Projects in Data Science

https://github.com/manateem/2025-FYP-groupManateem

Laci Gáspári, Valantis Koumkoumis, Matthew Rangstrup, Hjalte Erlang Treebak, Martin Mečiar

Contents

1 Abstract

Cancer is an ever-present threat in our lives, and being able to mitigate the risk of serious complications, it is best to stop it dead in its tracks. The PAD-UFES organization is a non-profit that runs a dermatological and surgical assistance program at a Brazilian university. Using data collected by PAD-UFES, containing hundreds of skin lesions, we can indicate whether a particular lesion shows signs of malignancy. This data shows great potential for extraction of many features that could all be used to train a classifier. However, the exploration in this report is focused on whether it is best to use many, if not all these features to help improve classification of skin cancer, as warns the curse of dimensionality. We aimed to implement as many features as possible in a model, then comparing this result to a baseline and reduced model. With this we hope to identify how far we can push our model with features, before the curse of dimensionality hits and negatively affects the performance of the model and its outcomes, highlighting the importance of carefully selecting which features to use. What our study showed is that a reduced final classifier model has the same good performance in precision, accuracy, and recall as larger feature models, while the ultra-classifier with 33 features was doing extremely well with internal data but most likely overfits due to a reduced dataset and bias potential. While our findings haven't shown much evidence of the curse of dimensionality, we are sure that they can still spark discussion that will further improve our understanding of how machine learning classifiers function. This can then help similar models avoid the curse of dimensionality and maximize the potential performance and accuracy of results without overfitting.

2 Introduction

Skin cancer is the 14th most common type of cancer, with 331,722 new cases reported in 2022 alone ("Skin cancer statistics", n.d.).

Fortunately, early detection can significantly increase the chances of successful treatment. Therefore, various methods have been developed to detect skin cancer in skin lesions. Machine learning models have shown to be very effective, as with the right data and training parameters, they can rapidly inspect multiple images of lesions and spot patterns that are invisible to the human eye. However, even when trained on identical datasets, the performance of these models differs widely. One of the main reasons for this difference in performance is the choice of the classifier's features. Working with image data opens us up to a large number of potential features, which can broaden choice and usage, dramatically influencing performance. With so many features to choose from, a common mistake when designing a classifier for a

model, is the inclusion of too many features. This can cause inconsistent decision making and reduce the performance of your model in many ways. This is known as 'the curse of dimensionality'. As you add more features, the data becomes scattered in a much higher-dimensional space. This makes it harder for the model to find useful patterns. Another issue that arises is that the model will start to 'memorize' the training data instead of learning general patterns. This is called 'overfitting'. When a model overfits, it tailors itself to the training data but performs poorly on new and unseen data. Apart from that, not all features are informative. Some of them are not helpful at all and can even divert the model's attention, causing it to detect patterns that are more hurtful than beneficial. For this reason, one should carefully choose the features to use. In hopes of further understanding this phenomenon, this report investigates the question: "When designing a classifier for detecting skin cancer in skin lesions, when does the curse of dimensionality appear, and how do the amount of features affect the final model?"

3 Data Exploration

The dataset we are given is the PAD-UFES-20 dataset (A. G. Pacheco et al., 2020). It contains 2298 close-up images of various skin lesions from the year 2020. There are 1373 different patients and 1641 skin lesions, labeled with 6 different types of skin lesions and 3 types of skin cancer. UFES is a university in Brazil working a non-profit organization PAD-UFES, a dermatological and surgical assistance program. The state they live in has a high incidence of skin cancer, but the majority of people live in rural or remote areas and can't afford private treatment. Thus, a system was developed to track and collect data, then used to create a dataset and algorithms to assist in detection, resulting in the dataset we have today (A. Pacheco, n.d.).

The images contain hair to various degrees, ranging from having no hair at all to having so much hair that even the detection of the skin lesion's edges is difficult. Therefore, the first step we must take is to remove the hair using inpainting. For the detection of hair, we convert the image to grayscale, and then apply the black hat filter for the detection of dark pixels against a white background. This gives us a mask, which we can use to in-paint the images to remove hairs. During hair removal, we can already extract the first feature: the amount of hair.

