Delphi Uthirakulathu

Machine Learning Project: Cervical Cancer Prediction

Part 1: Defining the Problem

Business Objective

Cervical cancer is the fourth most common cancer affecting women worldwide; therefore, it is a very impactful area of research for women's health. The business objective of this project is to develop a classification model which can utilize a patient's past health history to predict whether they have a high risk for cervical cancer. The biopsy result will be used as the target variable and label to classify a patient as high risk (positive = high risk, negative = not high risk). This model could help diagnose patients who otherwise would not be able to access proper medical care due to financial or socioeconomic challenges. It would benefit women's health overall by making diagnoses easier to access and reduce the burden involved with expensive testing and complex procedures.

Assumptions

1. The target variable to predict cancer risk will be the "biopsy" column, where 1 indicates a positive diagnosis and 0 indicates a negative diagnosis. We will assume that a positive biopsy result indicates that the patient has a high risk of cervical cancer, and a negative biopsy result indicates that the patient does not have high risk.
2. The target variable chosen for modeling is specifically predicting the results of a biopsy, so we will not use "Dx: cancer" as a target parameter, rather as a feature. In this model, we will assume that the "Dx: cancer" column represents whether a physician diagnosed the patient as having cancer, independent of the biopsy.
3. Each data point represents a unique patient, and one patient's health history is completely independent of another's.
4. Missing attribute values represent cases where a patient chose not to divulge information. We will assume that filling these values in appropriately will not introduce any bias. The plan for filling in missing values is included in Part 2: Data Analysis..
5. Columns which contain only values of 0 or 1 are assumed to be binary indicators. For example, 0 in the "smokes" column indicates that the patient is not a smoker and 1 indicates that the patient is a smoker.
6. Parameters with only one unique value in the entire dataset will be excluded from the feature space since they do not provide significant information gain to machine learning models. We will assume that this will not introduce any bias.

Solving the Problem without Machine Learning

There are standard medical procedures such as pap tests, biopsies, medical imaging, and colposcopies, which are currently used to diagnose cervical cancer. Physicians have also used statistical models to determine the correlation between specific health factors (smoking, number of pregnancies, etc.) and cervical cancer to form predictions. There have also been studies done to highlight patterns in the target population, which led to the creation of scoring systems to track a patient's risk for developing cervical cancer.

Part 2: Data Analysis

The dataset was collected at 'Hospital Universitario de Caracas' in Caracas, Venezuela. Each row in the dataset represents a unique patient. Some columns may have missing values because patients chose not to answer some of the questions.

Dataset: https://archive.ics.uci.edu/dataset/383/cervical+cancer+risk+factors

Data Dictionary

See Section 1 of Jupyter Notebook for accompanying code.

Figure 1: Data dictionary table.

Parameter	Definition	Units	Missing Values
Age	Age of patient	Years	No
Number of sexual partners	Count of sexual partners for patient	Count	Yes
First sexual intercourse	Patient's age at first sexual intercourse	Years	Yes
Num of pregnancies	Count of patient's pregnancies	Count	Yes
Smokes	Patient smokes (yes/no)	Binary Indicator	Yes
Smokes (years)	Number of years patient has been smoking	Years	Yes
Smokes (packs/year)	Number of packs patient smokes per year	Years	Yes
Hormonal Contraceptives	Patient has taken hormonal contraceptive (yes/no)	Binary Indicator	Yes
Hormonal Contraceptives (years)	Number of years patient has taken hormonal contraceptives	Years	Yes
IUD	Patient has had IUD	Binary Indicator	Yes
IUD (years)	Number of years patient has had IUD	Years	Yes
STDs	Patient has had an STD (yes/no)	Binary Indicator	Yes
STDs (number)	Number of STDs patient reports	Count	Yes
STDs:condylomatosis	Patient has had condylomatosis (yes/no)	Binary Indicator	Yes
STDs:cervical condylomatosis	Patient has had cervical condylomatosis (yes/no)	Binary Indicator	Yes

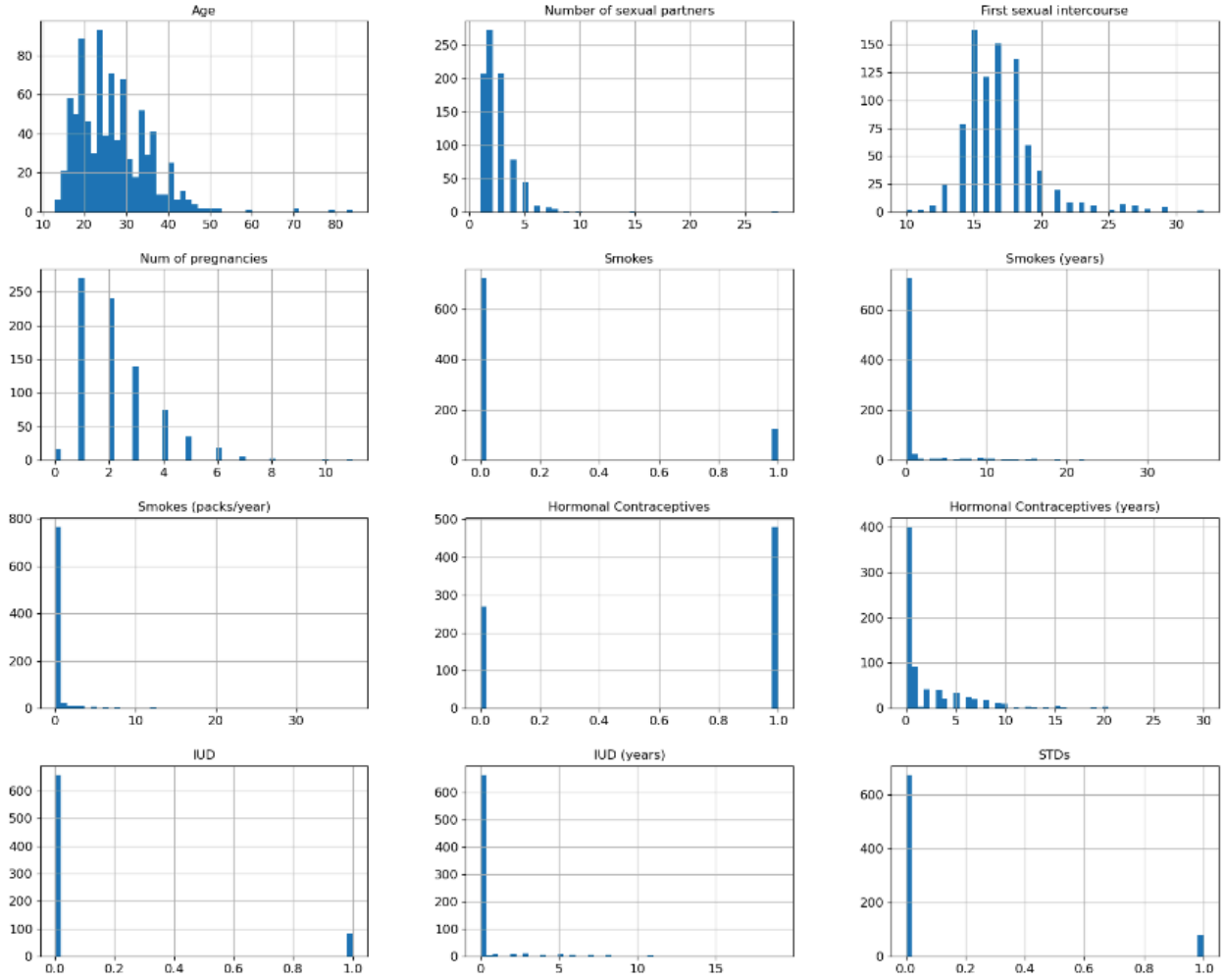
STDs:vaginal condylomatosis	Patient has had vaginal condylomatosis (yes/no)	Binary Indicator	Yes
STDs:vulvo-perineal condylomatosis	Patient has had vulvo-perineal condylomatosis (yes/no)	Binary Indicator	Yes
STDs:syphilis	Patient has had syphilis (yes/no)	Binary Indicator	Yes
STDs:pelvic inflammatory disease	Patient has had pelvic inflammatory disease (yes/no)	Binary Indicator	Yes
STDs:genital herpes	Patient has had genital herpes (yes/no)	Binary Indicator	Yes
STDs:molluscum contagiosum	Patient has had molluscum contagiosum (yes/no)	Binary Indicator	Yes
STDs:AIDS	Patient has had AIDs (yes/no)	Binary Indicator	Yes
STDs:HIV	Patient has had HIV (yes/no)	Binary Indicator	Yes
STDs:Hepatitis B	Patient has had Hepatitis B (yes/no)	Binary Indicator	Yes
STDs:HPV	Patient has had HPV (yes/no)	Binary Indicator	Yes
STDs: Number of diagnosis	Number of STDs patient has been diagnosed with	Count	No
STDs: Time since first diagnosis	Years since patient's first STD diagnosis	Years	Yes
STDs: Time since last diagnosis	Years since patient's last STD diagnosis	Years	Yes
Dx:Cancer	Patient has been diagnosed with cancer previously (yes/no)	Binary Indicator	No
Dx:CIN	Patient has been diagnosed with CIN (yes/no)	Binary Indicator	No
Dx:HPV	Patient has been diagnosed with HPV (yes/no)	Binary Indicator	No
Dx	Definition is unclear. May need to be excluded from the feature set.	Binary Indicator	No
Hinselmann	Patient has positive colposcopy result (yes/no)	Binary Indicator	No

