Gender Recognition of Face Images

Ramya Nivedha Raja Michael Ani Ta'Destiny Geiger

Georgia Institute of Technology
College of Science: Mathematics Department

December 10, 2023

Abstract

This research explores the effectiveness of machine learning models in gender recognition based on facial features. The study initially focused on the analysis of six facial features extracted from images in the dataset. Then non-facial features were incorporated, leading to improved model performance. The paper emphasizes the importance of supplementing facial features with relevant non-facial characteristics for enhanced gender recognition accuracy. Through a comprehensive comparative analysis of various models, including Linear Discriminant Analysis, Quadratic Discriminant Analysis, Support Vector Machine, Decision Tree, k-Nearest Neighbors, and Logistic Regression, the study provides insights into the strengths and limitations of different feature sets, contributing to the broader understanding of gender recognition in images.

Keywords:

Gender Recognition, Machine Learning Models, Feature Analysis, Facial Feature Extraction, Model Performance Evaluation

1 Introduction

The fundamental function of gender detection is to differentiate between individuals who identify as male or female based on unique facial characteristics. Studies on gender detection have changed over time as different approaches have been investigated and published in academic journals[4, 13]. Notably, the emphasis has been on classifying human gender

based on various features like the face, eyebrows, hand and body shapes, and fingernails. Nonetheless, because face images are so relevant and effective, the majority of research on gender detection has been conducted using this technique. As technology develops, gender detection through facial image analysis is anticipated to continue deeper and further.

Our aim is to establish a dependable system for the thorough analysis and recognition of facial images by utilizing Python and machine learning techniques. The focus is to identify characteristics unique to a person's gender that are captured in these pictures. With the characteristics, we want to classify and categorize these photos according to distinguishing characteristics that come from the data that is contained in the visual representation. It is crucial to remember that our aim is not to discredit individuals who identify outside of conventional gender roles or those who identify themselves as a gender different from their presentation.

2 Methodology

Tackling this extensive machine learning analysis requires a systematic approach. This involves clearly outlining the problem, selecting a dataset for addressing it, conducting a comprehensive literature review on each machine learning algorithm prior to implementation, organizing and cleaning the dataset for algorithmic application, extracting features from the dataset, implementing relevant machine learning algorithms across various categories, comparing the results, and, based on method comparison, highlighting potential new strategies or approaches for future studies.

While the theory of this process seems simple, the execution took more time than expected. First, the problem was defined to address the need for gender recognition analysis via facial image analysis. Our literature study explored understanding the mathematical formulation behind each method by reading *An Introduction to Statistical Learning: With Applications in R* [3], referring to relevant lectures from Dr. Wenjing Liao's Mathematics of Data Science course at the Georgia Institute of Technology, as well as reading relevant scholarly articles regarding the implementation of each algorithm. Once a solid understanding was formed for each method, we looked to identify a data set to help accomplish the analysis goals.

After organizing and cleaning our data, then extracting features from photos, we implemented our algorithms based on six features extracted from each image in a subset of our data. After gathering and analyzing those results, an additional three features, for a total of nine, were extracted from each image in the data (eyebrow length, eyebrow height, lip size, eye length, round face, face height, lipstick, makeup, and necktie). We wanted to observe if the addition of additional features would affect the analysis and provide different results from the data, through an extensive feature comparison.

2.1 Data

The data set we used for our project is CelebFaces Attributes Dataset [9]. The face images have a broad range of pose variations and background complexities, which guarantees a thorough representation of real-world scenarios. CelebFaces contains 202,599 face images,

10,177 unique identities, and comprehensive annotations with 5 landmark locations and 40 binary attribute annotations per image. The data set is also accompanied with a .csv file with information pertaining to a binary representation of the 40 features per image for all images in the data set.

2.1.1 Feature Selection

From the 40 binary attribute annotations, we used the 'Male' column as our predictor for our data set. While we had this labeled information, the scope of our project aims to extract features from the images to conduct our analysis. Thus we only used the 'Male' attribute to label our data for supervised learning. Further more, it is important to note the values in the .csv file of attributes are represented only as binary values, to conduct a more in depth analysis, our feature extraction sought to represent features as numerical values or decimals. Thus when we conducted our own feature extraction, we incorporated floating point variables to represent the features of each image rather than as binary values.

We then sought to expand analysis further by identifying facial landmarks that play a role in determining the gender of facial images. We wanted to choose the best features to predict gender. Thus, we mapped a correlation matrix based on our given, labeled, attributes to understand which features could serve as the best predictors. Because we specifically wanted to analyze facial attributes, features related to hair color, if wearing accessories, if the image was blurry, attractiveness (subjective) or otherwise cosmetic aspects, were categorized to be added as additional features for further analysis later on. The six facial features chosen for analysis include eyebrow width, eyebrow length, eye length, lip size, face height, and round

face. After the initial implementation was conducted on those six features, an additional three features pertaining to more cosmetic aspects we added for additional analysis (lipstick, makeup, and necktie presence represented as binary values). Please observe the correlation matrix, Figure 8, in the appendix.

2.1.2 Feature Extraction

The feature extraction process in gender detection can be broadly categorized into two main approaches: geometric-based and appearance-based. Geometric-based feature extraction involves the identification and extraction of different facial components or feature points that primarily represent the geometric structure of the face. This approach often relies on fixed points within the face image to derive essential information for gender classification. On the other hand, appearance-based feature extraction takes advantage of the entire face image or specific components within it by applying image filters to extract relevant features. For this project, we will use the geometric-based approach[15].