However, there are some challenges with the data. Sometimes, hair is so thick the lesion is barely seen. Sometimes the lesion has been circled in marker (usually blue, sometimes black), which would be a huge problem when attempting to threshold to create masks. Thankfully, we were provided with a set of masks from students of previous years' similar projects.

Some images are unbelievably low quality, which may affect training data by being imprecise but also helps increase data diversity. Sometimes, the lesions are multiple and occur in various locations, in a reddish tone or a darker tone around the main lesion. Finally, there can be duplicates of the same image. Sometimes, they are exactly the same photo, but categorized as different lesions, and sometimes the same photo but zoomed in. In other images, they are the same lesion but taken at different times and angles. In all cases, we wish to delete the exact duplicates or choose only one instance of a lesion (based on the lesion id) from every patient.

Along with our data, we were given a metadata.csv file, which provided the 'biopsed' label paired with the patient ID label necessary for our cancer classification. Additionally, a variety of labels could be added as features to a possible mega-classifier. However, we soon realize that there are a lot of missing values in certain fields. In some cases, one group of labels had missing or NaN values, while the other group of labels were filled. In other cases, it was the reverse, as if these two groups of labels were mutually exclusive. Even worse, for the data which had one group of labels missing, they were all biopsed as true, while for the other data which had the other group of labels missing, they were all biopsed as false. However, in around 1100 lesion instances, both groups of labels were filled. Here, we were forced to make a decision of which data we wished to use, as we decided it was unthinkable to fill in or train using the empty labels in some way due to the aforementioned bias and the various classifiers we planned to use. In the end, we decided to pick the 1100 lesion instances which had both groups of labels filled. We sacrificed the size of our sample data in favor of deeper exploring the curse of dimensionality with more features.

The features we extract from the images are the

following:

1. Rotational Asymmetry, found by rotating and folding the image
2. Compactness, calculating by using the perimeter and area of the lesion
3. Multicolor Rate, found by utilizing KNN on the different colors found in a lesion
4. Color Uniformity, found by getting the variance of every pixel to the mean color of the lesion
5. Average Lesion Color
6. Brightest/Darkest Pixel
7. Average Max Redness, found by averaging the largest 0.1% of red values
8. Convexity Score, aka Border Solidity
9. Convexity Variance, Average, Maximum; of all the convexes
10. GLCM (Gray-Level Co-occurrence Matrix) Contrast, Energy, and Homogeneity
11. Ratio of Hair to Image

We acknowledge that some features were inspired by another group working on the same assignment. (Jensen, 2025)
This will be used in combination with certain features from our metadata. We also find that some lesions do not have masks, so we decided to remove them from the data instead of trying to create our own masks for the sake of time and resources. We acknowledge the possibility of introducing bias by doing this. It is likely that masks are missing because of their lesion's difficulty to mask by previous years' students. This could be due to extremely similar skin colors, being extremely small, broken up, or spread out, low image resolution, or significant amounts of hair blocking the way. Thus, our data diversity becomes more restricted to lesions that are easier to see and mask and our classifiers become less general.

4 Methods

In order to analyze the effect of the curse of dimensionality on different classifiers, we narrow our scope to three different classifiers, with the goal of understanding and comparing their performance when given a large amount of features. The three classifying methods we have decided on are K-nearest-neighbors (with K always being 5), decision trees and logistic regression. We chose these three because of how easy they are to implement and compare, making it easier to form and examine our hypotheses.

As a baseline, we will train a basic model running on the classic ABCDE rule (Asymmetry, Border irregularity, Color variation, Diameter more than 6mm, and Evolving) which is an adequate general rule for signs of malignant melanoma (Daniel Jensen and Elewski, 2015). It is easier to implement some of these features than others, as we do not have adequate data to reliably obtain feature values for diameter and evolving, but the ABC part of the mnemonic can be found through features of the lesion images.

Next, we wish to load as many features as is possible into an "ultra-classifier". We will take both from the metadata provided from our data, and a multitude of features extracted from lesion images. Additionally, we wish to include some noise features to achieve a high number of features. This ultra-classifier is created with the intention of setting into motion the curse of dimensionality. However, doing this limits our data set to only 1100 images, as mentioned previously mentioned as a limitation.

