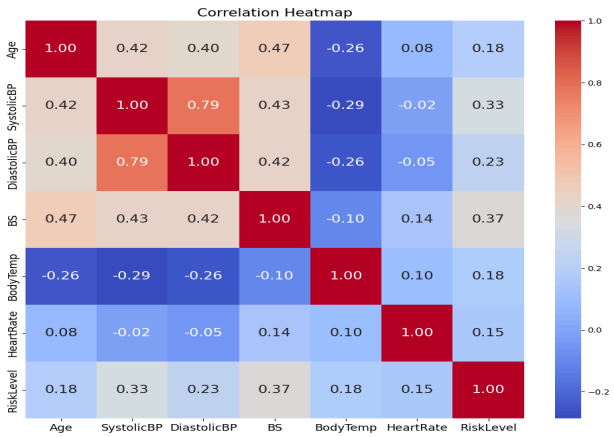
Project Report 7

Introduction

This project aims to create a Logistic Regression and Support Vector Machines with different kernels named: linear, polynomial, and RBF. The datasets used for this project are related to maternal health risk assessment by considering 6 factors such as Age, systolicBP, diastolicBP, BP, etc, and a target column named RiskLevel with 1014 samples, and all of the features are numerical and apple quality with 4001 samples and 7 numerical features such as A_id, Size, Weight, Sweetness, Juiciness, etc and a target column named quality. The stakeholders of the analysis (maternal dataset) are doctors to help them with diagnosis and patients to have a quick check-up of their situations and stakeholders of apple quality are farmers to help them group their apple products and consumers in buying. The target of the maternal dataset has 3 classes that we converted into 2 and that is we assumed mid and high-risk as one class (1) and low-risk as another (0) and for apple quality we had good and bad which we converted them into 1 and 0. For the first part, we looked over the SVM with kernel linear decision boundary of diastolicBP and systolicBP, and the Logistic Regression of BS and systolicBP which we found -6.86 for *β0, 0.02 for β1* and 0.51 for *β2*. For the second part, we used logistic regression on Ripeness and Juiciness and we found 0.01 for *β0, -0.29 for β1,* and 0.27 for *β2*. Also, we found SVM with linear kernel the best for both datasets but for the first analysis we had to reduce class weights class 0 (because class 1 is more important than 0) to operate slightly better among all other models with precision: 0.78, recall: 0.80, and f-score: 0.79 on the class label 1 and on next dataset we got precision: 0.75, recall: 0.76, and f-score: 0.75 on the class label 1. Links to GitHub and presentation.

Dataset

The dataset used for the first analysis is related to maternal health risk assessment and has numerical 6 features named: Age, SystolicBP[1], DiastolicBP, BS[2], BodyTemp, and HeartRate, and a target column named RiskLevel which has 3 unique values named low risk, mid risk, and high risk which for simplicity we combine and convert mid and high risk to 1 and low risk to 0. These features provide us with useful information for creating the models. In the following figure, it is obvious that systolic and BS have the most impact on the target column. We used all the



[1] BP: blood pressure
[2] Blood sugar

features here except HeartRate which had the least correlation with the target column. These columns have a good potential to create our predictive models such as logistic regression, and SVMs with different kernels (linear, polynomial, RBF). This dataset is pretty clean with no NaN values and it has 1014 rows and 7 columns which are mentioned above.

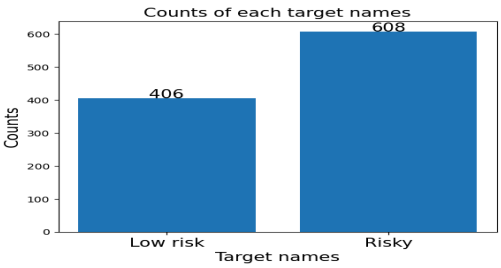
Analysis technique

First part: The analysis employed various techniques tailored to the dataset's characteristics and objectives. Initially, a bar chart visualized the number of different target class labels. Feature selection identified the top five correlated features. Pairwise analysis, like SVM with a linear kernel for diastolicBP and systolicBP, captured complex relationships and performance metrics. Logistic regression estimated coefficients for BS and Systolic and performance metrics with 80% training data and 20% test data. Multiple models, including logistic regression and SVMs with different kernels (linear, polynomial, RBF), were created by 80% of training data and evaluated by 20% of the test data using F1-score, precision, and recall. Class weights with values 0.9 for class label 0 and 1 for class label 1 were applied and then checked the performance of that. Moreover, we looked at the execution time of every model.