To extract these facial components, we incorporated a Python package called Dlib. It is a versatile open-source software library primarily written in C++ but with interfaces for Python as well. Dlib is designed to provide tools and algorithms for a variety of machine learning and computer vision tasks. It incorporates a face detector, a trained face key point detector, and a facial recognition model. It is widely known for its high accuracy of approximately 99% percent when it comes to face detection [17]. From the Dlib package, we extracted first six, then nine, features using facial estimation and coordinates, for example in Figure 1. The nine features include eyebrow length, eyebrow height, lip size, eye length,

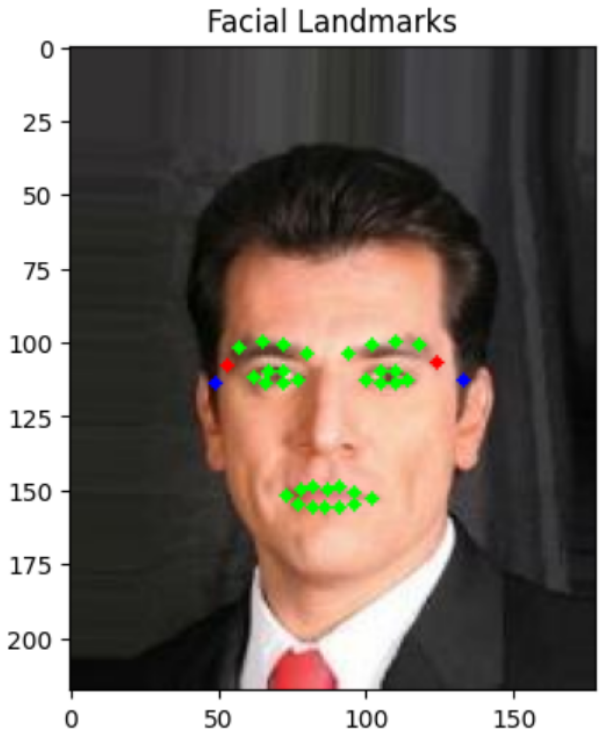


Figure 1: An example on the Dlib package with an arbitrary image in the Celeba Dataset

round face, face height, represented as floating point decimals, and lipstick, makeup, and necktie presence, represented as binary values. The definitions for these features are in Table 1.

2.1.3 Data-Splitting

In the CelebA Face Attributes Dataset, a notable class imbalance exists, with a considerably higher number of female images compared to male images. This posed challenges when we opted to split the dataset into training (80%) and test (20%) sets, particularly impacting the analysis of gender detection. An issue arose during our evaluation process where, despite obtaining an accuracy score for all predictions, the sensitivity metric yielded a 0% result, meaning the model was unable to correctly identify any positive instances. No sensitivity

Feature	Definition
Eyebrow Length	Distance between the innermost and outermost points of one eyebrow
Eyebrow Height	Distance between the highest and lowest points of one eyebrow
Lip Size	Distance between the two corners of the mouth
Eye Length	Distance between the inner and outer corners of the eye
Round Face	Binary indication of a relatively balanced width-to-height ratio
Face Height	Distance between the top of face to chin
Lipstick	Binary indication of the presence of lipstick around the mouth.
Makeup	Binary indication of the presence of makeup around the nose.
Necktie	Binary indication of the presence of necktie around face outline and neck.

Table 1: Definition of the Features

occurs when the model has absolutely no ability to recognize the positive class. In response to this phenomenon, we discovered that the CelebA dataset includes a partition column, categorizing images into training (1), validation (2), and test (3) sets.

To address the class imbalance and enhance the robustness of our gender detection model, we strategically curated our training dataset using a partition method, where the data was intentionally split into even testing and training sets [10]. Specifically, we selected 800 images, ensuring an equal distribution between female and male images, all sourced from the training partition set. For the testing set, comprising 700 images, we maintained a balanced gender representation, with half being female and half male images. Some images were sourced from the validation partition, while others were from the testing partition. This partitioning and selection process aimed to mitigate the challenges posed by class imbalance and improve the overall performance and reliability of our gender detection analysis on the CelebA dataset.

3 Machine Learning Algorithm Formulation

A variety of methods were chosen for analysis to provide a comprehensive understanding of the effectiveness of classification, dimensionality reduction, and clustering methods.

3.1 Classification Methods

Of the several categories of machine learning, classification falls under the category of supervised learning where the aim is to predict the correct labels of new cases based on prior observations. With classification methods, the respective machine learning algorithm is trained on a data set of labeled values. The main goal of classification methods are to use the trained data to learn and see if the algorithm can reproduce the results on portion of the data where labels have been removed. A few methods identified to aid in the analysis with gender recognition of face images include Logistic Regression, K Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Trees. Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) have also been chosen, but are classified as both classification and dimensionality reduction techniques.

Generative models for classification typically adhere to a certain mathematical formulation. We first denote π_k to represent the overall prior probability that a randomly chosen observation comes from the kth class. Then $f_k(x) = Pr(X|Y = k)$ denotes the density function of X for an observation that comes from the kth class. $f_k(x)$ is large in comparison if there is higher probability of an observation being in the kth class.