Following that, we will tone down our number of features as to work in tandem with an external dataset to test (maintaining our noise features). This external dataset will not have all the same metadata and so we must ditch a significant amount of features in order to be compatible, relying heavily on the features of the lesion images. Using an external dataset will help account for any sort of overfitting that may arise from using the same internal dataset. This compatible group of classifiers will be referred to as the "mega-classifier" set. The dataset in use is a small section of 100 images taken from the HAM10000 by the Harvard Dataverse. Tschandl, 2018.

Finally, we will create the final, reduced classifier. It will be a reduced model where we select only the most significant of features in classifying malignant lesions, meaning we ditch features that are meaningless or have insignificant effects in classification. The same external dataset will be used to test this classifier as well.

Models will be evaluated with Group K-fold cross-validation.

For our ABC classifier, we choose: Rotational *Asymmetry*, Compactness for *Border*, Multicolor Rate for *Color* as our features.

For our ultra-classifiers, we use the metadata from our metadata.csv file, and the following image features that we extract: Rotational Asymmetry, Compactness, Multicolor Rate, Color Uniformity, Average Color, Brightest Pixel, Darkest Pixel, Average Max Redness, Convexity Score, Convexity Variance, Convexity Average, Convexity Maximum, Ratio of Hair to Image, GLCM Contrast, Energy, and Homogeneity. Furthermore, we add some noise features.

For our image mega-classifier, working in tandem with an external dataset for testing, we simply exclude the metadata from above. We will, however, also include any metadata properties that our external dataset shares with our internal dataset.

For our reduced final classifier, we focus solely on a select few performance-based features garnered by analyzing the cross-validation metrics using Group-K-Fold, working in tandem with an external dataset for testing.

Before running the experiments, we establish a set of hypotheses based on theory and external related work, in order to have a baseline from which we can discuss and evaluate our final results. Specifically, we will focus on what happens to the three classifiers below as we increase the number of features.

4.0.1 K-Nearest-Neighbors

The K-Nearest-Neighbors model works by classifying a data point based on the majority label of its K closest neighbors. This means that the addition of a new feature subsequently adds a new dimension that KNN needs to account for. At higher dimensions, distance values start to converge to a similar value, making the actual distance metrics less and less distinguishable from each other.

4.0.2 Decision trees

In contrast to K-Nearest-Neighbors, decision trees do not rely on distance metrics to classify data. Instead, decision trees work by recursively splitting the data, essentially making a giant flowchart in the form of a (usually binary) tree. When classifying an image, it then runs the image through the flowchart until it reaches a leaf, where a probability is then outputted. The data is split by an impurity score, which in our case is the Gini impurity, which is a way to estimate how informative a feature is.

4.0.3 Logistic Regression

As for Logistic Regression, it is a straightforward and widely used method for classifying data into two groups. It works by combining the input features with certain weights to calculate a score, which is then transformed using a sigmoid function to give a probability between 0 and 1. These weights tell you how much each feature influences the outcome. To keep things from getting too complicated or overfitting, it applies a regularization technique (default L2) that keeps the weights from getting too large. Overall, Logistic Regression is a solid choice when the connection between the features and the result is mostly linear in terms of odds.

5 Hypotheses

Before beginning our experiments, we have several assumptions and hypotheses for their outcome. First, we hypothesize that our baseline ABC classifier will perform moderately well. It should be able to perform adequately whilst not overfitting and should run quickly, due to the low number of features. As a baseline among all our other classifiers, we think it will perform the 2nd best, beaten only by our reduced final model.

Next, there are conflicting perspectives on our ultra-model, which includes both metadata and all our image features. Some members in our group believe it will perform horribly when put up to testing data or any data it has never seen before. During training, its accuracy will be high, but that is due to extreme overfitting to the particular data that it was shown. It will test and train relatively slower and is likely to be the worst among all our models. However, other members believe it will perform well when put up to the testing data because it will actually be able to find some valid pattern on our data or because it will overfit to the internal dataset because of common factors in the internal data not relevant to the skin lesion.

Following that, our image mega-classifiers, which will only include image features (and any metadata that it may have in common with the external dataset), will have moderately fewer features, but will likely still overfit and perform

poorly, due to many seemingly irrelevant, overlapping, or redundant features that have been added. However, it will perform better than the mega-model mentioned before because of a reduced feature space.

