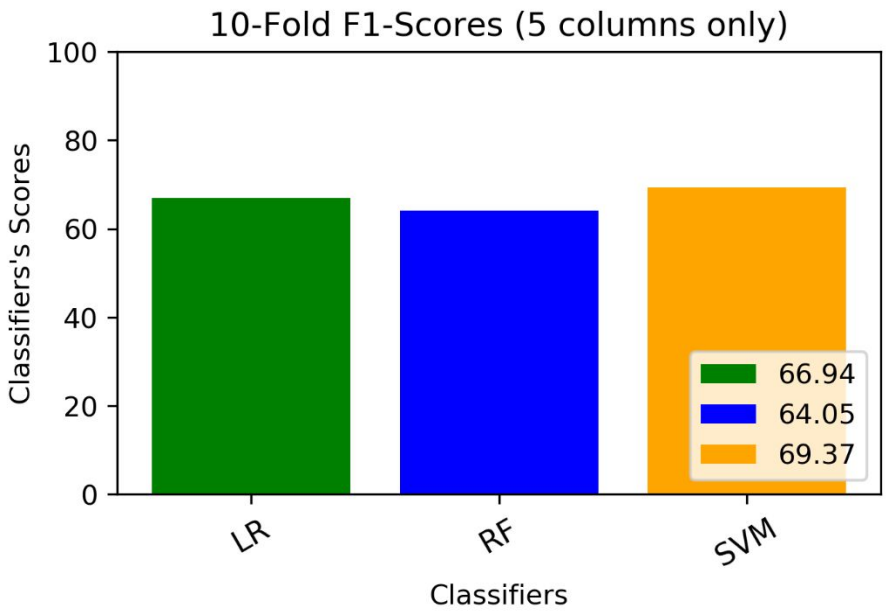

Figure 1: f1-scores for the employed classifiers with the baseline approach

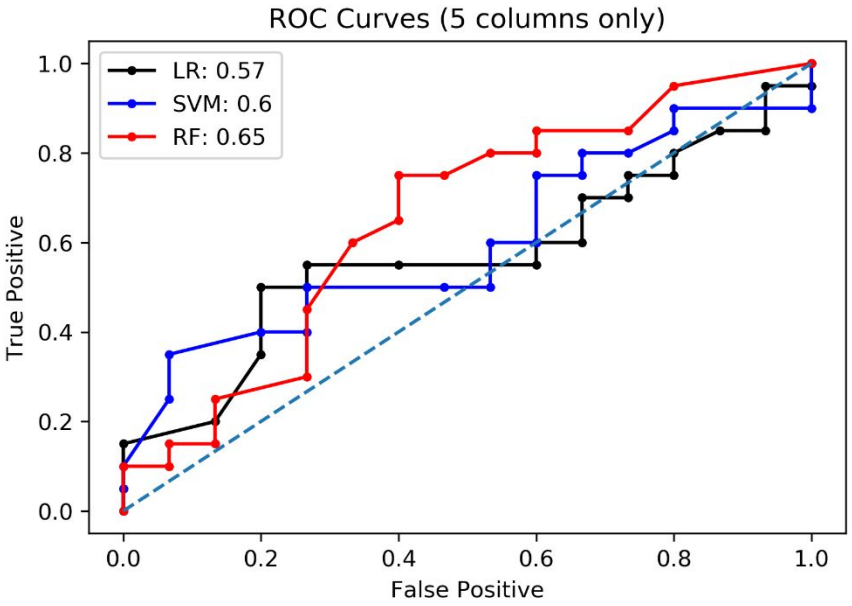

Figure 2: AUC scores for the employed classifiers for the baseline approach

Figure 2 reports the AUC score for the same classifiers as an alternative performance metric. In this category, RF provides the best performance. To further explore this result, we chart the decision boundary graphs for the classifiers to see the way they separate cancer patients from healthy individuals, and where they fail to do so. Figure 3 presents the decision boundaries for the three classifiers that we study in this report. We observe that RF has higher true positive rate and lower false

negative rate than SVM, and this is what AUC metric measures. Therefore, RF outperforms SVM under this metric.

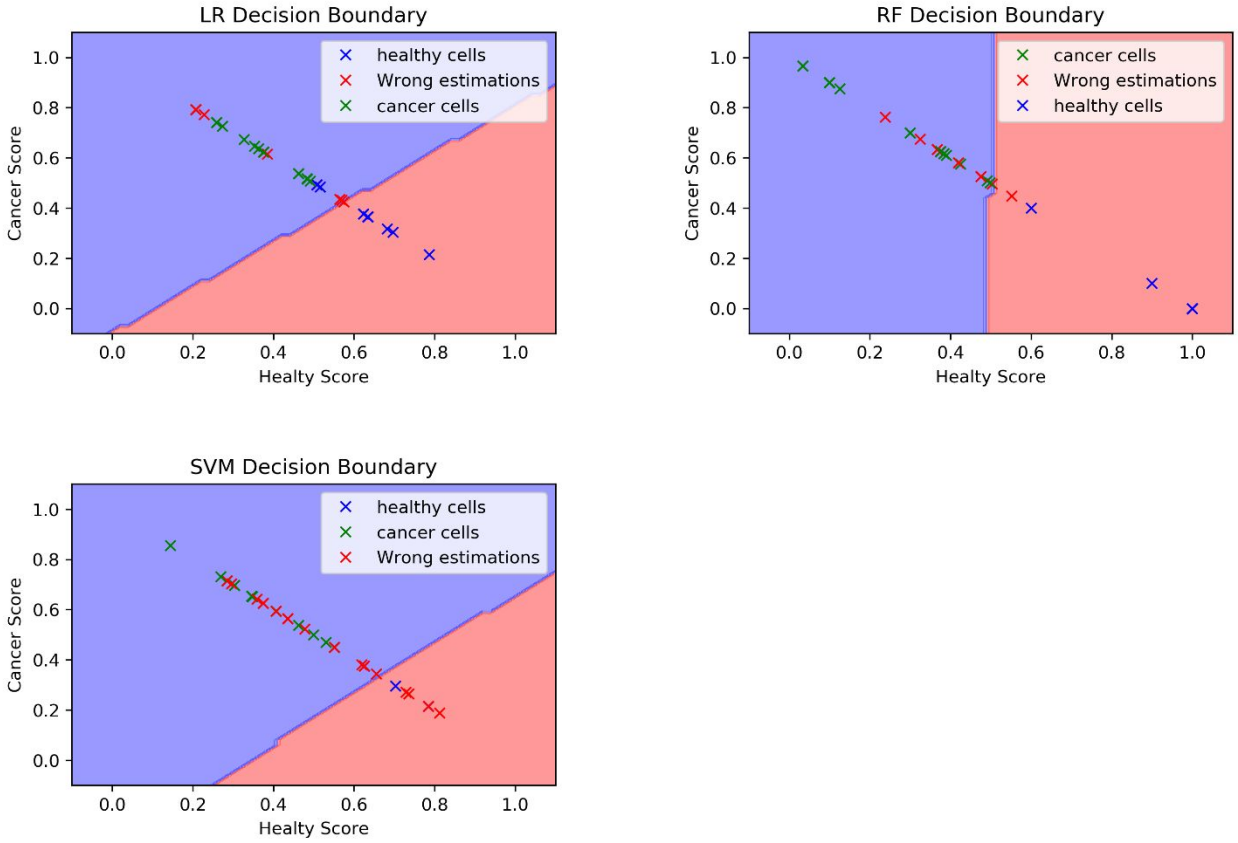

Figure 3: Decision boundary graphs for the classifiers with the baseline approach

We next study genes and their corresponding genotypes that are most discriminating between the control group and the cancer group with the baseline approach. Figure 4 charts the most discriminating factors between these two groups in sorted order (most discriminating is first). To compute the feature importances, we employed Random Forest's feature importance values. We observe that top-5 discriminating genotypes are G/C in TIM1, G/C in p16540, C/C in GAL3, C/C in p16540, and C/A in GAL3. In other words, under baseline approach, top 5 discriminating genotypes belong to 3 genes (i.e., TIM1, p16540, and GAL3) out of 5. And, the top-10 list includes genotypes from all 5 genes included in the baseline approach. This shows that the colon cancer is associated with multiple genes in a complex interactions.