With these new assumptions, an update to Bayes Theorem can be made as follows:

$$\text{Bayes Theorem: } Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

3.1.1 Logistic Regression

Logistic Regression is a machine learning technique used in accordance with classification. It is quite similar to linear regression, especially in formulation, as it essentially uses multiple multiple instances of linear regression, thus, we obtain this formula:

$$p(X) = \frac{e^z}{1 + e^z}$$

Where $z = \beta_0 + x_1\beta_1 + \ldots + x_n\beta_n$. As you can see in the formula for logistic regression, more than two model parameters are used. Thus logistic regression is more useful when dealing with defining the odds of an observed event falling into a certain class. Note, we use this formula to find the logistic regression output and classify it to one of two classes based on $m \in (0, 1)$, where a result above m puts the observation in one class, and a result below m puts the observation in a different class [14]. To calculate the actual odds for an observed event falling into a class, we take the logarithmic odds, which give us the actual regression formula:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

[14] Notice we have n input parameters that we observe to find our classification, and $\{x_0, \ldots, x_n\}$ is the basis of these parameters. It is usually in the formula, $x_0 = 1$ due to the basis. We also have our weights which influence the importance of the different observation parameters $\{\beta_1, \ldots, \beta_n\}$.

3.1.2 K Nearest Neighbors (KNN)

One of the simplest classification methods to be commonly used in the data science world is K- Nearest Neighbors. It is a well-liked option for many applications because it is simple and easy to understand. The Euclidean distance metric is used by the k-nearest-neighbor (KNN) classifier to assess how similar two test samples are to a given set of training samples. Define x_i as an input sample for p features, $(x_{1i}, x_{2i}, \ldots, x_{pi})$. Consider $|n|$ to be the total number of input samples such that $i = 1, 2, 3, 4, \ldots, n$. The Euclidean distance between x_i and x_j is

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$

where x_j is a new data point and $j = 1, 2, 3, \ldots, n$ [12]. After, arrange the distances in ascending order and select the k-nearest neighbors. For classification, assign the label with the majority vote among the neighbors[12]. The parameter k is typically chosen to be odd to avoid ties.

3.1.3 Support Vector Machine (SVM)

Support Vector Machine is adept at handling high-dimensional data and is celebrated for its efficacy in classification tasks [16]. The primary goal is to establish a decision boundary, referred to as the hyperplane, which maximizes the margin between the closest data points from each class, known as support vectors.

In a training dataset, x_i is a feature vector in $\mathbf{R}^n$ and $y_i \in \{-1, 1\}$ is the predictor variable for training compound i. The optimal hyperplane is expressed as $w \cdot x + b = 0$ where w is the weight vector, x is the input feature vector, and b represents the bias term [5]. To ensure effective separation, the weight vector w and bias b must satisfy specific inequalities for all elements of the training set

$$\begin{cases} w \cdot \mathbf{x}_i + b \geq +1 & \text{if } y_i = 1 \\ w \cdot \mathbf{x}_i + b \leq -1 & \text{if } y_i = -1 \end{cases}$$

The key objective during the training phase of an SVM model is to determine optimal values for w and b such that the hyperplane not only separates the data points but also maximizes the margin, $\frac{1}{\|w\|^2}$.

Support vectors are crucial elements in this process, as they are the data points for which the expression $|y_i(w \cdot \mathbf{x}_i + b)| = 1$ holds true [5].

3.1.4 Decision Trees

Decision Trees are pivotal in machine learning since they provide a transparent framework for decision-making [2]. In order to group samples with similar target values or labels together, a decision tree recursively partitions the feature space with a training vector $x_i \in \mathbf{R}^n$, where $i = 1, 2, \ldots, \ell$ and a predictor vector $y \in \mathbf{R}^\ell$ [11]. The decision function for classification can be expressed as

$$f(\mathbf{x}) = \text{argmax}_c \left(\sum_{i=1}^{N_t} I(y_i = c) \right)$$

We denote N_t is the number of samples in node t, y_i is the label of the i-th sample in node t, and c represents the class label. The splitting criterion is crucial, and popular measures include Gini impurity and entropy.

The Gini impurity for a node t is defined as

$$G(t) = \sum_{i=1}^{C} p_i(1 - p_i)$$

where C is the number of classes and p_i is the probability of class i in node t. Entropy, another splitting criterion, is defined as:

$$H(t) = -\sum_{i=1}^{C} p_i \log(p_i)$$

The goal is to minimize impurity or entropy across nodes, resulting in a well-structured Decision Tree [11, 2].

3.1.5 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a machine learning technique used for dimensionality reduction and classification. It is useful when dealing with the existence in multiple classes and seek to find the optimal linear combination of features that best discriminate between the different classes. The data is projected into a lower dimensional space while maximizing the separation between classes. LDA focuses on maximizing the variance between classes, while minimizing the variance within each class. It aims to find a linear transformation of the features that maximizes the separation between different classes in the data, and achieves this by maximizing the ratio of between-class scatter to within-class scatter. LDA follows the mathematical formulation of generative classification models and Bayes' Theorem holds true in the following form: $Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$ For the purposes of this analysis, LDA is dealing with multiple predictors, $p > 1$. Therefore, LDA assumes the multivariate normal Gaussian density function is as follows:

$$f(x) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} exp\big(- \frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\big)$$

For $p > 1$ predictors, the Bayes' classifier in LDA assumes he observations in the kth class are drawn from a multivariate Gaussian distribution $N(\mu_k, \Sigma)$, where μ_k is a class-specific mean vector, and Σ is a covariance matrix that is common to all K classes"([3]). Then after plugging the density function, $f_k(X = x)$, into Bayes Theorem, it results in our discriminant

function:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k$$

LDA's Bayes classifier then assigns an observation $X = x$ to the class with the highest discriminant score. The discriminant function includes estimations for parameters unique to classes such as π_k, the prior probability, μ_k, class specific mean vector, and Σ, a shared covariance matrix [3].