Finally, with our reduced final model, we will focus solely on the features that are significant and relevant to the classification. We hypothesize that this will be our best performing model, both in internal testing data and external testing data, when compared to our other models.

For our classifier types, KNN, Decision Trees, and Logistic Regression, we believe:

1. The accuracy of a KNN classifier will initially improve with the addition of relevant features, but will eventually decrease or plateau when the feature space reaches a certain size.

2. Decision trees will do well with a lot of irrelevant features added to the model, but will struggle and overfit when more relevant features are added. This is because since decision trees prioritize the information given by a feature, we believe that it would handle larger feature spaces quite well, as long as the meaningful features don't overlap too much. Irrelevant features would very rarely be reached, and thus, they will not be able to influence the final decision. However, many features overlapping could lead to overfitting, since the decision tree will get too much data to choose from, and possibly generate a tree that is way too specifically tailored to the training data.

3. Logistic regression is expected to have better accuracy but is more likely to overfit to the data, especially when using many correlated features even though we are trying to reduce overfitting by using a regularization technique that shrinks the coefficients. It is also likely to perform much faster than something like KNN because of the less complex calculations it must make.

6 Process

Along the way, we made some realizations regarding code efficiency, displaying the importance of efficient code, shortening the ETA from 6 hours to 2 minutes in some cases. Upon further inspecting our external dataset, we find that it's metadata shares nothing in common with our internal dataset's metadata. In this case, our image mega-classifiers and reduced final classifiers will focus solely on image features.

We started validating our image mega-classifiers using GroupKFold with $K = 5$ for all 16 features that we extracted from the images. In doing so, we received information for the accuracy, precision and recall rates for our model. In addition, we checked for highly correlated feature pairs and also got the coefficients and p-values for the Logistic Regression and Gini Feature Importance for the Decision Tree. While evaluating those parameters, we started removing features one by one, starting with those that had a high correlation and a large p-value and/or a low Gini importance, indicated low significance for our model. For example, at first, we had 3 pairs with a really high correlation (convexity and asymmetry with a correlation of 0.92; glcm_energy & glcm_homogeneity with a correlation of 0.98; averageMaxRedness and maxBrightness with a correlation of 0.88), so we removed energy that had a high p-value and a low Gini importance, also citing the importance of homogeneity when considering a lesion. We followed this procedure with the aim of keeping the accuracy, precision, and recall as high as possible. Always keeping in mind that our model needs to remain generalized and not overfit with our training data so it would perform well on unseen test data, but also in entirely new external data. For that reason, we tried to keep at least one feature of each of the basic ABC features.

Eventually, we settled on four features: asymmetry, border irregularity, glcm_homogeneity, and color uniformity. Three of these proved to be highly significant for our models, apart from asymmetry which while it did not look to be as important, we kept it to ensure that our model includes the "A" from ABC, keeping in this way all the core dermatological criteria. Near the end, due to time constraints, we found that it was also infeasible to try to integrate random features. As a continuation of this project, we could use random features to simulate the effect that completely redundant features have on the different classifying methods.

7 Results

Our results differ from our initial hypotheses on many occasions. For one, we do not see the amount of difference in our validation set and our external set as we thought we would. We can notice that our model with three and four image features performs quite well in accuracy, precision, and recall and maintains similar performance when using all 16 image features, apart from Logistic Regression, which appears to drop slightly in performance (accuracy, precision, recall). Finally, adding the features from the metadata file (for a total of 33 features) makes the model perform astonishingly well, which could be due to overfitting but also because we retained only the entries that had complete information for all features, reducing our dataset from 2,300 entries to around 1,100. We removed specific entries because, as mentioned previously, approximately 800 samples lacked any data regarding smoking, drinking, etc. and were all biopsied as FALSE. Additionally, another 400 samples were missing details regarding changes in lesions, pain, or growth, and were all confirmed as biopsied TRUE.

We believed that trying to fill in missing values by assigning 0.5 to them would introduce further bias. However, by doing so, we removed too many FALSE biopsied entries, resulting in a model trained on all 33 features having far more cancerous lesions than non-malignant ones, something also reflected in the significance matrices.