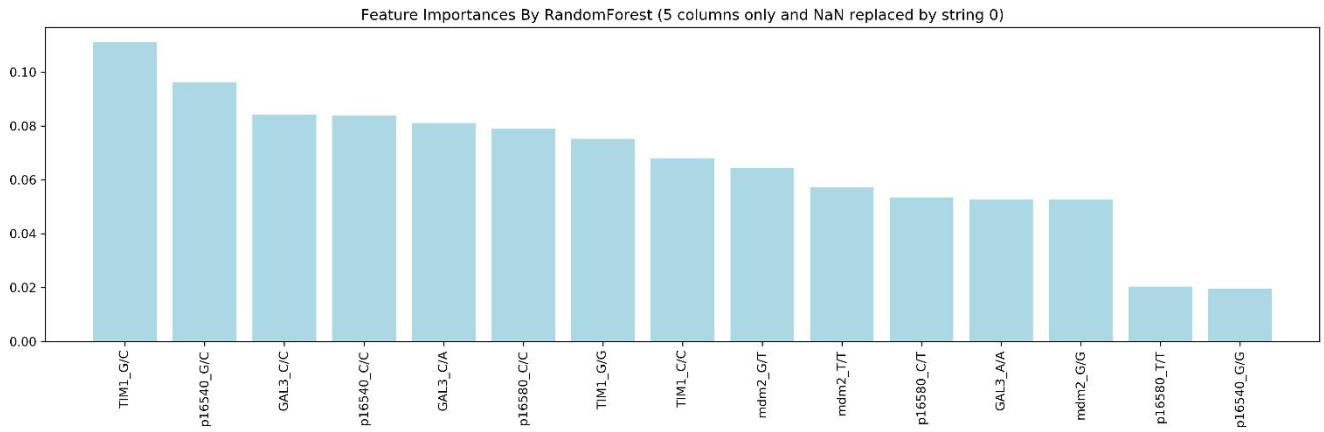

Figure 4: Most discriminating genotypes between control group and cancer group

Next, we evaluate the approaches that employ data imputation and include all the genes in the analysis. We consider two different imputation approaches, KNN-based imputation and Multivariate Imputation by Chained Equations (MISC). First, we evaluate these two approaches to determine the imputation method to be used in the following approaches. Figure 5 compares the f1-scores of these two imputation methods with 5-fold cross-validation within the setting of Approach C.

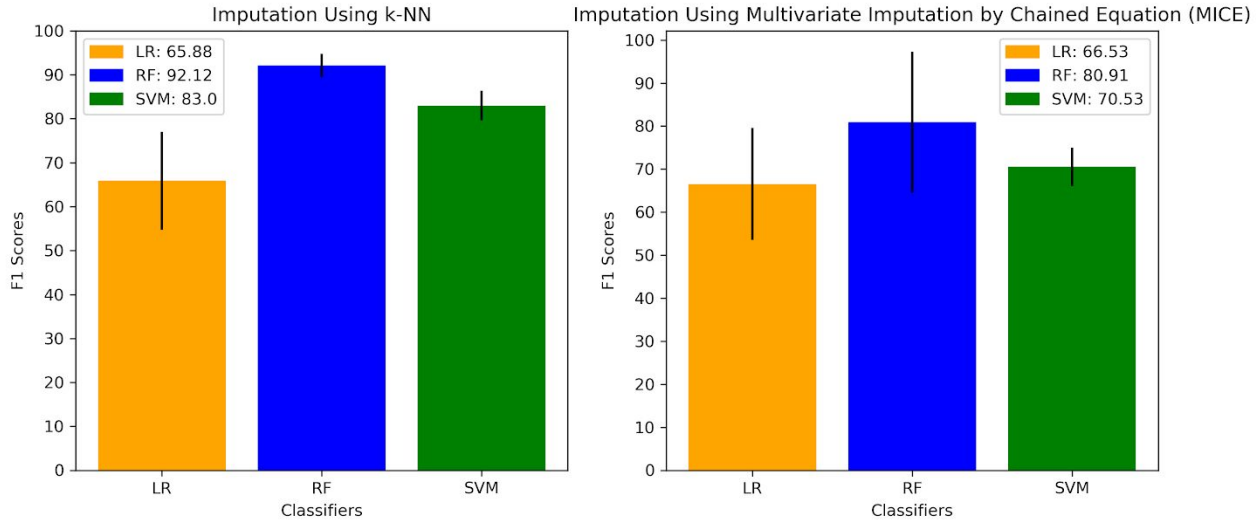

Figure 5: F1 scores of different imputation techniques

As we can see from figure 5, KNN-based imputation provides better estimates than MICE algorithm in this case. Besides, we also compared at the standard deviations of different classifiers under these two imputation methods between different folds in k-fold cross validation. We observe that with K-nn based Imputation, standard deviations are much smaller than that with the MISC algorithm. That means, KNN-based imputation provides more consistent results.

For the rest of the experiments, we employ KNN-based imputation as our default imputation method. We next compare the performance of approaches A and B.