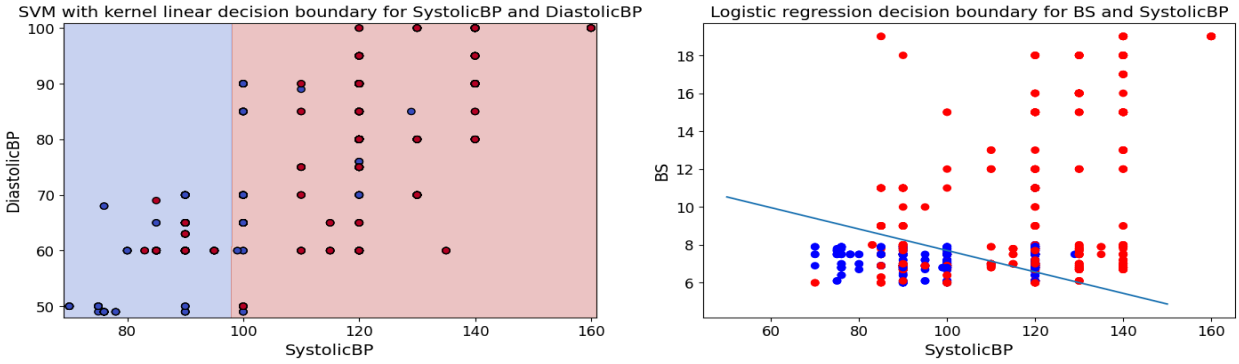
Second part: To analyze how well each of the categories was doing we first decided to look at logistic regression. We selected the top two items with the highest correlation with Quality based on the correlation plot. We came down to using Ripeness and Juiciness since those two attributes were the highest. Analyzing models happened by creating a model with our data. We then wanted to compare its performance with SVM (linear) by using all the attributes and creating with 80% train and test with 20% test, so we plugged the data into an SVM (linear kernel) machine and compared the results to find which one was better. After creating the model we found precision, recall, and f-score. For logistic regression, we can look at the decision boundary and metrics to decide how effective it is. Also, in the end, we provided the confusion matrix of our SVM (linear) to show how well it worked. It should be mentioned that we convert the class label values of to 0 (bad quality) and 1 (good quality) and here 1 has priority for us. Also, for creating the SVM (linear) we standardized the features. They were without class weights.

Result

First, as provided in the following figure, we looked at the count of different target values in the target column and we found 406 for low risk and 608 for mid and high risk.



In the following figure, we selected diastolicBP and systolicBP, and BS and systolicBP as the pairs that have the highest correlation with the target column. The left one is SVM with a polynomial kernel and with equal class weights. The right one is created with logistic regression based on the formula *BS = -(β0 + β1xSystolicBP) / β2*. We got β0 of -6.86, β1 of 0.02, and β2 of 0.51 then drew the line. The red points mean class label 1 and the blue means 0. The precision, recall, and f-score for (DiastolicBP & SystolicBP) are 0.67, 0.85, and 0.75 with SVM with linear kernel, and for (BS & SystolicBP) would be 0.76, 0.77, and 0.76 for logistic regression (on class label 1).