This leads to an extremely impressive ultra-model as shown in Figure 4 and Figure 5, that is most certainly over-fitted completely to the given data. Whether the number of features has an effect on this, or if it is only because of the reduced training data, we do not know for sure. If only the external data that we had, also had the relevant metadata, we would be able to confirm whether or not this model or these classifiers are extremely performant or simply over-fitted.

In Figure 2, we see strange behavior when our logistic regression model evaluates the external dataset. This might indicate some errors in the way we either trained the models or extracted the features for the external dataset, or simply differences between the two datasets that we did not account for properly (reprocessing, perhaps). Strangely, the other models work somewhat as expected from the cross-validation results.

In the end, the results were less than satisfying. Many models and classifiers achieved results that were essentially close to coin flips, and models that did perform well we believe to be most likely overfitted. In order to come to more concrete conclusions, we would need more specifically modeled data on the performance of the different classifiers when predicting the external dataset. Due to time constraints, the results were acquired in less than ideal circumstances, and therefore fail to provide the information we would need to confidently tackle our initial hypotheses.

8 Discussion

While this report did not make general progress on the topic of when the ”curse” appears, other more qualified people have succeeded. In researching this, we found the study ”The peaking phenomenon in the presence of feature-selection” by Chao Sima and Edward R. Dougherty. (Sima and Dougherty, 2008)
This study investigates a similar phenomenon, called the peaking phenomenon, stating that for a fixed sample size, the error of a classifier decreases and then increases as the number of features grows.
The method in which the hypotheses are tested in this study is, like this report, through multiple simulations. Although it is clear that the tests are a lot more structured and well thought-out. Reading through this study and other research on the topic made us realize why this phenomenon is sometimes referred to as a curse. Research regarding the topics quickly exceed our current knowledge and understanding of machine learning, leaving us more confused than before reading. A reason we believe for why our results are not as we expected, is because we didn't have enough features for the sample size we had in our ultra-classifiers' and mega-classifiers' training. Because of these lack of features, the 'curse of dimensionality' did not strike. Although our results may not have been as we had hoped, we are sure that they can still spark discussion that will further improve our understanding of how machine learning classifiers function.

References

Daniel Jensen, J., & Elewski, B. E. (2015). The ABCDEF Rule: Combining the "ABCDE Rule" and the "Ugly Duckling Sign" in an Effort to Improve Patient Self-Screening Examinations. *The Journal of Clinical and Aesthetic Dermatology*, *8*(2), 15. Retrieved May 28, 2025, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4345927/

Jensen, M. (2025, May). BossThePro/2025-FYP-groupKangaroo [original-date: 2025-03-26T09:34:58Z]. Retrieved May 30, 2025, from https://github.com/BossThePro/2025-FYP-groupKangaroo

Pacheco, A. (n.d.). Datasets through the looking glass- s03e02 - dr. andre pacheco. Retrieved May 28, 2025, from https://www.youtube.com/watch?v=q-DBwWZejMY

Pacheco, A. G., Lima, G. R., Salomão, A. S., Krohling, B., Biral, I. P., de Angelo, G. G., Jr, F. C. A., Esgario, J. G., Simora, A. C., Castro, P. B., Rodrigues, F. B., Frasson, P. H., Krohling, R. A., Knidel, H., Santos, M. C., do Espírito Santo, R. B., Macedo, T. L., Canuto, T. R., & de Barros, L. F. (2020). Pad-ufes-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Data in Brief*, *32*, 106221. https://doi.org/10.1016/j.dib.2020.106221

Sima, C., & Dougherty, E. R. (2008). The peaking phenomenon in the presence of feature-selection. *Pattern Recognition Letters*, *29*(11), 1667–1674. https://doi.org/https://doi.org/10.1016/j.patrec.2008.04.010

Skin cancer statistics. (n.d.). Retrieved May 26, 2025, from https://www.wcrf.org/preventing-cancer/cancer-statistics/skin-cancer-statistics/

Tschandl, P. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. https://doi.org/10.7910/DVN/DBW86T