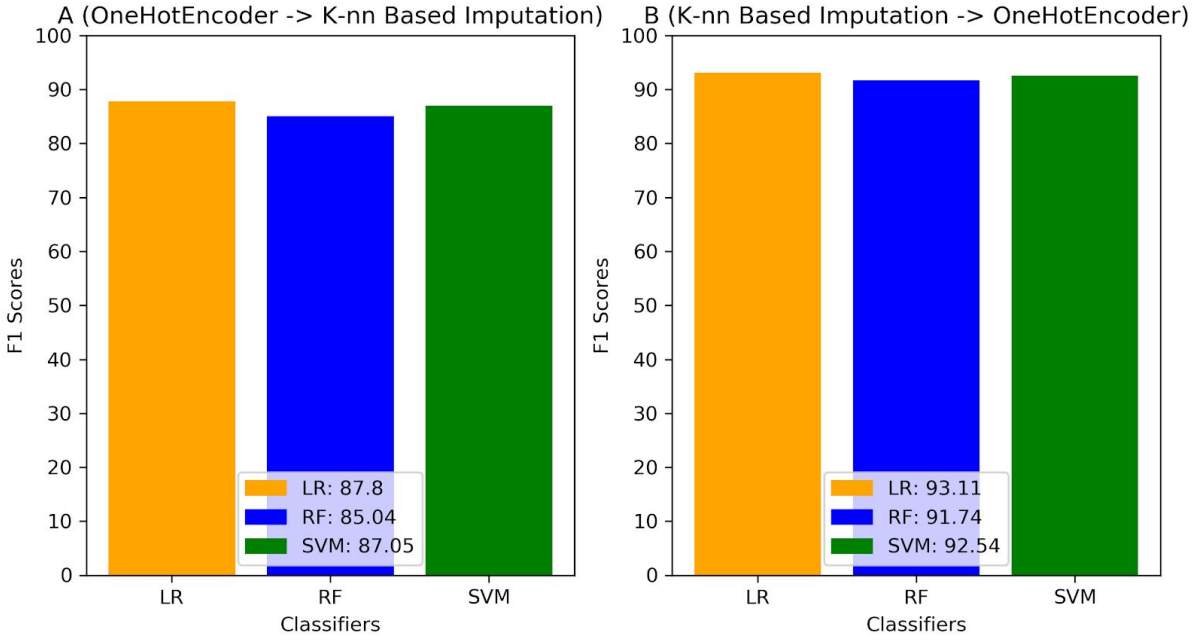

Figure 6: f1-scores of option A and B

Figure 6 shows that approach B provides better performance than approach A. We further investigated why Approach B performs better. We observed that data imputation fills in distinct values for the missing values for different individuals. We hypothesis that it may introduce some kind of bias in favor of healthy (i.e., control) or cancer group depending on which group has the highest number of missing values. To test this hypothesis, we employ slight variations of approach B. Approaches B2 and B3 remove the "uniquely-filled missing values" bias in two different ways, and provides us the opportunity to test the effect of this bias. Figure 7 shows that the f-scores for both approaches B2 and B3 are much lower than the original approach B. These results confirm the bias introduced by uniquely-filled values. Therefore, we ignore approach B, and consider approaches B2 and B3 for comparison purposes.

We next evaluate approach C. The left part of Fig. 8 shows the f1-scores for approach C. The results show that approach C outperforms approach A. However, based on our experience with the effect of "uniquely-filled missing values" bias in approach B, we further evaluated the effect of this bias within the context of approach C. Approach C2 removes the "uniquely-filled missing values" bias by converting the filled missing values to the nearest integer. The right part of Fig. 8 shows that after removing the effect of "uniquely-filled missing values" bias, the performance of Approach C decreases significantly. Hence, for comparison purposes, we ignore Approach C. Instead, we consider approach C2.

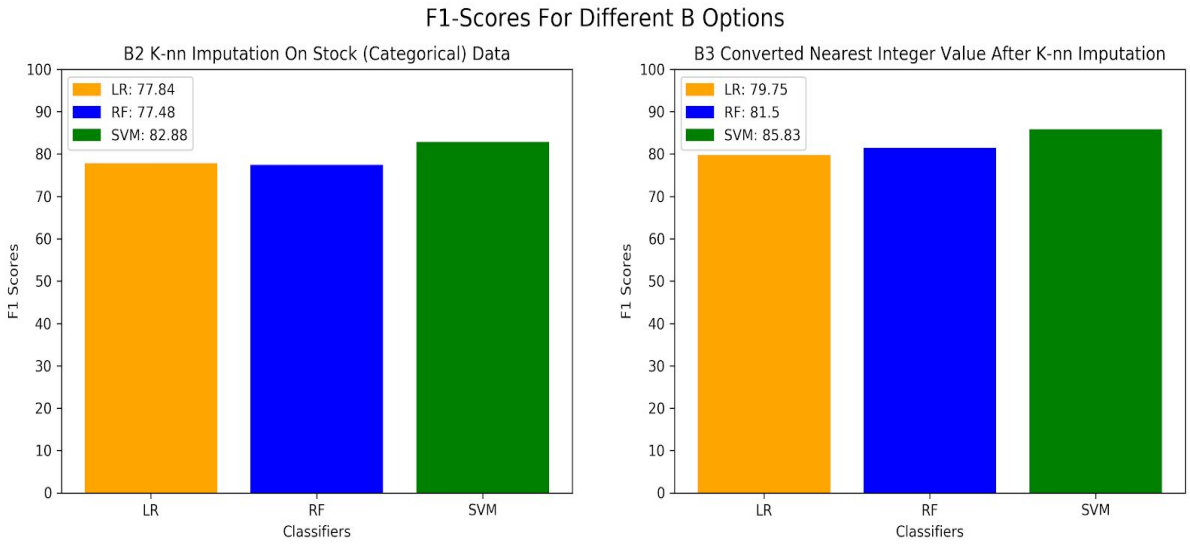

Figure 7: F1 scores for approaches B2 and B3

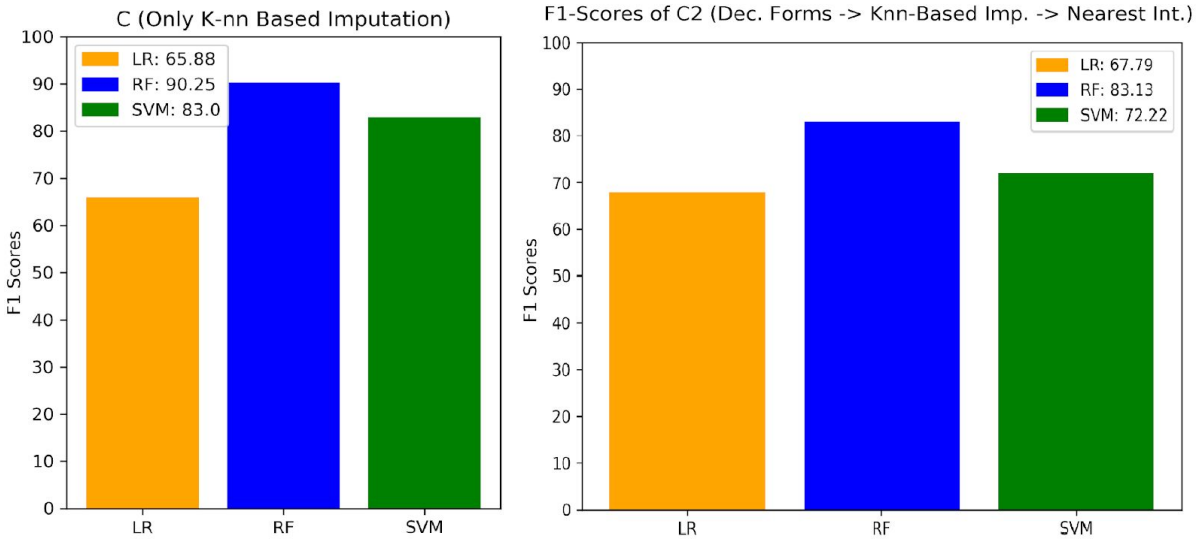

Figure 8: F1 scores for approaches C and C2

In conclusion, among all the evaluated approaches, approach A seems to provide the best f1-scores after eliminating the "uniquely-filled missing values" bias. In particular, Logistic Regression and SVM performs best with f1-score of around 88%.

We next further perform in-depth analysis using approach A.

Exploratory analysis of Approach A:

Figure 9 plots the ROC curve for the employed classifiers. Logistic Regression together with Random Forest provides the best AUC value. Combining Figure 9 with Figure 6, Logistic Regression seems to provide the best combined f1-score and AUC values.