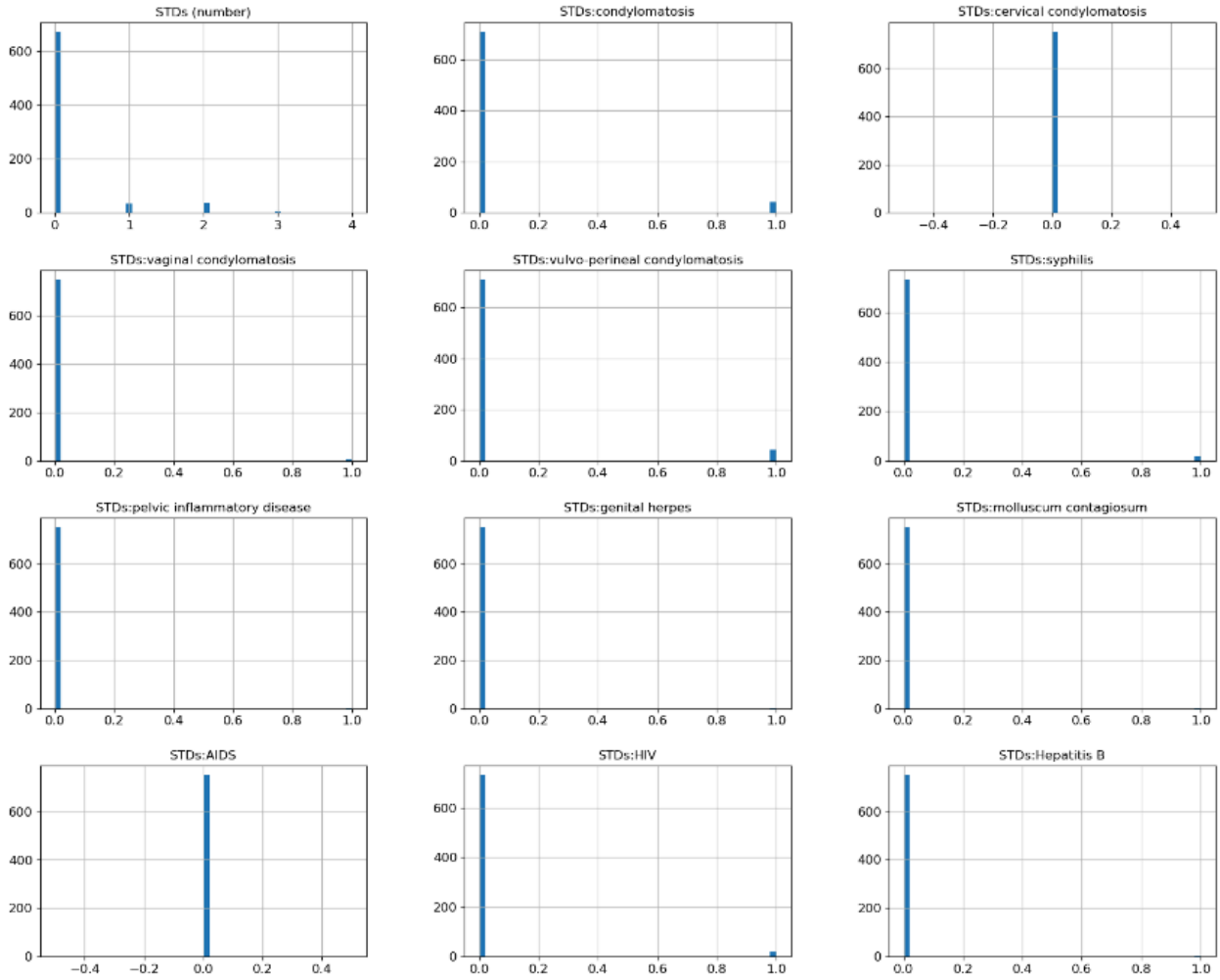
Schiller	Patient has positive Schiller result (yes/no)	Binary Indicator	No
Citology	Patient has positive cytology result (yes/no)	Binary Indicator	No
Biopsy	Patient has positive biopsy result (yes/no)	Binary Indicator	No

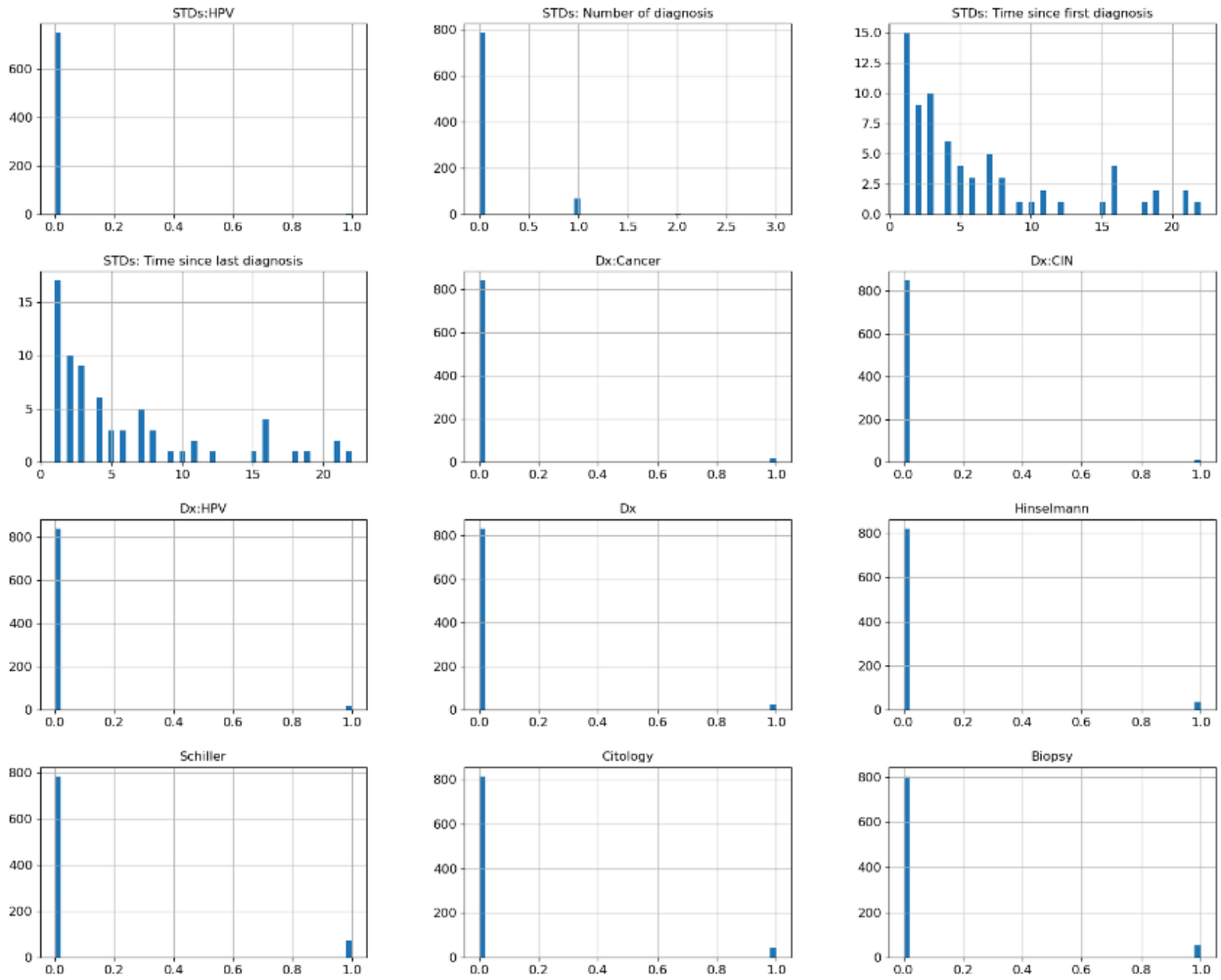
Data Visualization

See Section 2 of Jupyter Notebook for accompanying code.

Figure 2: Histograms plotting each parameter in original dataset.