To approximate the Bayes classifier, LDA inserts estimates for π_k, μ_k and σ^2 into the discriminant function, which is calculated for each class. $\hat{\mu}_k$ represents the class-specific mean vector, $\hat{\sigma}^2$ is the common within-class variance, and $\hat{\pi}_k$ is the estimated prior probability of class k.

The observation is assigned to the class with the highest discriminant score produced by

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

To approximate the Bayes classifer, LDA leverages estimates for π_k, μ_k, and σ^2. The key estimates used include:

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i - \hat{\mu}_k)^2$$

$$\hat{\pi_k} = \frac{n_k}{n}$$

Here, n denotes the total number of training observations, and n_k is representative of the

number of training observations in the kth class. $\hat{\mu_k}$ is the class-specific mean vector and $\hat{\sigma}^2$ is a common covariance matrix for all classes. LDA then weighs all of these estimates to approximate the Bayes classifier and make predictions on a data set [3].

3.1.6 Quadratic Discriminant Analysis (QDA)

Where LDA works with observations within classes from a multivariate, normal distribution, have a class-specific mean vector, and a covariance matrix that is common to all classes, Quadratic Discriminant Analysis (QDA) classifies data in a slightly different way. Simply, QDA classifies data based on the assumption of class specific observations being from a Gaussian distribution, then, parameters in the Bayes' theorem are substituted for estimates to perform accurate prediction [3].

The notable difference between LDA and QDA, is that each class in QDA has a unique covariance matrix. Considering an observation in the kth class with the form $X \sim N(\mu_k, \Sigma_k)$, where Σ_k represents the covariance matrix for the kth class, the Bayes classifier categorizes an observation $X = x$ into the class for the largest $\delta_k(x)$, the value of the discriminant function [3]. Similar to LDA, the QDA classifer inputs estimations for Σ_k, μ_k, and π_k into the discriminant function, $\delta_k(x)$.

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

$$= -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

The parameters, Σ_k, μ_k, and pi_k, are estimated based off of the training data, which

ensures the classifier is unique to each class. After the QDA classifier inputs the different estimates, the decision rule assigns an observation $X = x$ to the class with the largest discriminant score, δ_k [3].

Similar to the discriminant function in LDA, QDA's $\delta_k(x)$ is used to classify observations into different classes based on the value of the features. In QDA, the discriminant function for the respective kth class is will usually be of order 2 to ensure the best results and is used to find the probability of an observation belonging to class k.

QDA is often preferred over LDA because it is able to account for covariance matrices specific to each . (LDA makes the assumption of having the same covariance matrix across all classes.) The flexibility provided with QDA's classification can prove to be useful when considering data sets that have distinct variance within classes. QDA's quadratic discriminant function allows for more complex and intricate decision boundaries, thus proving to be useful when dealing with nonlinear relationships in data [3].

3.2 Dimensionality Reduction and Clustering Methods

Dimensionality Reduction is a common machine learning technique used to reduce the number of features used while making predictions. Certain features have more impact on the accurate prediction of a model than others, thus dimensionality reduction seeks to help the model focus on relevant features, improve the computational efficiency, and boost model performance. While in many cases, having a lot of data provides more information for training, high dimensional data sets can lead to over fitting and model complexity, thus increasing the

computational power needed to solve the model as well as increasing model complexity [1].

Where classification is a supervised learning method with labeled data, clustering is a unsupervised learning technique used to describe grouping together similar data that does not have any labels. Often the goal of clustering methods are not to perform prediction, but to organize data by similarities into groups.

3.2.1 Principal Component Analysis

Principal Component Analysis' main function is to identify a smaller number of dimensions that are the most "interesting" to the data. Interesting in the sense of the observations varying throughout each dimension.

If we have some elements ϕ_{j1}, recall that normalized means $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ If $X_1, X_2, ..., X_p$ is a set of features, the ideal dimensions or principal components are found through the normalized linear combination that has the biggest variance:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p$$

"The elements $\phi_{11}, ..., \phi_{p1}$ are referenced as the *loading* of the first principal loading component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11}\phi_{21}...\phi_{p1})^T$" ([3]). The loading elements are normalized and restricted to one to reduce the computational complexity of the model.