9 Figures

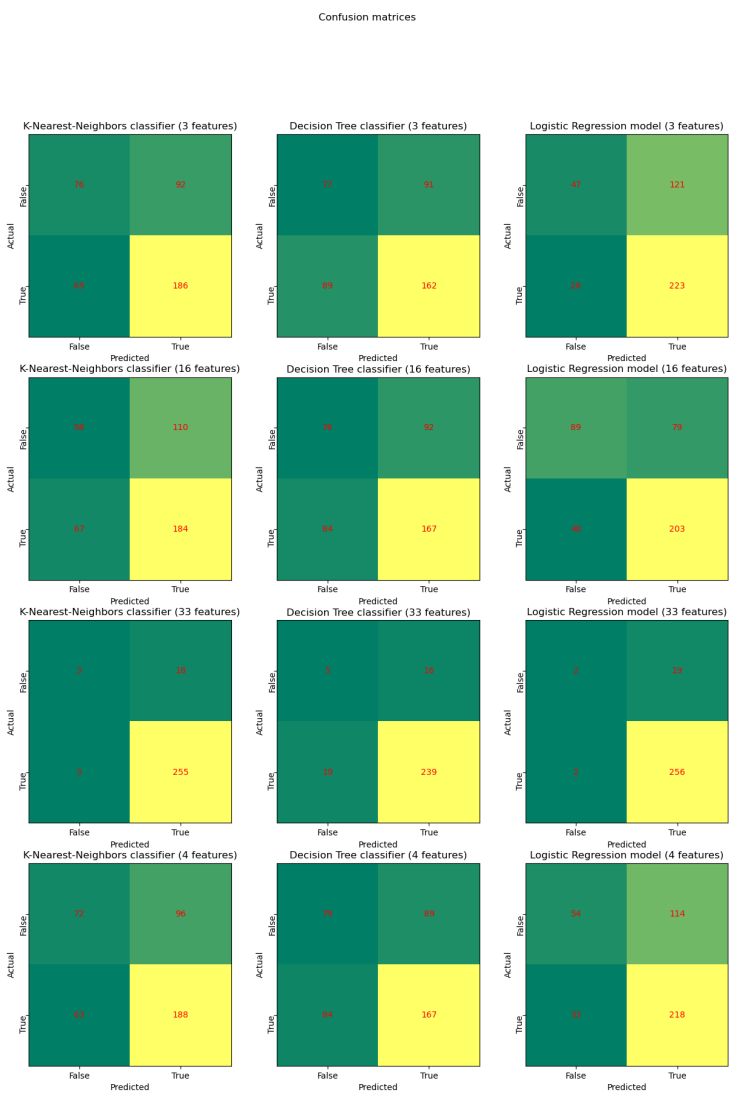

Figure 1: Confusion matrices for cross validation

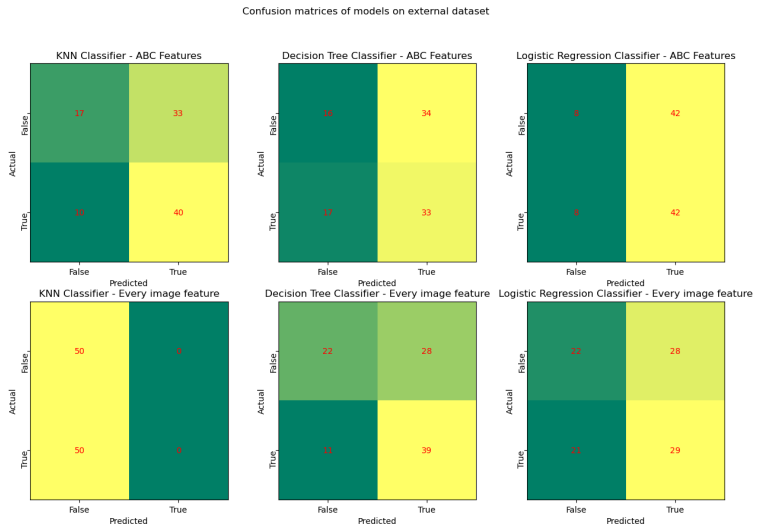

Figure 2: Confusion Matrices for Performance on The External Dataset (Note that the bottom left figure is not working as intended because of a bug.)