STDs (number)
STDs:condylomatosis
STDs:cervical condylomatosis
STDs:vaginal condylomatosis
STDs:vulvo-perineal condylomatosis
STDs:syphilis
STDs:pelvic inflammatory disease
STDs:genital herpes
STDs:molluscum contagiosum
STDs:AIDS
STDs:HIV
STDs:Hepatitis B



These histograms highlight key points that must be taken into consideration when preparing the data. The first point is that some parameters contain the same value for all rows, such as ‘STDs: AIDs’. Features with the same value in the entire dataset will not provide significant information to the model, so they were removed. Another key point is that several parameters, such as ‘Smokes’ and ‘Biopsy’, do not have an equal frequency across values. Therefore, it is important to ensure that when the data is split into training and test data, there is an equivalent proportion of values that is representative of the true population.

Additional descriptive insights on the dataset include the following. The ages of patients in the dataset range from 13-84. The average number of pregnancies is 2 (rounded to whole number). The average number of years patients have taken hormonal contraceptives is 2.25 years. 50 percent of patients have had 2 pregnancies or more at the time the dataset was collected.

Figure 3: Heatmap of parameters from original dataset.

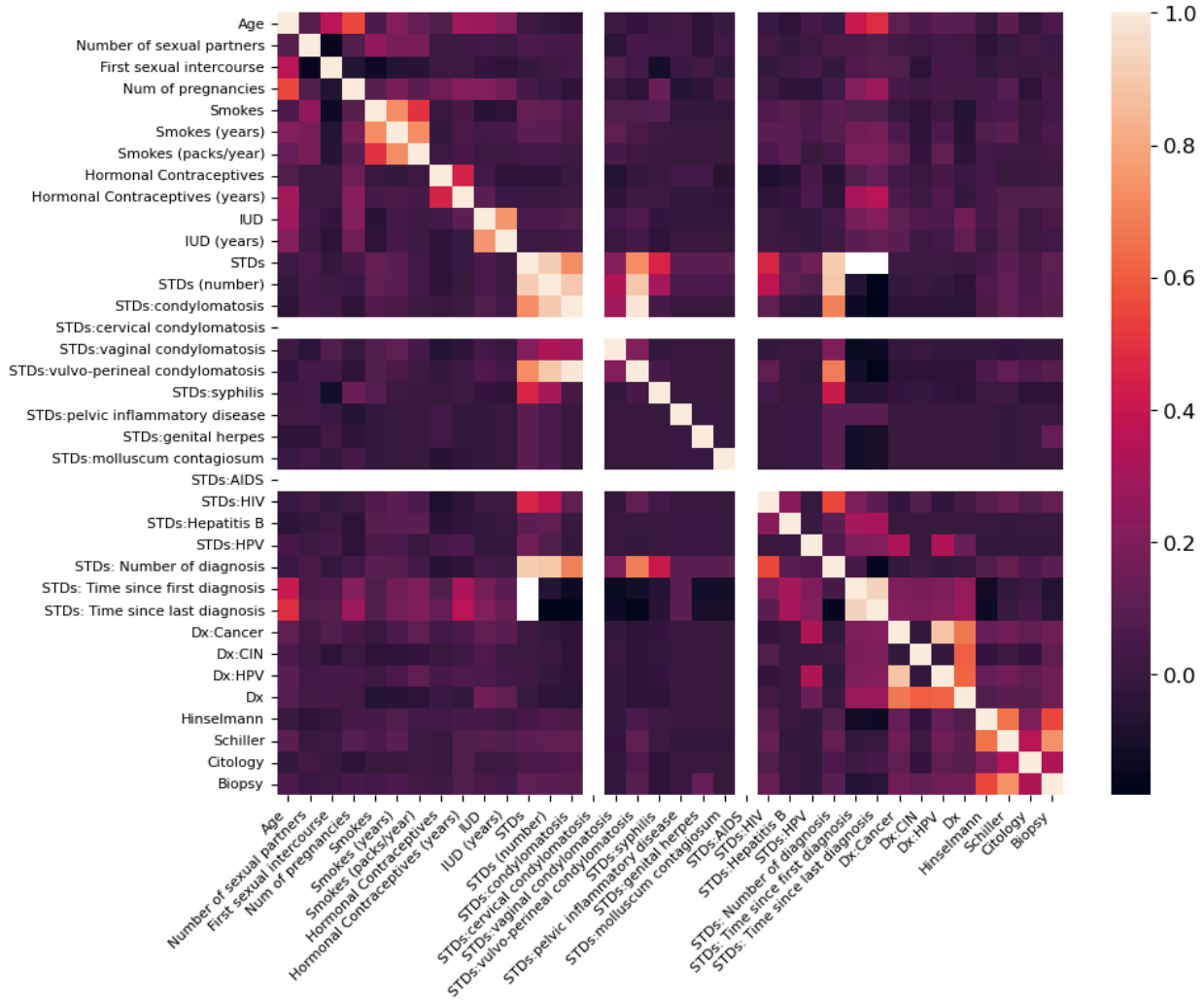


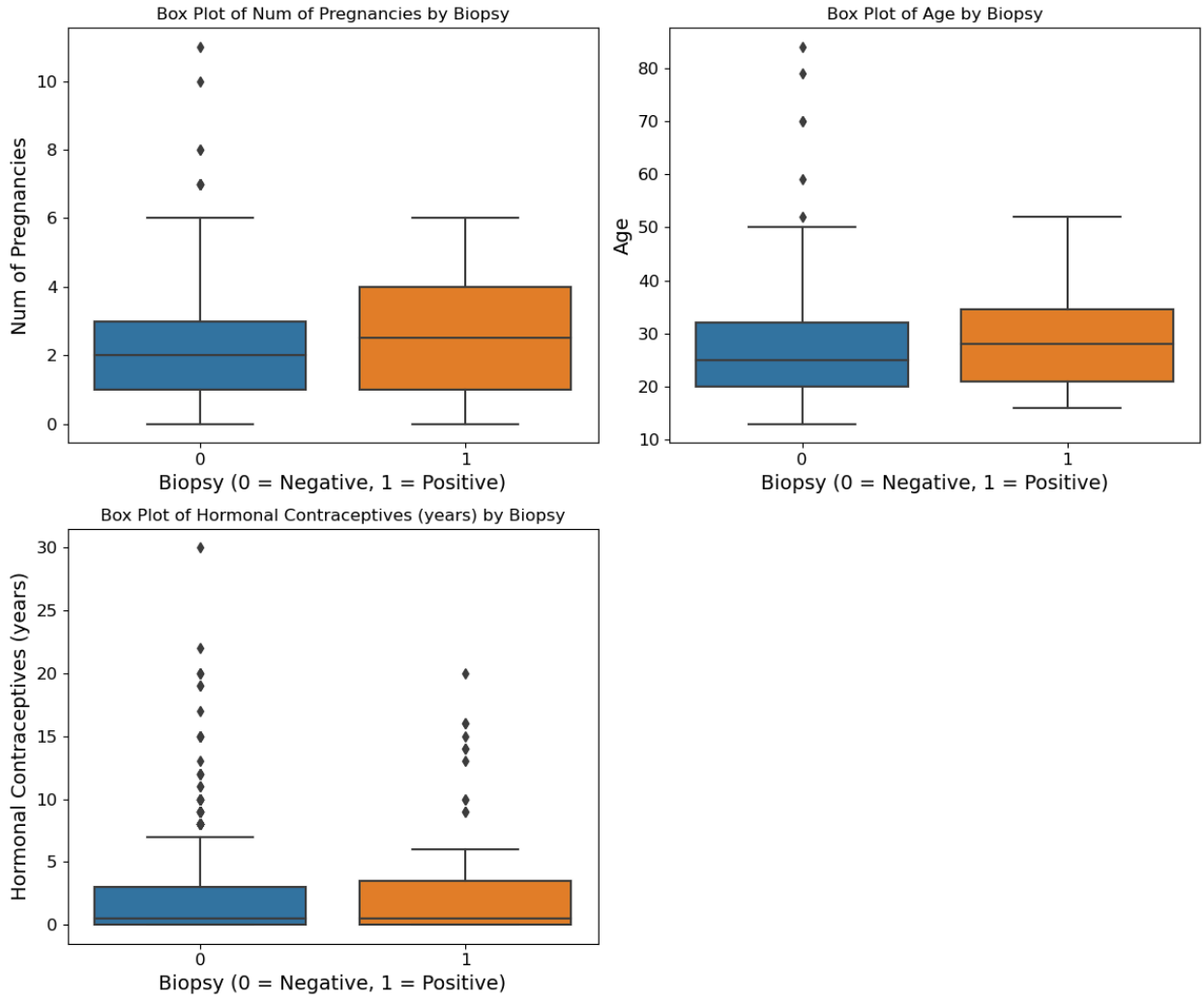
Figure 4: Correlation of parameters to the target variable, 'Biopsy.'