As our final analysis for this part, we created different models and analyzed them based on precision, recall, and f1-score. Also, we applied class weights to them 0.9 for class label 0 and 1 for 1 and here detection of class label 1 is the most important thing because of mid and risk therefore, we reduced the weight for class label 0. The Results of each of them are provided in the following table:

	Logistic regression		SVM (linear)		SVM (linear with class weights)		SVM (poly)		SVM (poly with class weights)		SVM (RBF)		SVM (RBF with class weights)	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
precision	0.63	0.82	0.61	0.85	0.68	0.78	0.60	0.82	0.65	0.78	0.63	0.72	0.72	0.65
recall	0.77	0.70	0.82	0.65	0.66	0.80	0.78	0.66	0.67	0.77	0.53	0.79	0.22	0.94
f1-score	0.69	0.76	0.70	0.74	0.67	0.79	0.68	0.73	0.66	0.77	0.58	0.75	0.34	0.77

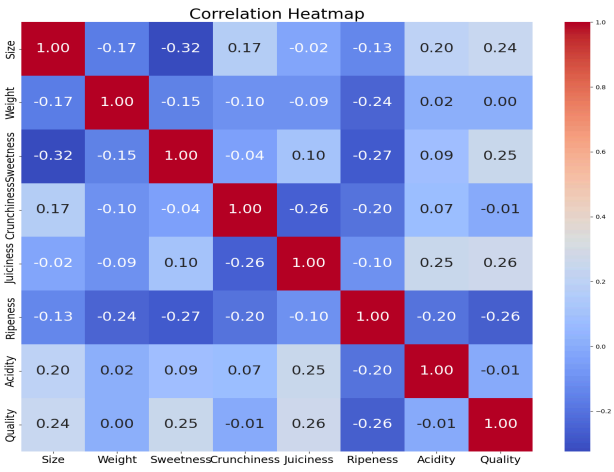
Because in high-bias models we would have higher precision and lower recall and in high-variance is reversed we should select the model with a good tradeoff, it looks like the SVM with a linear kernel with applying class weights is better than the rest because we have a good trade-off between precision, recall, and f1-score. Also, the execution times of different models were likely the same (0.0, ~0.2).

PART 2

Dataset

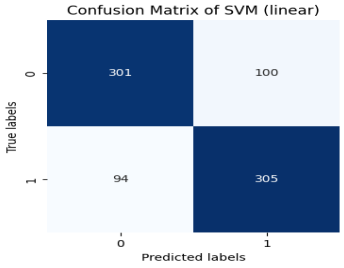
This dataset is made up of 4001 (4000 after removing NaN values) different apples all of the same types. They contain information about various attributes of a set of fruits,

providing insights into their characteristics. The dataset information such as fruit ID, size, weight, sweetness, crunchiness, juiciness, ripeness, acidity, and quality. This dataset can be used as a classification model to categorize fruits based on their features. It can also be used to build a model to predict the quality rating of fruits using various attributes. Some data had to be cleaned up into integer values and null values had to be dropped entirely. We also removed the IDs since they didn't serve any significance to us. As we see in the following correlated figure, ripeness, juiciness, sweetness, and size are the most correlated ones. Also, the Acidity column is converted as float type and the target variables good and bad are converted to 1 and 0 all of the columns are numerical.



Results

After looking at the heat map (provided in the dataset section) and deciding to go with the Juiciness and the Ripeness we immediately plugged those values into our logistic regression machine to find our results. The results don't seem to be very good (maybe because of not using enough features). Our precision values were 0.65 and 0.65, our recall was 0.66 and 0.64 and our f-score was 0.65 and 0.65 on class labels 0 and 1 which is not perfect! Looking at it further we looked to see where between these two values it would decide to put our decision boundary. At first, we couldn't see it but after playing around with the window a bit we were able to draw it out the only issue with this is now when we went to plot our decision boundary it wasn't the strongest. The decision boundary looks good but as you can see there are a lot of miss classifications. After viewing this we concluded that logistic regression wasn't very strong with this graph so it was time to move on and view what SVM (linear) would do for us. Immediately we get an f1-score back of 0.75 which is much better than before. Then a recall of 0.76 and a precision of 0.75. When we then graphed our confusion matrix and gave it the values we trained our model on we were left with this. As we see the number of true predicted 0 is 301 and the number of true predicted 1 is 305. 100 items predicted 0 which is false and 94 items predicted 1 which is false again. In the aspect of execution time, both models were the same (0.0, ~ 0.3)[3].



[3] Here the execution time mean time spent per cell in jupyter notebook.