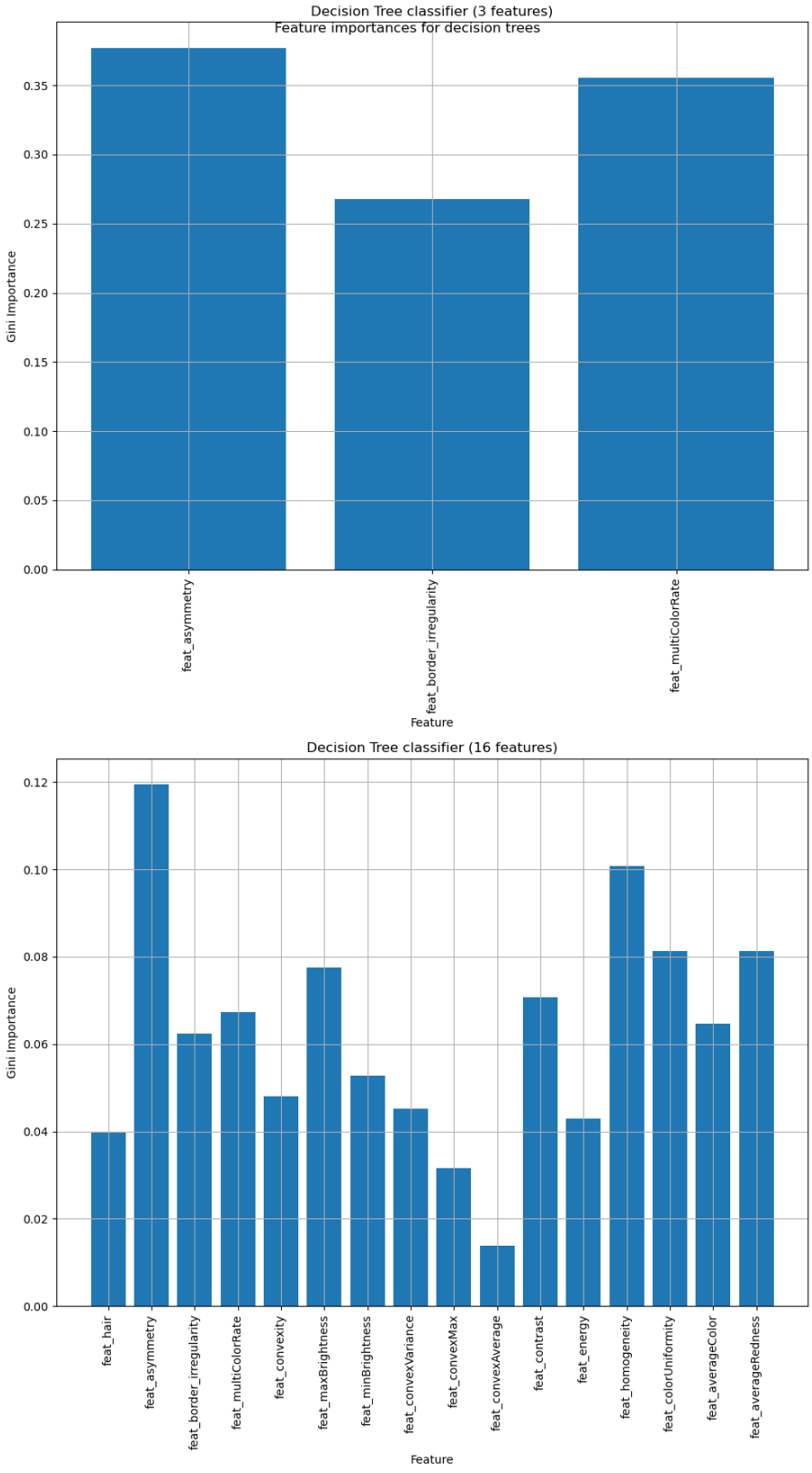

Figure 3: Feature Importance for Decision Trees

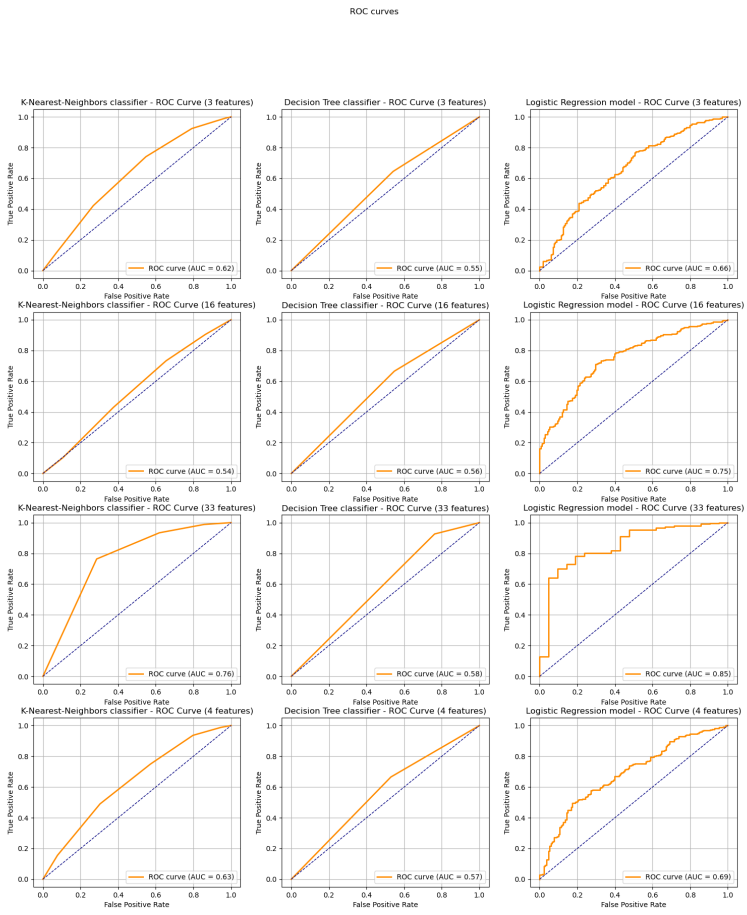