Biopsy	1
Schiller	0.733204
Hinselmann	0.547417
Citology	0.327466
Dx:Cancer	0.160905
Dx:HPV	0.160905
Dx	0.157607
STDs:genital herpes	0.132526
STDs:HIV	0.12688
Dx:CIN	0.113172

STDs	0.109099
STDs (number)	0.098347
STDs: Number of diagnosis	0.097449
STDs:vulvo-perineal condylomatosis	0.088902
STDs:condylomatosis	0.08639
Hormonal Contraceptives (years)	0.079388
Smokes (years)	0.062044
Age	0.055956
IUD	0.053194
Num of pregnancies	0.046416
IUD (years)	0.033275
Smokes	0.029356
Smokes (packs/year)	0.024882
Hormonal Contraceptives	0.00775
First sexual intercourse	0.007264
Number of sexual partners	-0.001442
STDs:pelvic inflammatory disease	-0.010034
STDs:molluscum contagiosum	-0.010034
STDs:Hepatitis B	-0.010034
STDs:HPV	-0.0142
STDs:vaginal condylomatosis	-0.020108
STDs:syphilis	-0.043061
STDs: Time since last diagnosis	-0.047585
STDs: Time since first diagnosis	-0.070153
STDs:cervical condylomatosis	NaN
STDs:AIDS	NaN

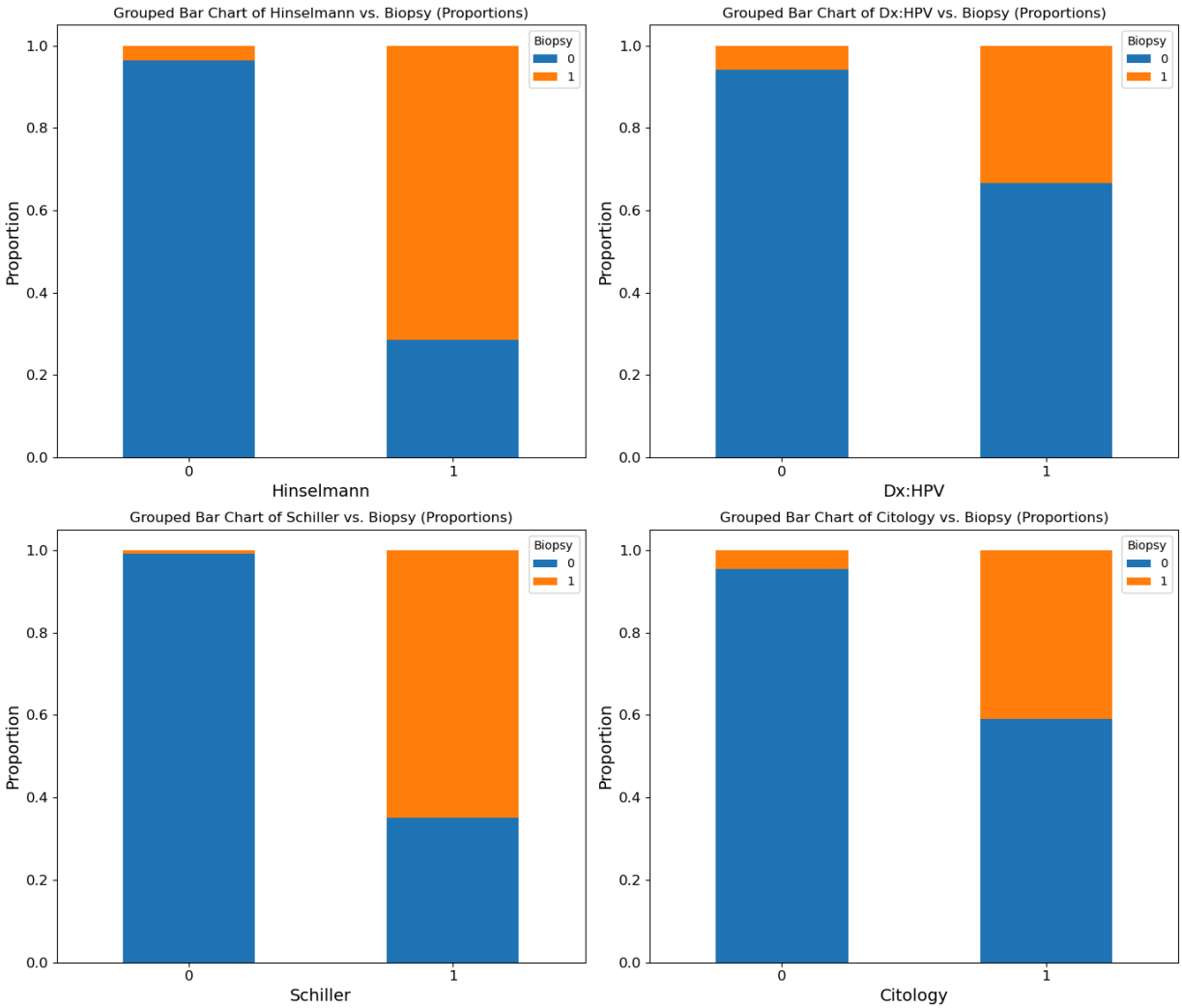
Based on the heatmap and correlation matrix, features related to cervical cancer testing (‘Schiller’, ‘Hinselmann’, ‘Citology’), specific STDs (‘STDs:genital herpes’, ‘STDs:HIV’), and two diagnoses (‘Dx: Cancer’, ‘Dx:HIV’) appear to correlate most with biopsy results. Other features have weaker correlations; therefore, feature extraction may be useful when implementing models.

Figure 5: Boxplots of three continuous data features vs. binary biopsy results.



There appears to be a slight pattern showing positive biopsy results occurring with a higher value for number of pregnancies, age, and years of taking hormonal contraceptives. This is shown by the greater interquartile range and the higher upper quartile value of the boxplot for a biopsy value of 1.

Figure 6: Stacked bar charts for three binary indicator features vs. biopsy results. Biopsy results are shown as a proportion.



It appears that a higher proportion of positive biopsy results appear with positive results for colposcopy (Hinselmann), Schiller testing, Cytology testing, and HPV diagnosis. This aligns with the results seen in the correlation matrix.

Separating Data

See Section 3 of Jupyter Notebook for accompanying code.

The original data was split into training and test data using train_test_split from scikit-learn. 20 percent of the data was utilized for testing.

The proportions for the features ‘Biopsy’ and ‘Smokes’ were chosen as examples to be validated in the original dataset, training data, and test data. This was done to ensure that the spread of data in each set is representative of the population.

Plan for Missing Data

See Section 4 of Jupyter Notebook for accompanying code.

For features that are binary indicators, missing values were replaced with the median of the training data set.

All other features have continuous data, so those missing values were replaced with the mean of the training data set.

Plan for Additional Parameters or Data

No additional parameters or data will be added at this time.

Transformations

See Section 5 of Jupyter Notebook for accompanying code.

1. Features that contain the same value in all rows were removed from the feature space.
2. Missing data was filled in as mentioned above. The training data mean and median were used to fill in missing values for the test data.
3. Continuous data was standardized using StandardScaler from scikit-learn. StandardScaler was fit only on the training data and was used to transform the training and test data. Binary data was not standardized to prevent confusion in the model which may lead to poor performance.
4. Additional dimensionality reduction may be pursued in later stages once initial exploration of model performance is conducted.

Part 3: Modeling

The dataset was evaluated using k means clustering, linear kernel support vector machine (SVM), RBF kernel support vector machine, and logistic regression models. All models underwent hyperparameter tuning using GridSearchCV from scikit-learn.

Additional Dimensionality Reduction

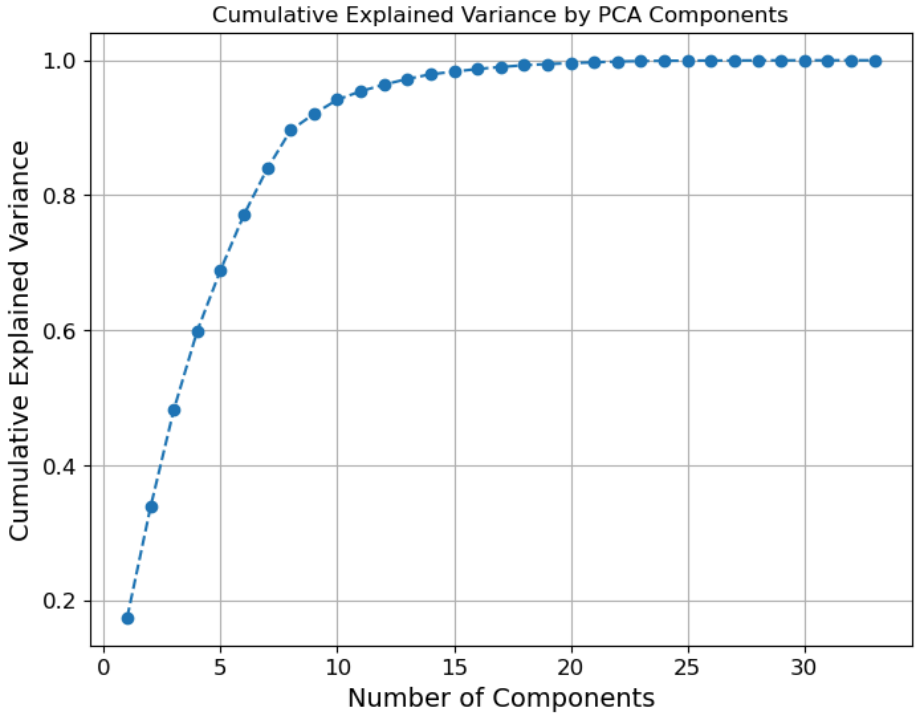
See Section 6 of Jupyter Notebook for accompanying code.