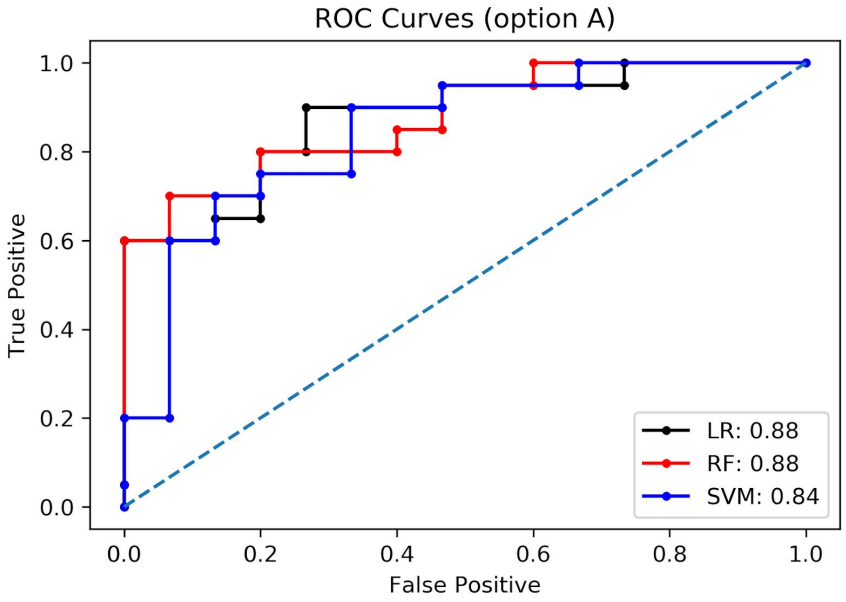

Figure 9: ROC Curves of Approach A

Next, we investigate the most important features that differentiate between healthy and colon cancer subjects. To this end, we first chart Random Forest Classifier's feature importance values. Figure 10 lists the features in their decreasing order of importance. According to this list, C/T and C/C genotypes of CD40, C/C genotype of CD28, C/T genotype of pd1duz, C/T and C/C genotypes of trail gene constitutes the most important feature set.

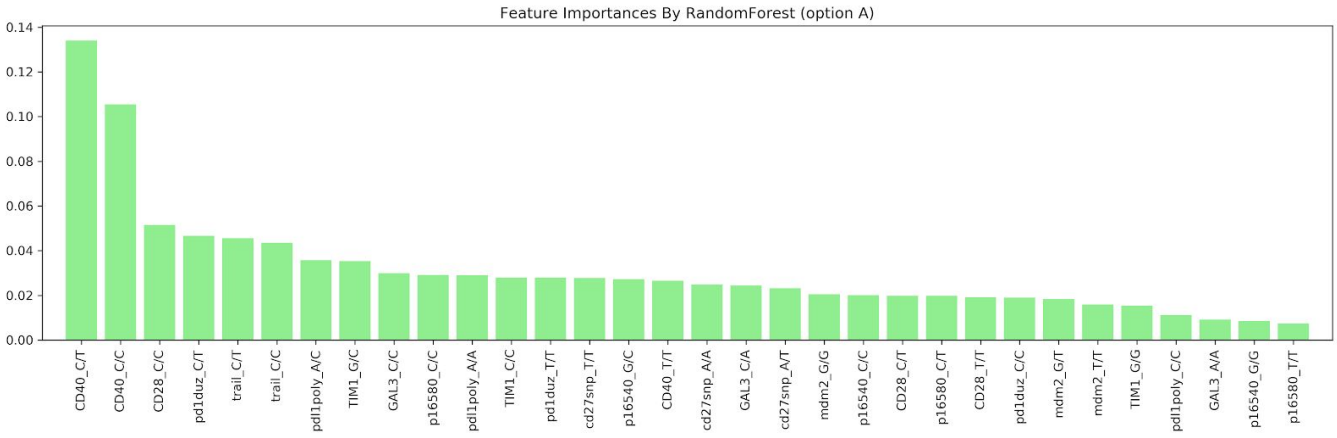

Figure 10: Feature importance for Approach A

F and P Scores (option A:
knn-imputation after one-hot encoding)

	f-values	p-values
CD40_C/C	37.50834	1.36e-08
CD28_C/C	18.61001	3.44981e-05
CD40_C/T	14.33445	0.0002467735
pd1duz_C/T	10.51366	0.0015578771
trail_C/T	7.28887	0.0080047526
trail_C/C	7.28887	0.0080047526
CD28_T/T	6.78186	0.0104460442
cd27snp_T/T	6.15964	0.0145409241
pdl1poly_A/C	5.70264	0.0185977448
pdl1poly_A/A	5.64805	0.0191561912
GAL3_C/C	4.85052	0.0296682817
p16580_C/C	3.5873	0.0607794608
p16580_T/T	2.37728	0.1259076342
p16540_G/C	2.2478	0.1365928798
TIM1_G/C	2.17463	0.1430843327
GAL3_A/A	2.13901	0.1463700831
pd1duz_T/T	2.13901	0.1463700831
GAL3_C/A	1.97588	0.1625683733
mdm2_G/T	1.97588	0.1625683733
p16540_C/C	1.85884	0.1754689513
p16580_C/T	1.78639	0.1840512736
mdm2_T/T	0.99796	0.3199385709
CD28_C/T	0.95458	0.3306433891
TIM1_C/C	0.94452	0.3331946167
cd27snp_A/A	0.58247	0.4469345534
pdl1poly_C/C	0.55925	0.4561177819
CD40_T/T	0.55406	0.4582085937
TIM1_G/G	0.55329	0.458518892
mdm2_G/G	0.46456	0.4968964059
p16540_G/G	0.12691	0.7223269386
pd1duz_C/C	0.0404	0.8410522268
cd27snp_A/T	0.00034	0.9853021956

Feature Importance Scores (option A:
knn-imputation after one-hot encoding)

	feature_importance_score
CD40_C/T	0.13637
CD40_C/C	0.10524
CD28_C/C	0.05087
trail_C/C	0.04855
pd1duz_C/T	0.04689
trail_C/T	0.04637
TIM1_G/C	0.03647
pdl1poly_A/C	0.03591
TIM1_C/C	0.02938
GAL3_C/C	0.02851
p16540_G/C	0.02826
pd1duz_T/T	0.02776
pdl1poly_A/A	0.0269
cd27snp_T/T	0.02684
p16580_C/C	0.02626
cd27snp_A/A	0.02568
CD40_T/T	0.02548
GAL3_C/A	0.02456
cd27snp_A/T	0.02346
CD28_C/T	0.02088
mdm2_G/G	0.0207
p16540_C/C	0.02044
p16580_C/T	0.01966
CD28_T/T	0.01821
pd1duz_C/C	0.01768
mdm2_G/T	0.01717
TIM1_G/G	0.0169
mdm2_T/T	0.01563
pdl1poly_C/C	0.01029
p16540_G/G	0.00861
GAL3_A/A	0.00827
p16580_T/T	0.0058

Figure 11: ANOVA analysis of features sorted by F-values and P-values of A

Second, We perform ANOVA analysis and compute statistical significance of features in differentiating healthy subjects and colon cancer patients. Figure 11 provides F-values and p-values of features where the list is sorted by F-values. The statistically significant features (p-val < 0.05) are as follows: C/C and C/T genotypes of both Trail and CD40, C/C and T/T genotypes of CD28, C/T of pd1duz, T/T of CD27, A/C and A/A of pd1poly, and C/C of GAL3.