Figure 4: ROC curves for Cross-validation Results

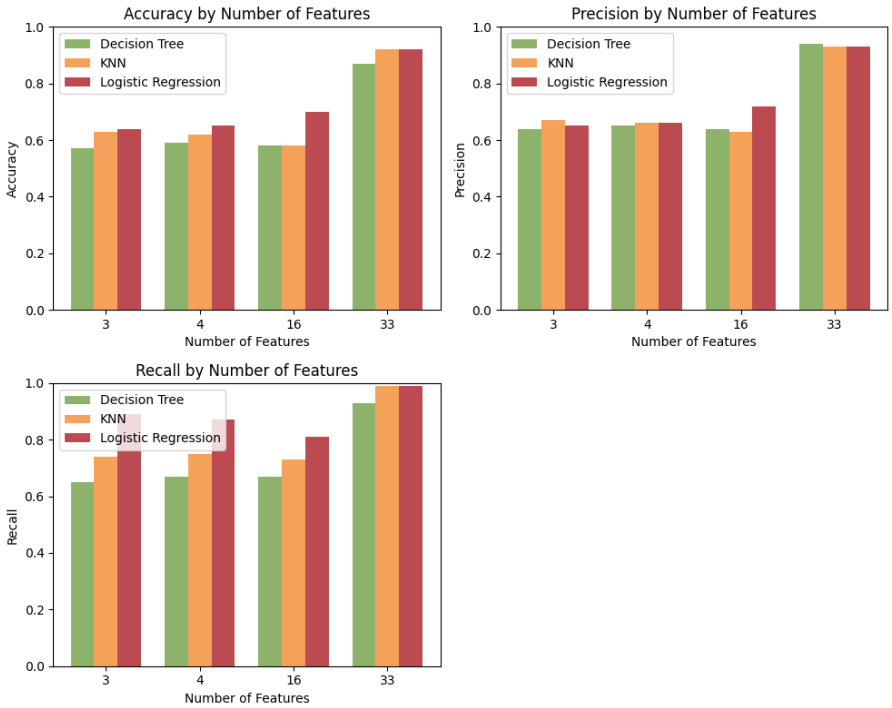

Figure 5: Model Performance on Test Data