All three models were run with three different versions to compare the effects of additional dimensionality reduction beyond what was done in the transformations section (omitted features with same value in all rows). The three versions are: no additional dimensionality reduction, linear discriminant analysis (LDA), and principal component analysis (PCA).

For LDA, the number of components was set to one (number of classes - 1) since there are two classes (0 and 1) for the target variable ‘Biopsy.’

For PCA, the number of components was determined based on the cumulative explained variance. The number of components was set to seven in order to capture at least 80 percent of the variance. See figure 7 below.

Figure 7: Cumulative Explained Variance by PCA Components



K Means Clustering Model

See Section 7 of Jupyter Notebook for accompanying code.

K means clustering is a form of unsupervised learning, which was used to group the data points into clusters. Although the target variable is known, the number of clusters was not assumed to be two. Rather, it was included in the process of hyperparameter tuning.

The following options were chosen for hyperparameter tuning. The optimal parameters were determined using GridSearchCV to assess accuracy.

Figure 8: Hyperparameter Tuning for K Means Clustering

Parameters	Values Tested	Optimal Values
n_clusters	2, 3, 4, 5, 6	2
init	k-means++, random	k-means++
n_init	5, 10, 15	15
max_iter	300, 500	300
tol	1e-4, 1e-3	0.001
algorithm	lloyd, elkan	lloyd

Since the optimal number of clusters was determined to be two, the clusters were assigned a label which corresponded to the true value of the majority of data points. The model was still run in an unsupervised

manner. The labels were simply assigned to calculate accuracy, which was used to compare the results of this model to the SVM models and logistic regression.

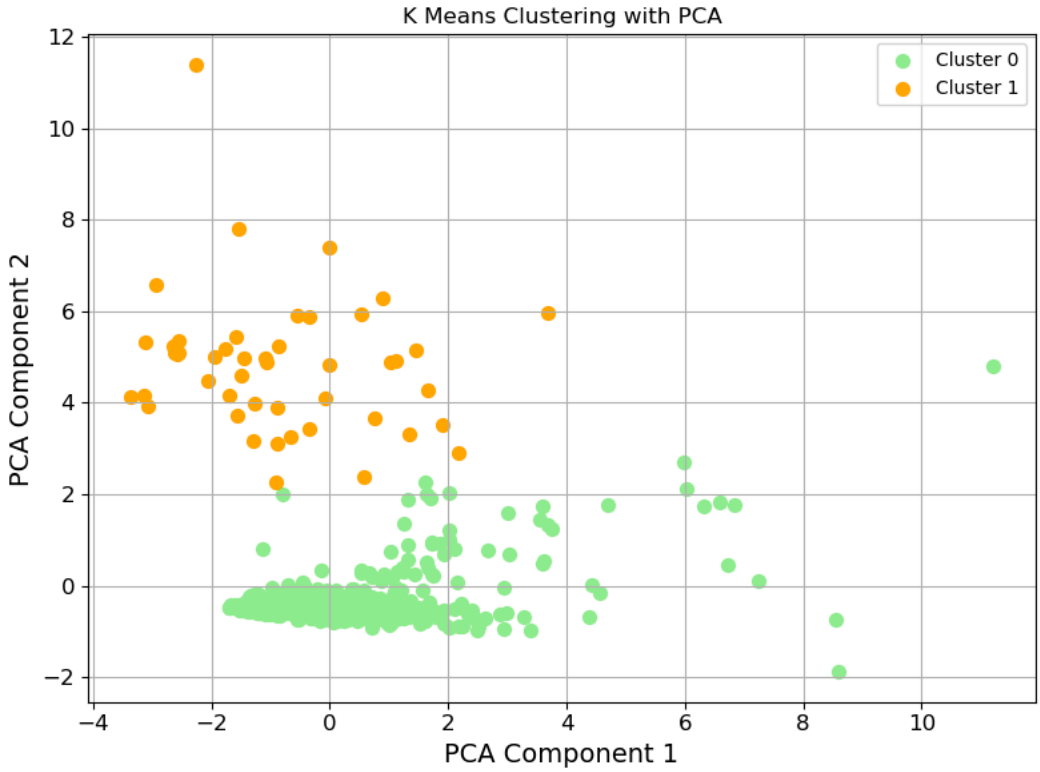
Figure 9: Accuracies and Confusion Matrices for K Means Model

No Additional Dimensionality Reduction		**LDA (1 component)**		**PCA (7 components)**	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.9359	0.936	0.965	0.9419	0.9359	0.936

```
Results without dimensionality reduction:
Training Confusion Matrix:
[[642   0]
 [ 44   0]]
Training Accuracy: 0.9359

Test Confusion Matrix:
[[161   0]
 [ 11   0]]
Test Accuracy: 0.9360

Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[621  21]
 [  3  41]]
Training Accuracy: 0.9650

Test Confusion Matrix:
[[152   9]
 [  1  10]]
Test Accuracy: 0.9419

Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[642   0]
 [ 44   0]]
Training Accuracy: 0.9359

Test Confusion Matrix:
[[161   0]
 [ 11   0]]
Test Accuracy: 0.9360
```

For this model, dimensionality reduction using LDA resulted in the best performance since it produced the highest training and test accuracy. It is also the only version of the model that produced any true positive results (see confusion matrices in figure 9), which is important for predicting true high risk cases.

Figure 10: Visualization of k means clusters with PCA. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



Linear Kernel Support Vector Machines (SVM) Model

See Section 8 of Jupyter Notebook for accompanying code.

The SVM model is a form of supervised learning, which was used to classify the data points via a decision boundary. This section covers an SVM model which utilizes a linear decision boundary to separate classes. The class_weight parameter was set to 'balanced' in order to account for the larger number of negative biopsy examples in the dataset. This allows weights to be adjusted inversely proportional to class frequencies in the data *(Reference: https://scikit-learn.org/dev/modules/generated/sklearn.svm.LinearSVC.html)*.

The following options were chosen for hyperparameter tuning. The optimal parameters were determined using GridSearchCV to assess accuracy.

Figure 11: Hyperparameter Tuning for Linear Kernel SVM

Parameters	Values Tested	Optimal Values
C	0.1, 1, 10, 100	1
max_iter	1000, 5000, 10000	1000
tol	1e-3, 1e-4, 1e-5	0.001

The optimal values displayed in figure 11 were used for all three versions of the linear kernel SVM model when testing the effects of dimensionality reduction.

Figure 12: Accuracies and Confusion Matrices for Linear Kernel SVM Model

No Additional Dimensionality Reduction		LDA (1 component)		PCA (7 components)	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.965	0.936	0.965	0.9419	0.691	0.75