To determine the number of components to retain, we plot the explained variance ratio. The cumulative explained variance helps to strike a balance between retaining sufficient

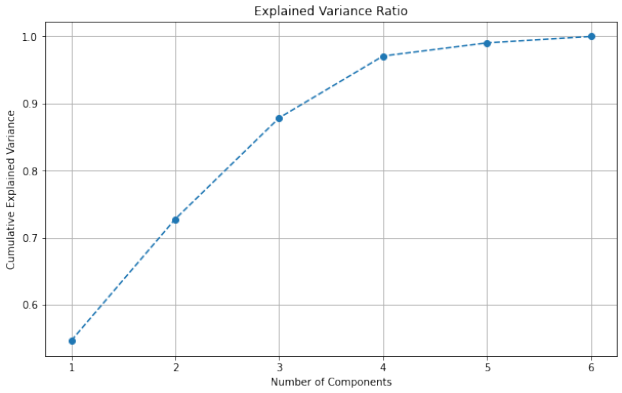


Figure 2: Explained Variance Ratio Plot to determine number of principal components

information and reducing dimensionality. The plot visually displays the explained variance for each component, while showing the proportion of total variance explained as the number of components increases [7]. Similar to the elbow method for KNN, we then choose the best number of components accordingly, in Figure 2, our number of principal components would be four.

Let's say we have some data set $n \times p$, X, where each variable has been centered to have a mean of 0. To compute the first principal component of the np data set we have to find a linear combination of the sample features that are of the form:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \ + \phi_{p1}x_{ip}$$

This form maximizes the variance, thus putting all this information together, the optimization problem becomes:

$$\text{maximize}_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1}x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Because the variables are being centered, the average values will also be 0. Therefore, from this we know the solution provides the loading vector $\phi_1 = (\phi_{11}, \phi_{21}, ..., \phi_{p1})$ [3].

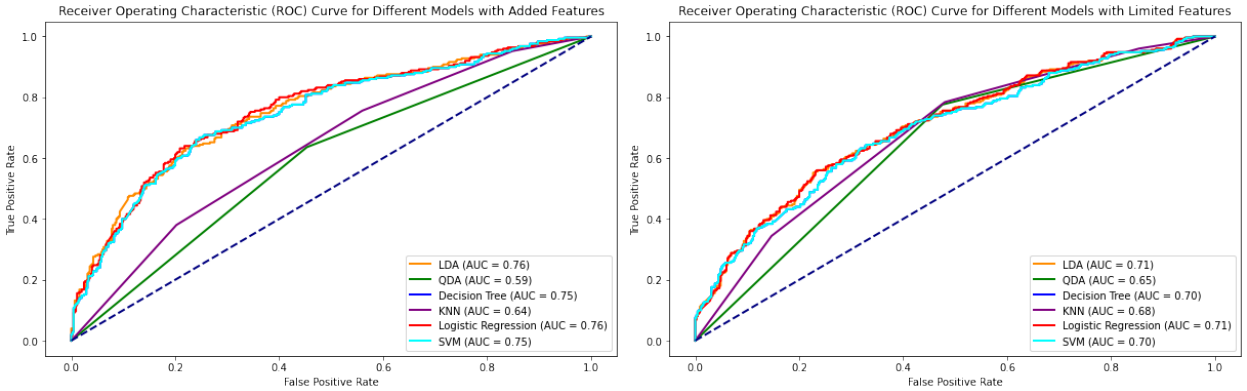
It is important to note that where LDA and QDA qualify as both dimensionalty reduction and classification algorithms, Principal Component Analysis (PCA) is exclusively used for dimensionality reduction. Therefore, after PCA is enacted on a data set, in order to determine it's effectiveness, an additional classification technique must be performed.

3.2.2 K-Means Clustering

The K-means algorithm is a dimensionality reduction machine learning algorithm with the purpose of classifying observations into certain subset of predictions. It is very useful for grouping data based on patterns within observations. Unlike many machine learning algorithms, this does not require data to be trained as it uses a different method based on the already present observations, thus it is an unsupervised learning method. K-means groups data into K groups, or 'clusters' based on their proximity to each other using centroids. Thus, the objective is to minimize this function:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} ||x_i - c_k||^2$$

Here, w_{ik} is a binary variable for if observation i is in group k. We also have the centroids c_k, and the observations x_i. Minimizing this function gives us the best possible clusters [6].



(a) ROC Curve with Added Features (b) ROC Curve with Facial Features

Figure 3: ROC Curve Analysis observes additional features provides higher accuracy.

4 Number of Features Comparison and Analysis

In the pursuit of unraveling gender patterns from facial images in our database, our initial focus centered on a set of six extracted facial features. Following a comprehensive analysis of these features, we sought out to learn whether the integration of additional, non-facial features could potentially enhance our results.

Motivated by insights derived from the correlation matrix, we strategically introduced supplementary features into our analysis and extracted those facial landmarks. Landmarks pertaining if the individual in the image was wearing lipstick, a necktie, or makeup. The outcome from the additional feature integration not only showcased a divergence in results but, notably, reflected an overall improvement. This led us to the conclude the inclusion of these extra features is beneficial. See Figure 3

Specifically aiming for higher accuracy with nine features as opposed to six, we conducted a percent difference analysis across all methods, focusing particularly on accuracy. For the purpose of algorithmic comparison, it is established that relying solely on facial features

might be insufficient for gender recognition in images. Features such as the presence of lipstick, makeup, and necktie, while seemingly unrelated to facial structure, play a pivotal role in refining the accuracy and robustness of gender classification algorithms. Through this analysis, we discovered that algorithms implemented with nine features — eyebrow length, eyebrow height, lip size, eye length, round face, face height, lipstick, makeup, and necktie — performed better. Consequently, for the remainder of our analysis, we plan to proceed using these results.