We next study the decision boundaries of the employed classifiers. Figure 12 charts the decision boundaries of the employed classifiers. SVM and LR has simpler decision boundaries, while RF seems to overfit the data; thus, it seems to be less generalizable.

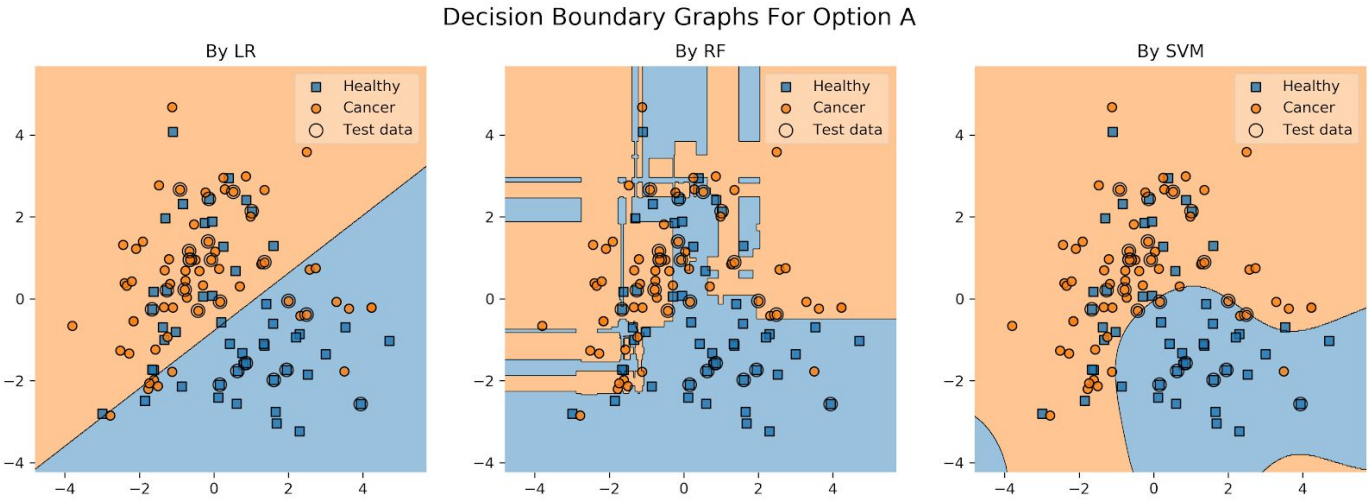

Figure 12: Decision Boundaries of LR, RF and SVM Classifier

Clustering Analysis:

We next study the subgroups within the cancer patient subjects based on their clinical and pathological information. To this end, we employed two kinds of methods: (i) PCA-based clustering analysis, and (ii) Hierarchical clustering analysis.

Figures 13 through 18 chart the PCA clustering analysis on different clinical and pathological features, such as cancer stage, tumor location, etc. In particular, for visualization purposes, we focus on the three most significant principal components that are obtained from the original 11 main features (SNP genotypes of 11 genes) after applying one-hot encoding and knn-based imputation.

In general, PCA-based clustering visualizations in 3-dimensions do not provide clearly separable clinical or pathological subgroups based on genotype information.