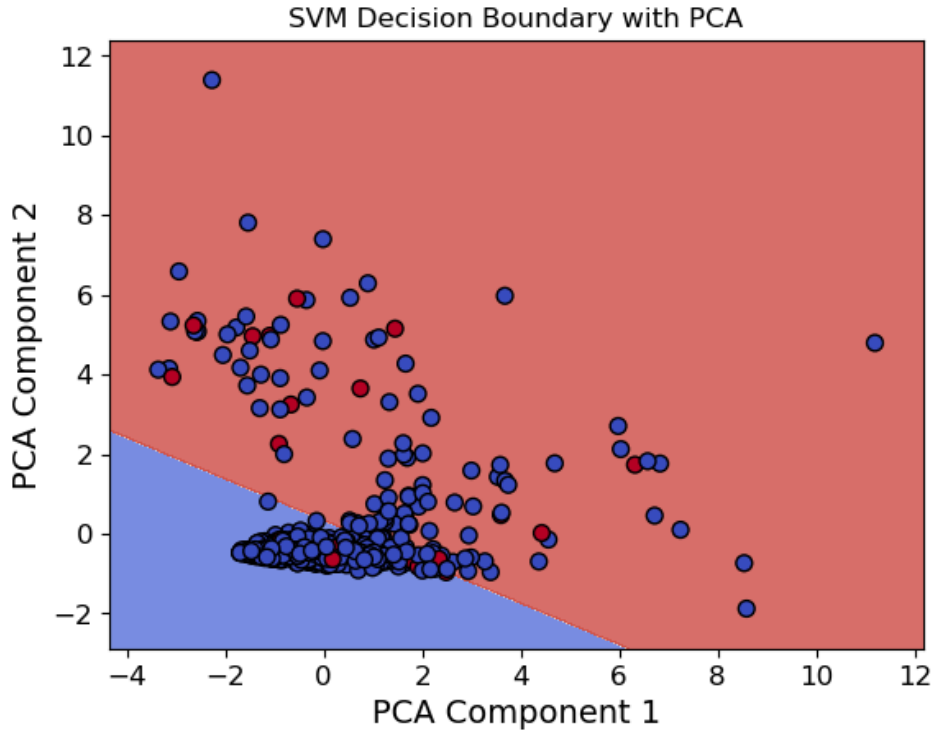
```
Results without dimensionality reduction:
Training Confusion Matrix:
[[621  21]
 [  3  41]]
Training Accuracy: 0.9650

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       642
           1       0.66      0.93      0.77        44

    accuracy                           0.97       686
   macro avg       0.83      0.95      0.88       686
weighted avg       0.97      0.97      0.97       686


Test Confusion Matrix:
[[152   9]
 [  2   9]]
Test Accuracy: 0.9360

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.50      0.82      0.62        11

    accuracy                           0.94       172
   macro avg       0.74      0.88      0.79       172
weighted avg       0.96      0.94      0.94       172
```

```
Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[621  21]
 [  3  41]]
Training Accuracy: 0.9650

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       642
           1       0.66      0.93      0.77        44

    accuracy                           0.97       686
   macro avg       0.83      0.95      0.88       686
weighted avg       0.97      0.97      0.97       686


Test Confusion Matrix:
[[152   9]
 [  1  10]]
Test Accuracy: 0.9419

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.53      0.91      0.67        11

    accuracy                           0.94       172
   macro avg       0.76      0.93      0.82       172
weighted avg       0.96      0.94      0.95       172
```

```
Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[452 190]
 [ 22  22]]
Training Accuracy: 0.6910

Training Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.70      0.81       642
           1       0.10      0.50      0.17        44

    accuracy                           0.69       686
   macro avg       0.53      0.60      0.49       686
weighted avg       0.90      0.69      0.77       686


Test Confusion Matrix:
[[123  38]
 [  5   6]]
Test Accuracy: 0.7500

Test Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.76      0.85       161
           1       0.14      0.55      0.22        11

    accuracy                           0.75       172
   macro avg       0.55      0.65      0.53       172
weighted avg       0.91      0.75      0.81       172
```

For this model, dimensionality reduction using LDA also resulted in the best performance. The training accuracy for no additional dimensionality reduction is equivalent to LDA; however, the test accuracy for LDA is slightly higher.

Figure 13: Visualization of Linear Kernel SVM decision boundary with PCA. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



RBF Kernel Support Vector Machines (SVM) Model

See Section 9 of Jupyter Notebook for accompanying code.

The SVM model can also implement non-linear separation of classes. The RBF kernel was utilized for this purpose because it can form more complex decision boundaries.

As before, hyperparameter tuning was done using GridSearchCV, and the optimal values were used for three versions of this model.

Figure 14: Hyperparameter Tuning for RBF Kernel SVM

Parameters	Values Tested	Optimal Values
C	0.1, 1, 10, 100	10
gamma	1, 0.1, 0.01, 0.001	0.001

Figure 15: Accuracies and Confusion Matrices for RBF Kernel SVM Model

No Additional Dimensionality Reduction		LDA (1 component)		PCA (7 components)	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.9636	0.9535	0.9636	0.9419	0.7959	0.8314

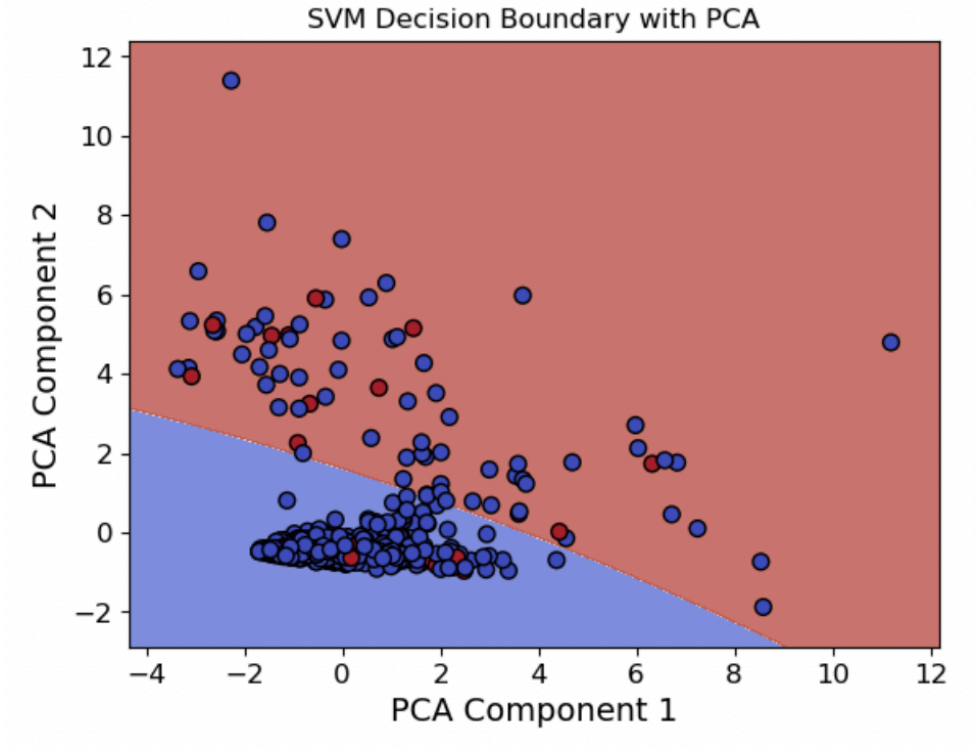
```
Results without dimensionality reduction:
Training Confusion Matrix:
[[623  19]
 [  6  38]]
Training Accuracy: 0.9636

Training Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       642
           1       0.67      0.86      0.75        44

    accuracy                           0.96       686
   macro avg       0.83      0.92      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[154   7]
 [  1  10]]
Test Accuracy: 0.9535

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.96      0.97       161
           1       0.59      0.91      0.71        11

    accuracy                           0.95       172
   macro avg       0.79      0.93      0.84       172
weighted avg       0.97      0.95      0.96       172
```

```
Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[620  22]
 [  3  41]]
Training Accuracy: 0.9636

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       642
           1       0.65      0.93      0.77        44

    accuracy                           0.96       686
   macro avg       0.82      0.95      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[152   9]
 [  1  10]]
Test Accuracy: 0.9419

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.53      0.91      0.67        11

    accuracy                           0.94       172
   macro avg       0.76      0.93      0.82       172
weighted avg       0.96      0.94      0.95       172
```

```
Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[528 114]
 [ 26  18]]
Training Accuracy: 0.7959

Training Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.82      0.88       642
           1       0.14      0.41      0.20        44

    accuracy                           0.80       686
   macro avg       0.54      0.62      0.54       686
weighted avg       0.90      0.80      0.84       686


Test Confusion Matrix:
[[139  22]
 [  7   4]]
Test Accuracy: 0.8314

Test Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.86      0.91       161
           1       0.15      0.36      0.22        11

    accuracy                           0.83       172
   macro avg       0.55      0.61      0.56       172
weighted avg       0.90      0.83      0.86       172
```

For this model, using no additional dimensionality reduction resulted in the highest test accuracy across the three versions. Implementing LDA and PCA did not improve the accuracy of the model in this case.

Figure 16: Visualization of RBF Kernel SVM decision boundary with PCA. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



Logistic Regression Model

See Section 10 of Jupyter Notebook for accompanying code.

The logistic regression model is a supervised learning model which performs binary classification of data. Since the target variable ‘Biopsy’ is a binary value, this model was selected for testing as well.

Similar to the previous models, hyperparameter tuning was done using GridSearchCV, and the optimal values were used for three versions of this model.