	LDA	**QDA**	**SVM**	**DT**	**KNN**	**Log Reg**	**Avg. % Diff.**
Accuracy % Diff.	11%	-6%	11%	0%	-11%	8%	2%
Sensitivity % Diff.	1%	-20%	1%	2%	-4%	4%	-2%
Specificity % Diff.	21%	4%	22%	-2%	-17%	13%	7%
AUC % Diff.	7%	-9%	8%	0%	-9%	7%	1%

Table 2: Percent Difference between using 9 Features vs. 6 Features in Analysis

For precise data related to this feature comparison please see Table 7, Table 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, and Figure 14 in the appendix.

5 Method Analysis

For each machine learning algorithm, an accurate analysis can be conducted by observing the accuracy, confusion matrix, sensitivity, specificity, area under the curve (AUC), and receiver operating characteristic (ROC) curve. The accuracy represents how correct a classification model is and is calculated as $\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$. A confusion matrix is a table used to represent a summary of the prediction results by showing the total number of true positives, true negatives, false positives, and false negatives, displayed in a matrix form.

The sensitivity or true positive rate, represents the proportion of actual positive or correct instances identified by the model, $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$ The specificity or true negative rate represents the proportion of actual negative or incorrect instances that were identified, $\frac{\text{True Negative}}{\text{True Negative + False Positive}}$. The ROC curve graphically represents the balance between the true positive rate and the false positive rate. The ideal model, should be represented by the curve adhering to the top-left corner of the graph [3].

5.1 Logistic Regression

Recall that logistic regression is one of the machine learning algorithms that prefer data to be split into training and testing. Here, the training data is used to calculate that m to figure out where the observations should be separated into different classes. The testing data uses that m and checks how reliable the algorithm is. Here, our methodology is that we split the data of the 200,000 face images into training and testing data based off of 6 certain features. We used the training data to calculate the best fitting weights $\{w_0, w_1, w_2, w_3, w_4, w_5\}$. Running the model for the training data gives us this log-odds function:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = 0.01846423 - 0.01066424x_1 + 0.13663294x_2 - 0.13912467x_3$$

$$-0.0190916x_4 + 0.24235172x_5$$

We use the testing data to see how well it matches with our weights for the regression function, where $m = 0.5$ as it is the median. Unfortunately, we see that linear regression

was 60% accurate, meaning that the algorithm as a whole was not very reliable. We could say that we had an unsatisfactory m, which may be a possibility to our low accuracy, but we also have a high sensitivity of 85%, which an incorrect m wouldn't simply solve. We also have a specificity of 46%, which is also quite high. Lastly, we have 0.76 for the AUC. In general, these high offsets may be due to complications in the actual data we extracted, or maybe issues within our facial data extraction tool, as we tried to extract features such as eyebrow length.

5.2 K Nearest Neighbors (KNN)

In evaluating the performance of the KNN model, we employed standard classification metrics, including accuracy, sensitivity, specificity, and area under the curve. Additionally, we examined the confusion matrix to gain insights into the model's predictions. We achieved an accuracy of 55.29%, sensitivity of 75.6%, specificity of 44%, and area under the curve of 0.598. The reason these results were obtained can be due to many factors. For instance, the choice of k, or the feature selection.

Throughout different literature reviews, the best way of selecting the optimal k value is $k = \sqrt{n}$ where n is the total number of input samples in the training dataset [8]. For our project, we decided to approach the value of selecting k by using the Elbow Method. Usually, the Elbow Method is implemented for clustering models like K-means, but for this case, the Elbow Method identifies the lowest error rate for $k = 1, 2, \ldots, 40$. Based on *Figure* 4, the k value we choose is $k = 21$. The decision to select $k = 21$ aligns with the principle

of preventing overfitting while maintaining a sufficiently generalized model which in return boosted our model up a little in accuracy. On the other hand, as previously stated in the

Figure 4: Elbow Method for KNN

Methodology: Data section of this paper, there were around 200,000 face images with various poses, backgrounds, and faces, indicating a diverse data set. However, due to the sheer volume of faces, the Euclidean Distance calculation encountered challenges. Specifically, when classifying face images based on facial features, such as eyebrow height, the numeric values for these features often exhibited minimal variations across gender categories.

In the context of facial features, it was observed that certain characteristics, including eyebrow height, showed similarities between male and female face images. The abundance of face images, regardless of gender, led to instances where the Euclidean Distance between corresponding facial features was relatively small. Consequently, KNN struggled to effectively discriminate between male and female faces solely based on these subtle variations, impacting its classification performance.

5.3 Support Vector Machine (SVM)

Like KNN, we analyzed the performance using the classification metrics of accuracy, sensitivity, specificity, and area under the curve. The SVM achieved accuracy of 59%, sensitivity of 84.8%, specificity of 44.89%, and area under the curve of 0.6484. Out of all the kernels we considered, the linear kernel consistently outperformed the others in terms of accuracy. Despite the linear kernel yielding the highest accuracy, the overall accuracy was still relatively low.

As mentioned in the KNN results above, the facial features selected for our data lack a clear distinction between female and male images. Consequently, some data points may overlap with the hyperplane, posing a challenge for the SVM to achieve higher accuracy.

Interestingly, while accuracy was low, sensitivity was relatively high. This observation could be attributed to the class imbalance within the dataset, with more female images than male images. However, it is essential to interpret these results cautiously, taking into account the imbalanced nature of the dataset and the inherent challenges posed by the similarity in facial features between male and female images.