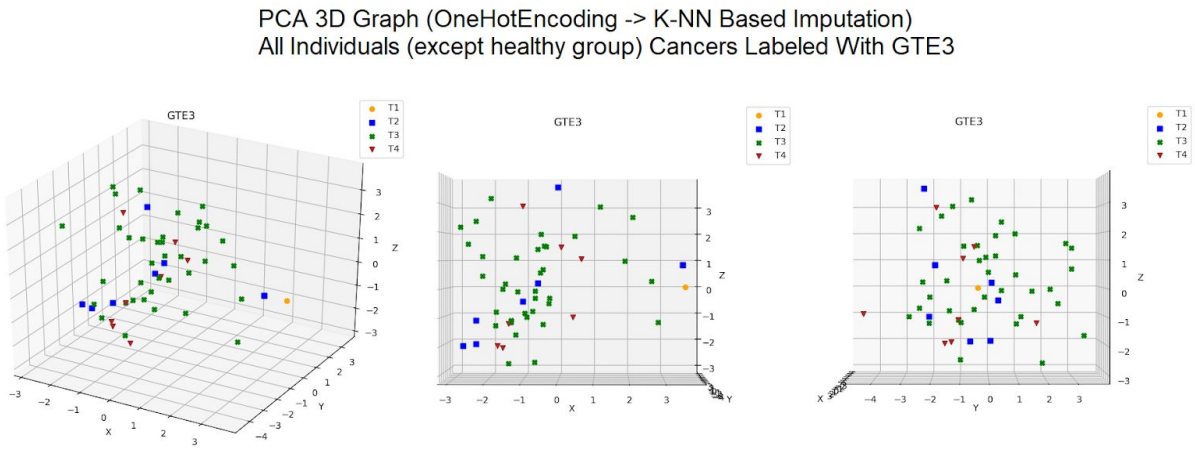

Figure 13: PCA graph of GTE3 column

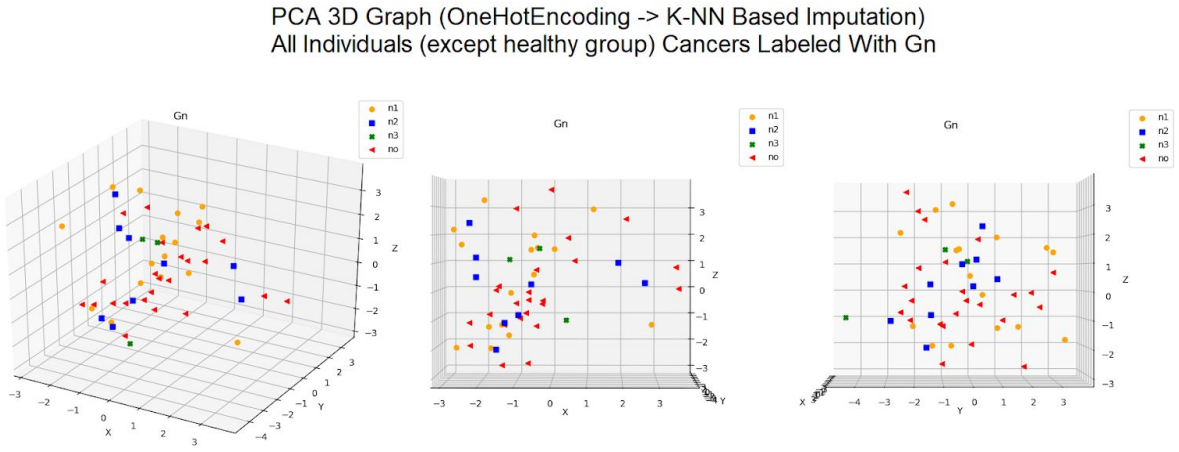

Figure 14: PCA graph of Gn column

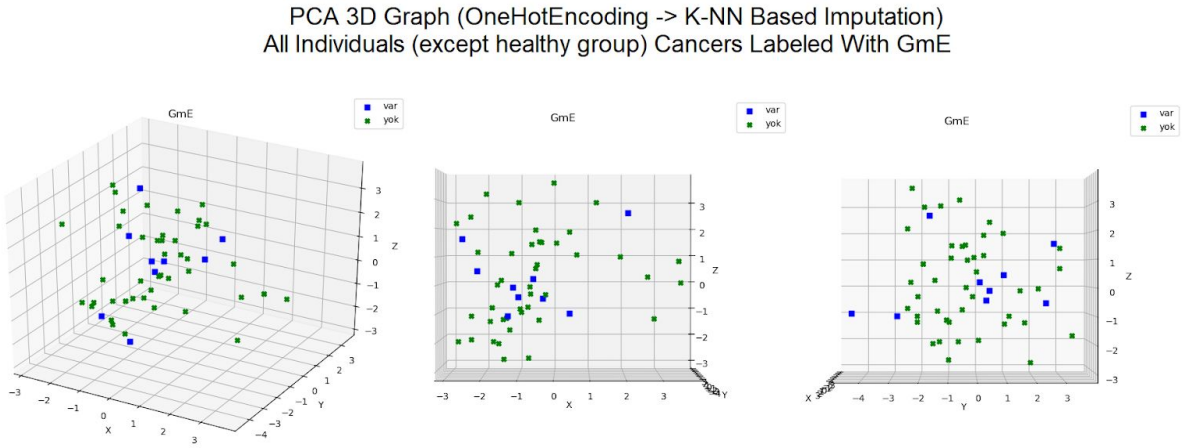

Figure 15: PCA graph of GmE column

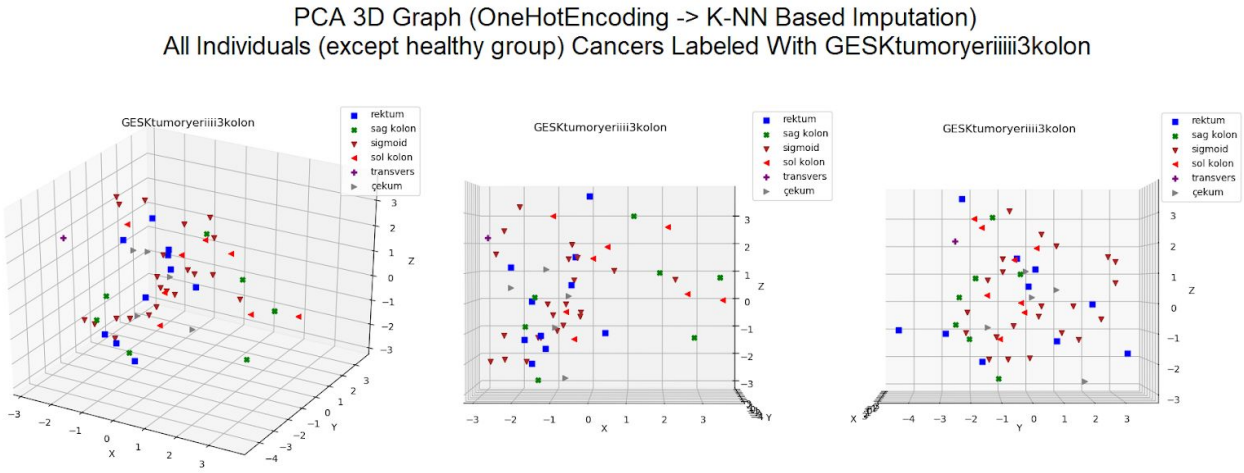

Figure 16: PCA graph of GESKtumoryeriiii3kolon column

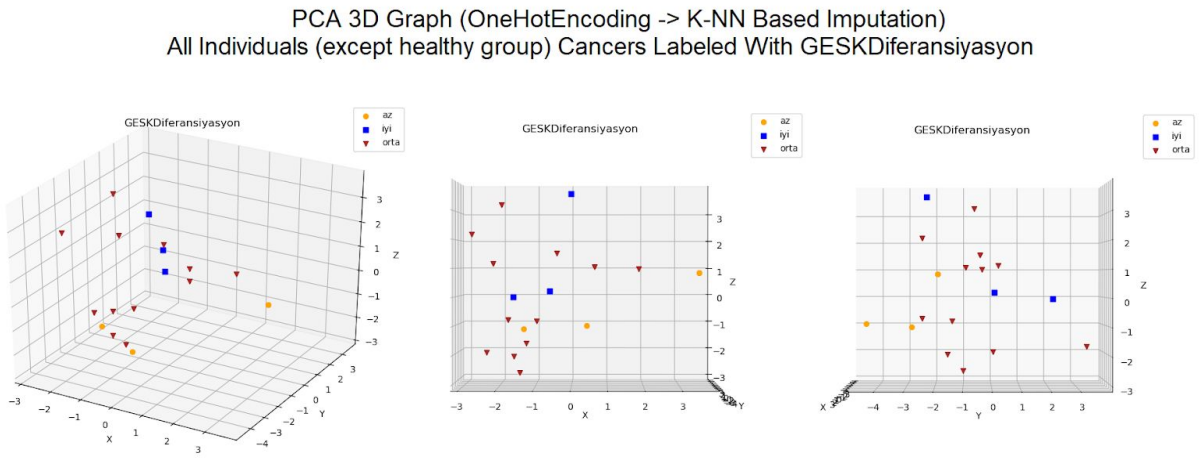

Figure 17: PCA graph of GESKDiferansiyasyon column

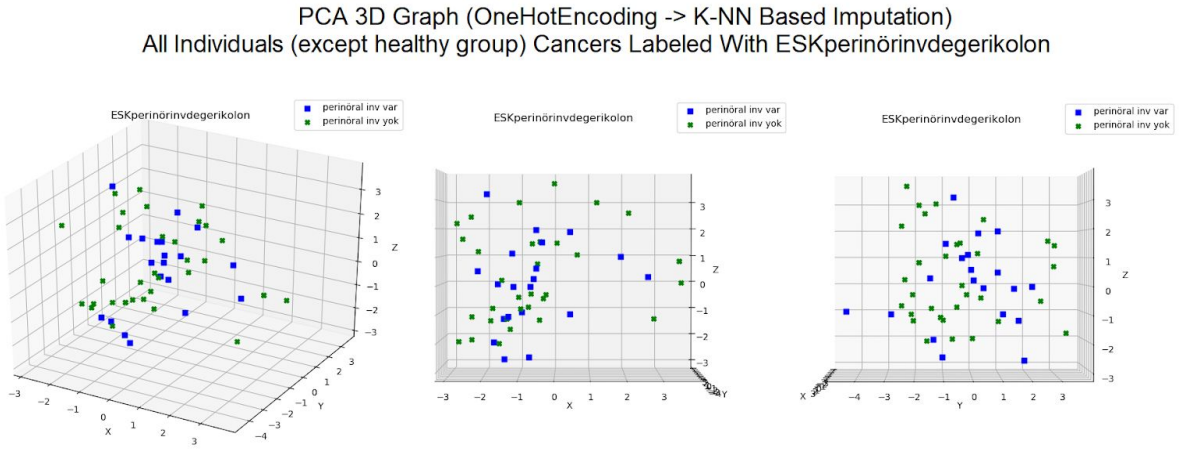

Figure 18: PCA graph of ESKperinörinvdegerikolon column

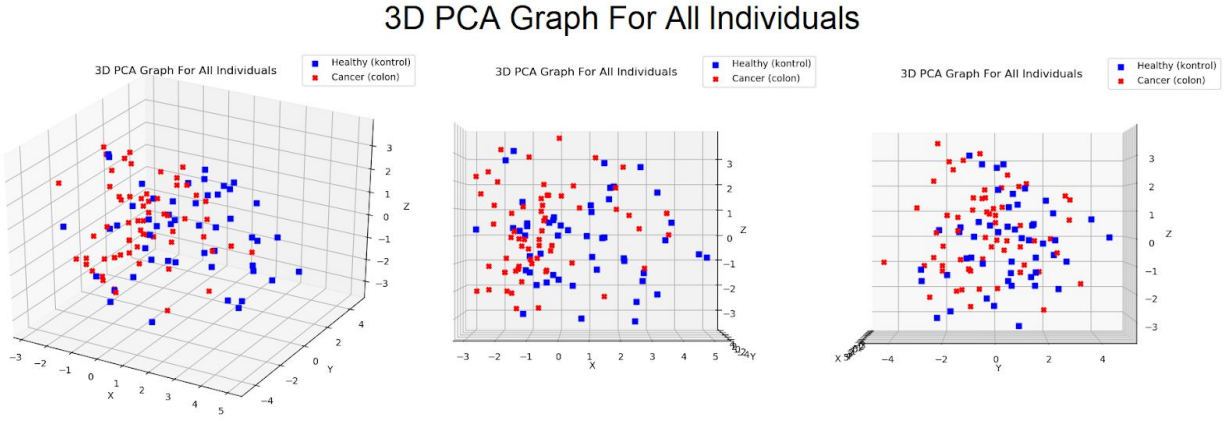

Figure 19: PCA graph of all individuals (healthy and cancer)

We next employ hierarchical clustering to discover subgroups among cancer patients. To this end, we use Euclidean distance as the distance metric. Figures 20 through 25 visualizes the hierarchical clustering results as dendrograms. In each dendrogram, only the labels of the subjects change based on the studied clinical or pathological feature. An immediate observation is that there are 4 distinct clusters/groups marked by the hierarchical clustering results. This may point to 4 different stages of tumors. However, Figures 20 and 21 do not confirm this consideration, as most of the clusters are heterogenous in terms of their T-stage and n-stage labels.

Figure 24 presents somewhat more intuitive clustering results in terms of GESK Differentiation. Neighboring clusters are also immediate preceding or following values in terms of the qualitative differentiation labels.

In all other clustering graphs, there is no visually clear separation in terms of the studied clinical or pathological features.