Figure 17: Hyperparameter Tuning for Logistic Regression

Parameters	Values Tested	Optimal Values
C	0.1, 1, 10, 100	1
max_iter	1000, 5000, 10000	1000
tol	1e-3, 1e-4, 1e-5	0.001

Figure 18: Accuracies and Confusion Matrices for Logistic Regression Model

No Additional Dimensionality Reduction		LDA (1 component)		PCA (7 components)	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.9636	0.936	0.9621	0.9419	0.691	0.7442

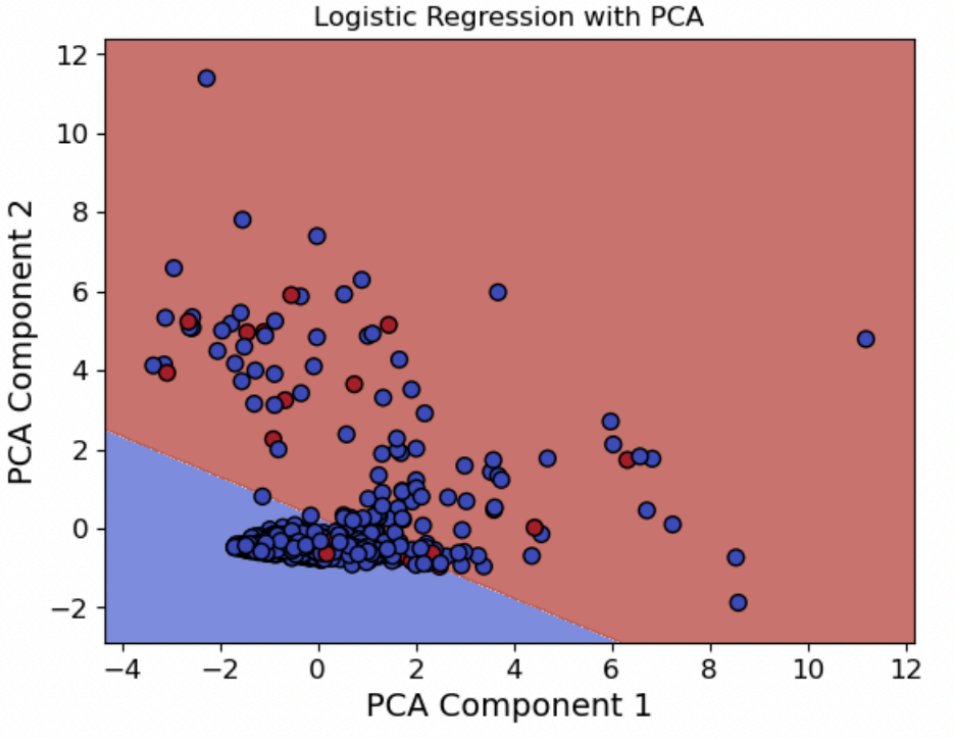
```
Results without dimensionality reduction:
Training Confusion Matrix:
[[620  22]
 [  3  41]]
Training Accuracy: 0.9636

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       642
           1       0.65      0.93      0.77        44

    accuracy                           0.96       686
   macro avg       0.82      0.95      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[152   9]
 [  2   9]]
Test Accuracy: 0.9360

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.50      0.82      0.62        11

    accuracy                           0.94       172
   macro avg       0.74      0.88      0.79       172
weighted avg       0.96      0.94      0.94       172
```

```
Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[619  23]
 [  3  41]]
Training Accuracy: 0.9621

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98       642
           1       0.64      0.93      0.76        44

    accuracy                           0.96       686
   macro avg       0.82      0.95      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[152   9]
 [  1  10]]
Test Accuracy: 0.9419

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.53      0.91      0.67        11

    accuracy                           0.94       172
   macro avg       0.76      0.93      0.82       172
weighted avg       0.96      0.94      0.95       172
```

```
Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[452 190]
 [ 22  22]]
Training Accuracy: 0.6910

Training Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.70      0.81       642
           1       0.10      0.50      0.17        44

    accuracy                           0.69       686
   macro avg       0.53      0.60      0.49       686
weighted avg       0.90      0.69      0.77       686


Test Confusion Matrix:
[[122  39]
 [  5   6]]
Test Accuracy: 0.7442

Test Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.76      0.85       161
           1       0.13      0.55      0.21        11

    accuracy                           0.74       172
   macro avg       0.55      0.65      0.53       172
weighted avg       0.91      0.74      0.81       172
```

For this model, dimensionality reduction using LDA resulted in the highest test accuracy across the three versions. The highest training accuracy resulted from performing no additional dimensionality reduction.

Figure 19: Visualization of Logistic Regression boundary with PCA. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



Summary of Modeling

Figure 20: Accuracies and Confusion Matrices for all Models

	No Additional Dimensionality Reduction		**LDA (1 component)**		**PCA (7 components)**	
Model	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
K Means	0.9359	0.936	0.965	0.9419	0.9359	0.936
SVM LinearSVC	0.965	0.936	0.965	0.9419	0.691	0.75
SVM RBF	0.9636	0.9535	0.9636	0.9419	0.7959	0.8314
Logistic Regression	0.9636	0.936	0.9621	0.9419	0.691	0.7442

All four of the models performed reasonably well for this dataset as they each have scenarios that resulted in an accuracy of greater than 90 percent. However, the ideal model, based on the metrics displayed in figure 20, appears to be the RBF kernel SVM with no additional dimensionality reduction conducted. This model resulted in a test accuracy of 0.9535, which is the highest out of all models tested.

Although dimensionality reduction did not appear to improve the performance of the RBF kernel SVM model, it is worth noting that for the other three models, implementation of LDA did improve the test accuracy. However, PCA did not appear to work well for these models, as indicated by the lower accuracies. Therefore, for this dataset, maximizing separation between classes may be a better strategy than focusing on removing redundant features.

Part 4: Further Tuning of Hyperparameters

RBF Kernel Support Vector Machines (SVM) Model

See Section 11 of Jupyter Notebook for accompanying code.

The SVM model using the rbf kernel was selected for further tuning of hyperparameters since it performed the best out of all the tested models.

Additional values for the regularization parameter, C, and gamma were tested. The effect of the probability parameter was also tested to determine which value produced the highest accuracy. As before, GridSearchCV was utilized to determine the optimal parameters based on accuracy. Then, the model with the best parameters was run on the training dataset and the test dataset.

Fine tuning the hyperparameters with additional values resulted in the same optimal value for C and gamma. The optimal value for the probability parameter was determined to be True.

Figure 21: Further Hyperparameter Tuning for RBF Kernel SVM

Parameters	**Values Tested**	**Optimal Values**
C	0.1, 1, 10, 100, 1000	10
gamma	1, 0.1, 0.01, 0.001, 0.0001, 'auto', 'scale'	0.001
probability	True, False	TRUE

Figure 22: Accuracies and Confusion Matrices for RBF Kernel SVM Model (with further fine tuning)

No Additional Dimensionality Reduction		**LDA (1 component)**		**PCA (7 components)**	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.9636	0.9535	0.9636	0.9419	0.7959	0.8314

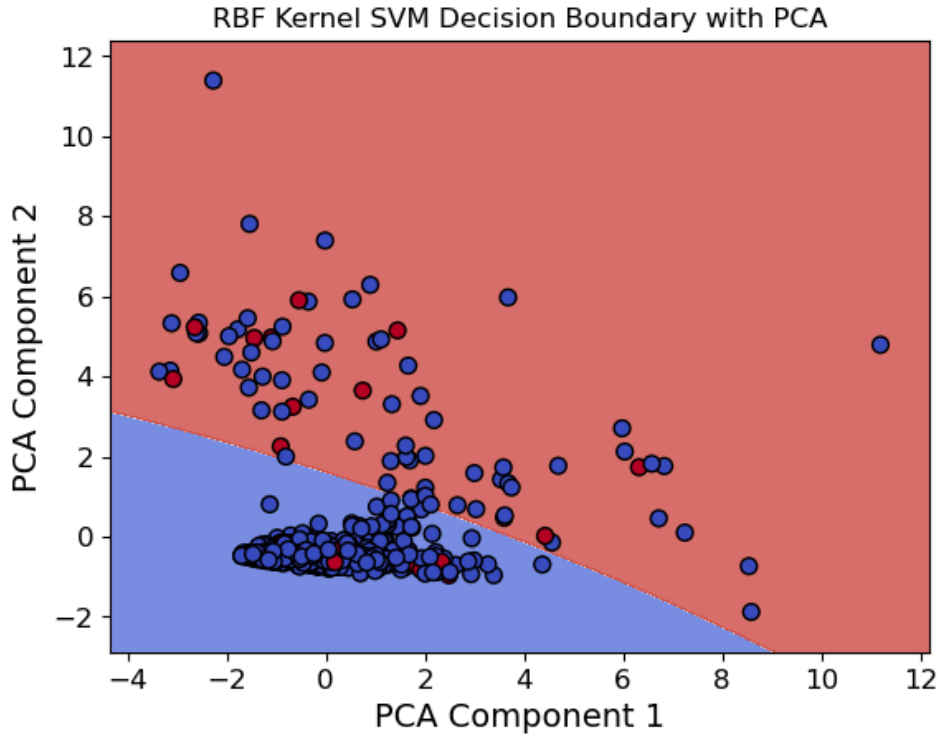
```
Results without dimensionality reduction:
Training Confusion Matrix:
[[623  19]
 [  6  38]]
Training Accuracy: 0.9636

Training Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       642
           1       0.67      0.86      0.75        44

    accuracy                           0.96       686
   macro avg       0.83      0.92      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[154   7]
 [  1  10]]
Test Accuracy: 0.9535

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.96      0.97       161
           1       0.59      0.91      0.71        11

    accuracy                           0.95       172
   macro avg       0.79      0.93      0.84       172
weighted avg       0.97      0.95      0.96       172
```

```
Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[620  22]
 [  3  41]]
Training Accuracy: 0.9636

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       642
           1       0.65      0.93      0.77        44

    accuracy                           0.96       686
   macro avg       0.82      0.95      0.87       686
weighted avg       0.97      0.96      0.97       686


Test Confusion Matrix:
[[152   9]
 [  1  10]]
Test Accuracy: 0.9419

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.94      0.97       161
           1       0.53      0.91      0.67        11

    accuracy                           0.94       172
   macro avg       0.76      0.93      0.82       172
weighted avg       0.96      0.94      0.95       172
```

```
Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[528 114]
 [ 26  18]]
Training Accuracy: 0.7959

Training Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.82      0.88       642
           1       0.14      0.41      0.20        44

    accuracy                           0.80       686
   macro avg       0.54      0.62      0.54       686
weighted avg       0.90      0.80      0.84       686


Test Confusion Matrix:
[[139  22]
 [  7   4]]
Test Accuracy: 0.8314

Test Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.86      0.91       161
           1       0.15      0.36      0.22        11

    accuracy                           0.83       172
   macro avg       0.55      0.61      0.56       172
weighted avg       0.90      0.83      0.86       172
```