5.4 Decision Trees

The Decision Tree achieved an accuracy of 57.86 %, sensitivity of 71.6%, specificity of 50.22%, and area under the curve of 0.61. In Figure5, illustrates part of the decision tree that led to the model's classification metrics. It is important to note that Decision Trees are known for over-fitting the training data, which can result in lower accuracy on unseen data. Despite

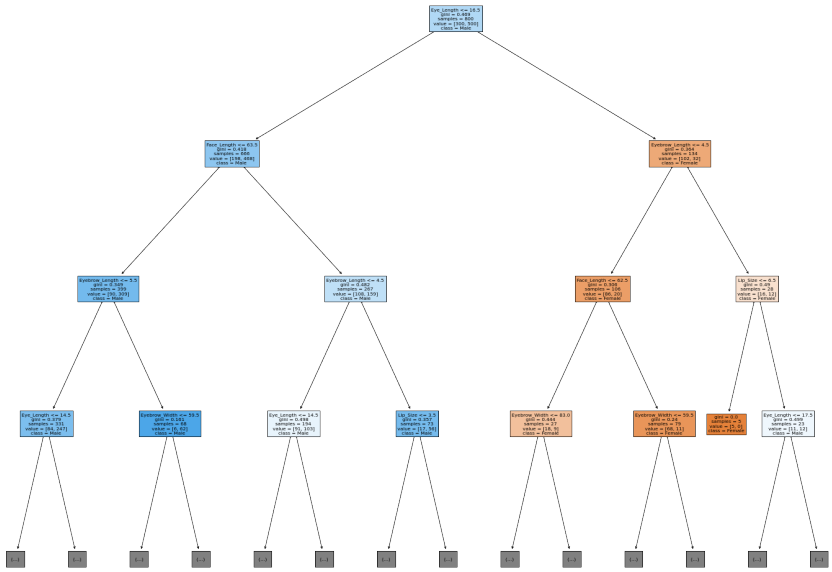


Figure 5: Decision Tree Model Based on Nine Features

the model's reasonable performance on the provided data set, there is a concern that the Decision Tree might have captured noise or specific patterns in the training data that do not generalize well to new, unseen instances, particularly from the class imbalance.

5.5 Linear Discriminant Analysis (LDA)

In the provided analysis of machine learning models using various features, Linear Discriminant Analysis (LDA) achieved an accuracy of 61.71%. This indicates that LDA correctly predicted the gender of the subjects in the dataset with an overall accuracy of 61.71%. However, to gain a deeper understanding, it's essential to consider other metrics.

The sensitivity of LDA is 0.828, indicating its ability to correctly identify males and females from the data set. The specificity is 0.5, suggesting a moderate ability to correctly

identify non-males and non-females. The area under the curve (AUC) is 0.7565, reflecting the model's ability to discriminate between the two classes.

The low accuracy score for LDA could be influenced by bad feature representation, insufficient influence in the selected features, or an inherent complexity in predicting gender from these attributes. Additionally, issues such as data imbalance and the need for model adjustments might contribute to the observed performance challenges.

Overall, LDA demonstrates relatively good performance in gender prediction based on the provided features. Additionally, comparing these results with other models in the analysis can provide insights into the relative performance of LDA in the given task.

5.6 Quadratic Discriminant Analysis (QDA)

In the analysis of machine learning models, Quadratic Discriminant Analysis (QDA) achieved an accuracy of 57.86%. This implies that QDA accurately predicted the gender of the subjects in the data set with an overall accuracy of 57.86%. To delve deeper into the performance, let's consider other important metrics.

The sensitivity of QDA is 0.636, indicating its ability to correctly identify males and females in the data set. The specificity is 0.5467, suggesting a moderate ability to correctly identify non-males and non-females. The area under the curve (AUC) is 0.5913, reflecting the model's ability to discriminate between the two classes.

While QDA demonstrates reasonable accuracy, its performance metrics, especially sensitivity and specificity, should be carefully considered based on the specific requirements of

the application. These metrics provide insights into the model's ability to correctly identify positive and negative instances, respectively.

The relatively low performance of the QDA in gender prediction may stem from factors such as the model's complexity, especially in cases where the assumption of different covariance matrices for each class does not align well with the data. The selected features, including facial features and additional attributes like makeup, necktie, and lipstick presence, might not sufficiently capture gender-related patterns. Imbalances in gender representation, and noise in the data could also contribute to the challenges faced by the QDA model. Addressing these issues, refining feature selection, and exploring alternative modeling approaches may be essential to enhance the predictive accuracy for gender recognition.

5.7 Principal Component Analysis

In this section, we present the results of the Principal Component Analysis (PCA) conducted on the entire dataset encompassing all 40 features. Additionally, to maintain consistency with the analysis conducted on the original six features as well as nine features, PCA was applied to the respective subsets.

	LDA	**QDA**	**SVM**	**Decision Tree**	**KNN**	**Log Reg**
Accuracy	44.71%	57.00%	36.00%	64.29%	63.86%	64.14%
Sensitivity	0.9240	0.8560	1.0000	0.0000	0.0440	0.0440
Specificity	0.1822	0.4111	0.000	1.0000	0.9689	0.9689
AUC	0.6879	0.6336	0.5000	0.5000	0.5000	0.5000

Table 3: Comparison of analysis results of multiple machine learning models after principal component analysis has been run on the original six extracted features.