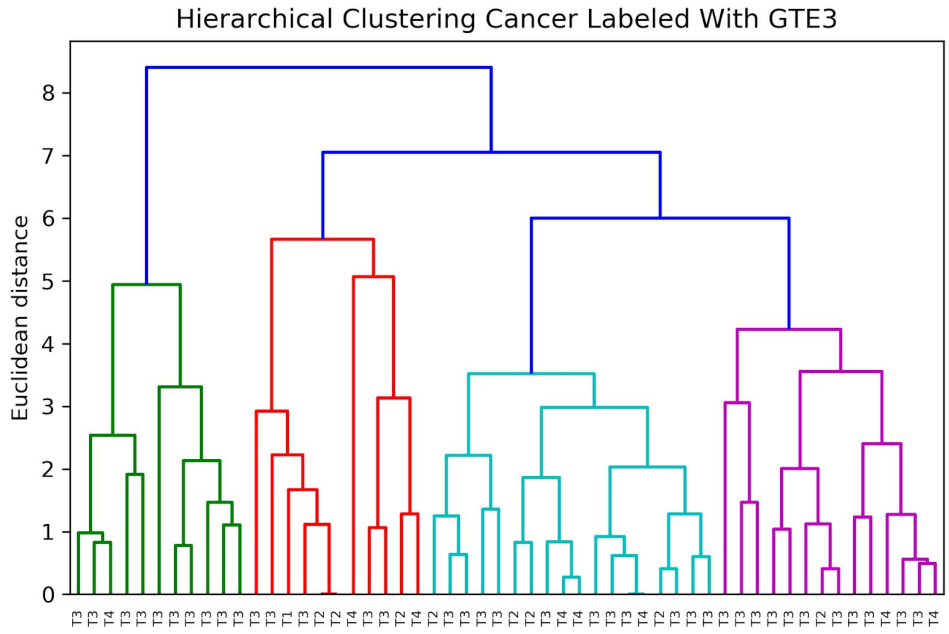

Figure 20: Hierarchical Clustering Dendrogram of GTE3

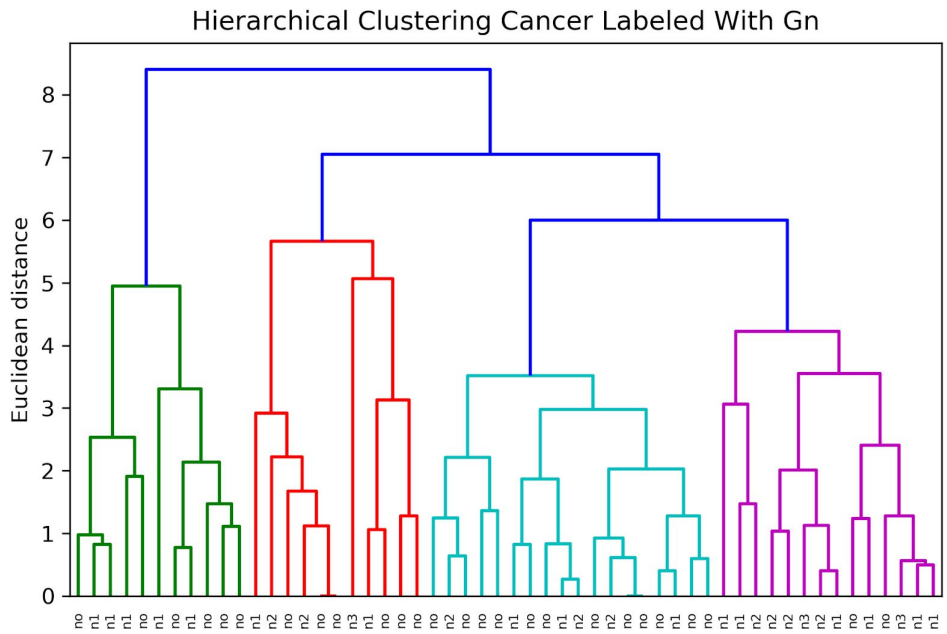

Figure 21: Hierarchical Clustering Dendrogram of Gn

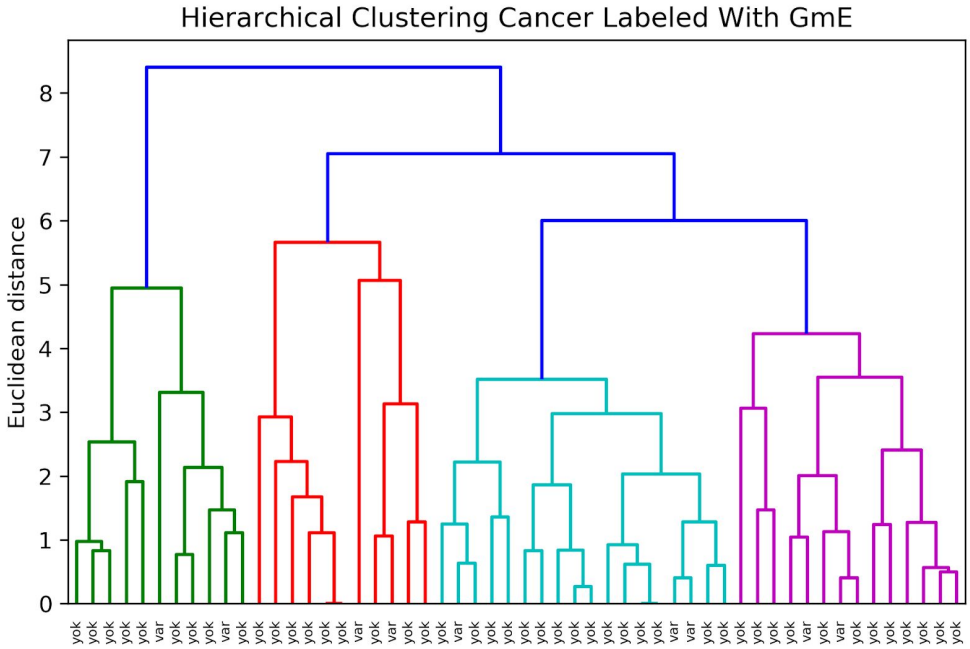

Figure 22: Hierarchical Clustering Dendrogram of GmE

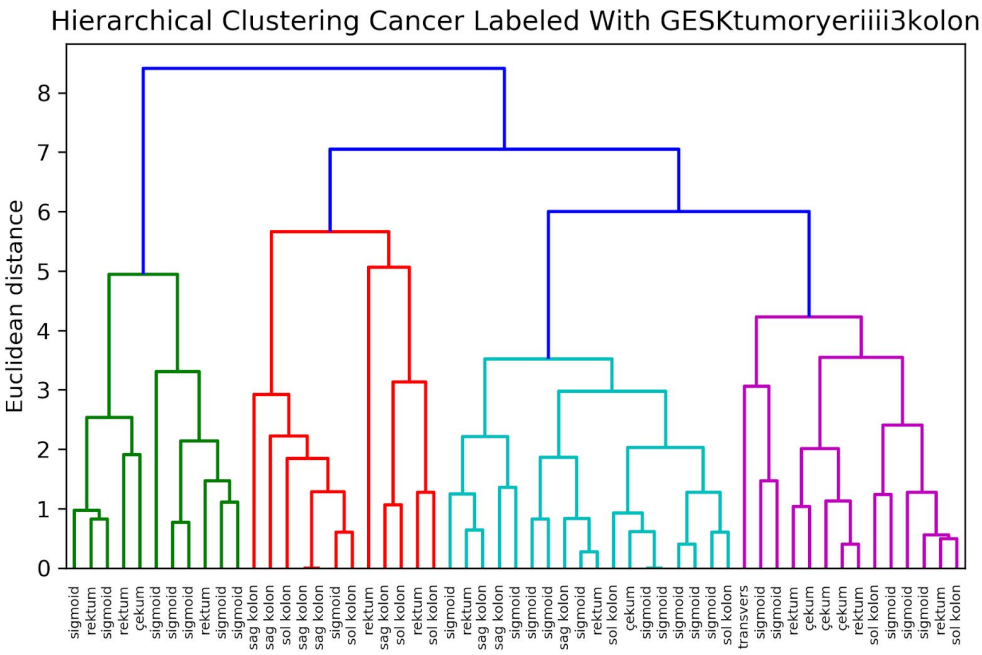

Figure 23: Hierarchical Clustering Dendrogram of GESKtumoryeriiii3kolon

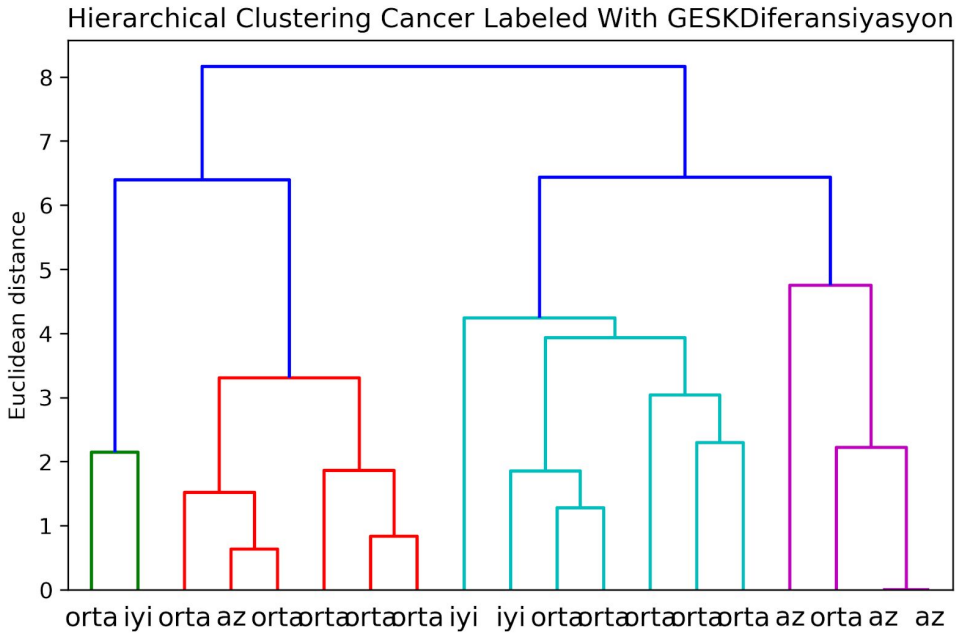

Figure 24: Hierarchical Clustering Dendrogram of GESKDiferansiyasyon

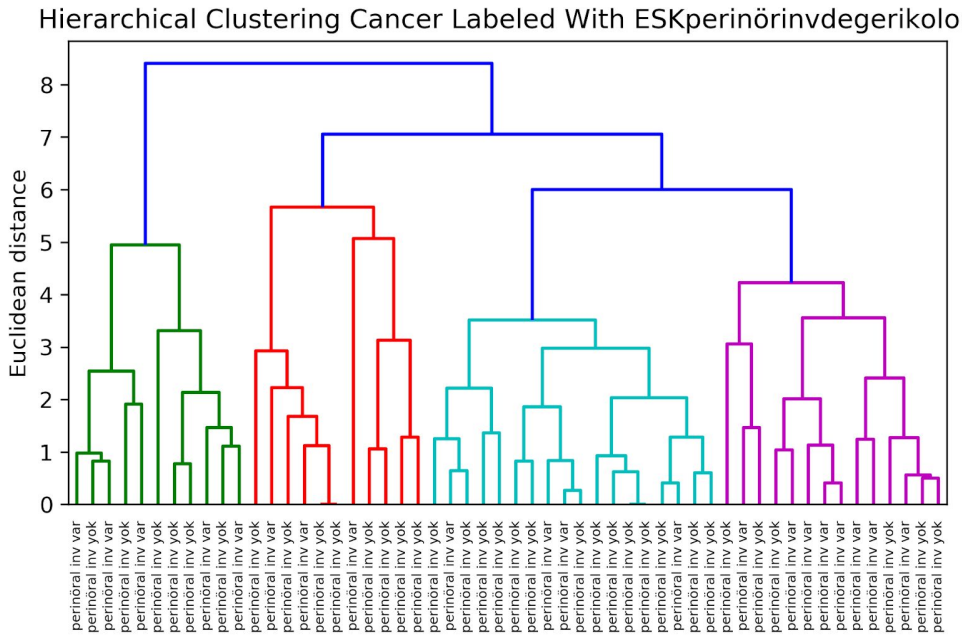

Figure 25: Hierarchical Clustering Dendrogram of ESKperinörinvdegerikolon