The only hyperparameter which changed during further fine tuning was the probability parameter, which was previously using a default value of False. This did not appear to make a significant impact on the accuracy. This version of the model performs the same as the previous version when looking at the training and test accuracy.

Figure 23: Visualization of RBF Kernel SVM decision boundary with PCA after further hyperparameter tuning. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



Polynomial Kernel Support Vector Machines (SVM) Model

See Section 12 of Jupyter Notebook for accompanying code.

The polynomial kernel was also tested to see if this would result in improved model performance. The same hyperparameters used for the kernel RBF were tested with this model; however, the degree for the polynomial was also tested to determine the optimal value.

Figure 24: Hyperparameter Tuning for Polynomial Kernel SVM

Parameters	**Values Tested**	**Optimal Values**
C	0.1, 1, 10, 100, 1000	10
gamma	1, 0.1, 0.01, 0.001, 0.0001, 'auto', 'scale'	scale
probability	True, False	TRUE
degree	2, 3, 4, 5	2

Figure 25: Accuracies and Confusion Matrices for Polynomial Kernel SVM Model

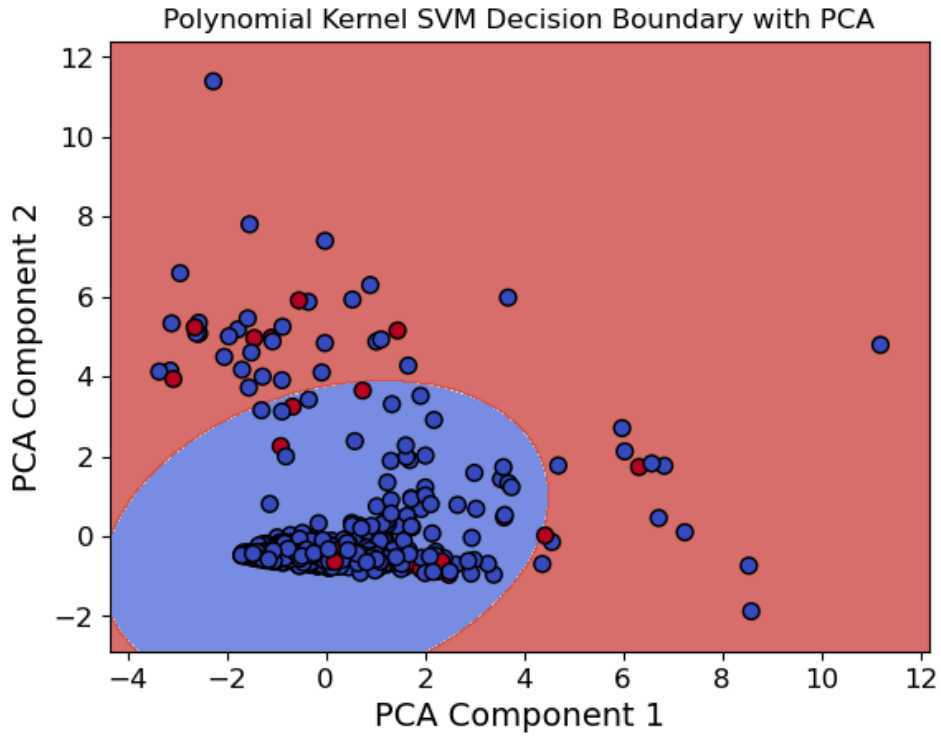
No Additional Dimensionality Reduction		LDA (1 component)		PCA (7 components)	
Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
0.9767	0.9477	0.9679	0.9477	0.8848	0.8953

```
Results without dimensionality reduction:
Training Confusion Matrix:
[[629  13]
 [  3  41]]
Training Accuracy: 0.9767

Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       642
           1       0.76      0.93      0.84        44

    accuracy                           0.98       686
   macro avg       0.88      0.96      0.91       686
weighted avg       0.98      0.98      0.98       686


Test Confusion Matrix:
[[158   3]
 [  6   5]]
Test Accuracy: 0.9477

Test Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       161
           1       0.62      0.45      0.53        11

    accuracy                           0.95       172
   macro avg       0.79      0.72      0.75       172
weighted avg       0.94      0.95      0.94       172
```

```
Results with LDA dimensionality reduction:
Training Confusion Matrix:
[[625  17]
 [  5  39]]
Training Accuracy: 0.9679

Training Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       642
           1       0.70      0.89      0.78        44

    accuracy                           0.97       686
   macro avg       0.84      0.93      0.88       686
weighted avg       0.97      0.97      0.97       686


Test Confusion Matrix:
[[153   8]
 [  1  10]]
Test Accuracy: 0.9477

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.95      0.97       161
           1       0.56      0.91      0.69        11

    accuracy                           0.95       172
   macro avg       0.77      0.93      0.83       172
weighted avg       0.97      0.95      0.95       172
```

```
Results with PCA dimensionality reduction:
Training Confusion Matrix:
[[590  52]
 [ 27  17]]
Training Accuracy: 0.8848

Training Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94       642
           1       0.25      0.39      0.30        44

    accuracy                           0.88       686
   macro avg       0.60      0.65      0.62       686
weighted avg       0.91      0.88      0.90       686


Test Confusion Matrix:
[[151  10]
 [  8   3]]
Test Accuracy: 0.8953

Test Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.94       161
           1       0.23      0.27      0.25        11

    accuracy                           0.90       172
   macro avg       0.59      0.61      0.60       172
weighted avg       0.90      0.90      0.90       172
```

Figure 26: Visualization of Polynomial Kernel SVM decision boundary with PCA.. Note: only two components were selected in this case, as compared to seven, in order to visualize in a 2D plane.



Summary of Further Tuning of Hyperparameters

Figure 27: Summary of Further Tuning of Hyperparameters

	No Additional Dimensionality Reduction		**LDA (1 component)**		**PCA (7 components)**	
Model	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy	Training Accuracy	Test Accuracy
SVM RBF (with further fine tuning)	0.9636	0.9535	0.9636	0.9419	0.7959	0.8314
SVM Polynomial	0.9767	0.9477	0.9679	0.9477	0.8848	0.8953

The RBF kernel SVM model with no additional dimensionality reduction resulted in the highest test accuracy overall. With LDA and PCA reduced data, the polynomial SVM model performed better. With no additional dimensionality reduction, the polynomial SVM also showed higher accuracy on training data. However, higher training accuracy and lower test accuracy means that the polynomial model may be prone to overfitting. Therefore, the kernel RBF SVM model appears to be the best choice of this dataset.

Since the kernel RBF SVM model obtained an accuracy greater than 95 percent, combining models was not deemed necessary.

Further Work

To improve on the results seen with this dataset, it may be useful to explore deep learning methods, for example, neural networks. Neural networks may help uncover patterns in the dataset that are not as clear when utilizing SVM. This may also help improve the false positive and false negative rates of the model, as those metrics are important when predicting cancer risk.

Although LDA and PCA dimensionality did not improve accuracy in the models tested, it is worth testing additional methods for dimensionality reduction in the future, as there are several features in the dataset that measure similar patient characteristics.

Currently this model performs classification, but in the future, it may also be useful to calculate a risk score as the target variable. This task must be run in an unsupervised manner, as risk score is not an available feature in the dataset. However, when predicting risk for cancer, it would be better to provide a risk score, rather than deeming a patient as high risk vs. not high risk.