After plotting the explained variance, four principal components were chosen. The results

of PCA applied to the original six extracted features yielded varied results among the machine learning models (Table 3). Notably, Decision Tree achieved the highest accuracy at 64.29%, outperforming others. However, this high accuracy seems to be driven by a lack of sensitivity, as evidenced by a specificity score of 1.0, implying a tendency to predict the positive class without effectively capturing true negatives. SVM, on the other hand, demonstrated a lower accuracy of 36.00%, indicating challenges in correctly classifying instances. The sensitivity score of 1.0000 suggests a perfect ability to identify true positives, but this is offset by the low specificity, resulting in an overall accuracy drop.

It is important to consider the trade-off between sensitivity and specificity when interpreting these results, as models may excel in one aspect while struggling in another. Additionally, the relatively low AUC scores across all models signal a need for further investigation into the model's ability to discriminate between classes.

After plotting the explained variance, four principal components were chosen for this analysis as well. PCA when applied to the nine extracted features also resulted in diverse performance across machine learning models (Table 4). Decision Tree and SVM exhibited accuracy scores at 53.43% and 36.00%, respectively. However, these models showed differing strengths. Decision Tree achieved a balanced sensitivity of 0.6760 and specificity of 0.4556, while SVM demonstrated perfect sensitivity (1.0000) but lacked specificity (0.0000). This implies that SVM excelled in correctly identifying positive instances but struggled with true negatives. The low AUC scores across all models, including the 0.5000 AUC for SVM, suggest challenges in discriminating between classes. It's essential to carefully consider the specific

goals and requirements of the application when interpreting these results, as different models may have distinct strengths and weaknesses.

	LDA	**QDA**	**SVM**	**Decision Tree**	**KNN**	**Log Reg**
Accuracy	46.29%	53.43%	36.00%	53.43%	50.14%	46.57%
Sensitivity	0.8720	0.7960	1.0000	0.6760	0.8680	0.8680
Specificity	0.2356	0.3889	0.0000	0.4556	0.2978	0.2978
AUC	0.6600	0.5924	0.5000	0.5000	0.5000	0.5000

Table 4: Comparison of analysis results of multiple machine learning models after principal component analysis has been run on the nine extracted features.

Because PCA is a dimensionality reduction technique, how results would look if we applied it to the most amount of features available, which was not the features we extracted from the images, but from the attribute list provided with the dataset, which contains 40 features for each image in the dataset. After plotting the explained variance, 10 principal components were chosen for this method. Then PCA was applied and the results are listed in Table 5.

	LDA	**QDA**	**SVM**	**Decision Tree**	**KNN**	**Log Reg**
Accuracy	91.79%	88.80%	92.58%	91.05%	92.60%	92.57%
Sensitivity	0.9322	0.7966	0.9339	0.8963	0.9208	0.9211
Specificity	0.9077	0.9530	0.9200	0.9206	0.9297	0.9289
AUC	0.9200	0.8748	0.9270	0.9084	0.9252	0.9250

Table 5: Comparison of analysis results of multiple machine learning models after principal component analysis has been run on all 40 features in the dataset (data pulled from the .csv file of attributes).

The results indicate that PCA on the complete set of 40 features maintained high accuracy across various models, including LDA, QDA, SVM, Decision Tree, KNN, and Logistic Regression. Notably, SVM achieved the highest accuracy at 92.58%, making it a promising choice for subsequent analysis.

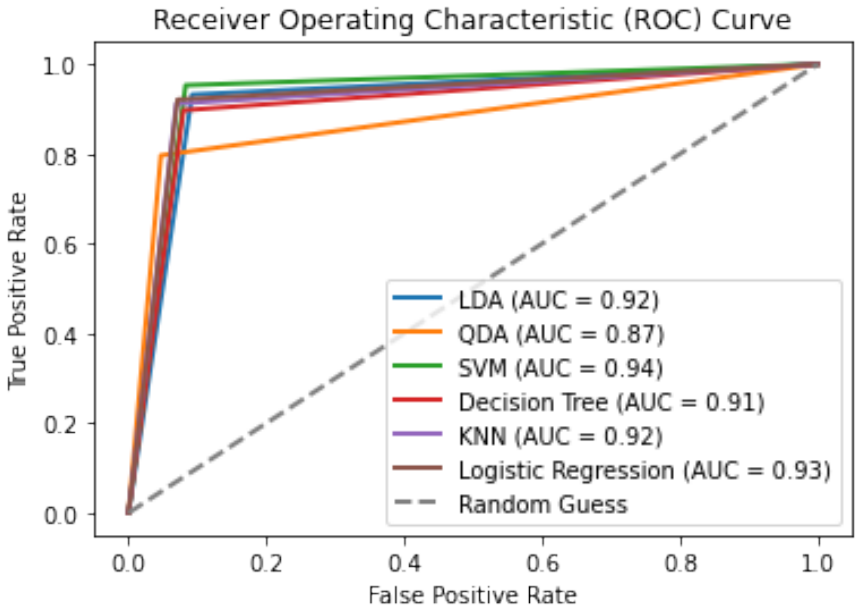


Figure 6: PCA Analysis AUC with 40 Features

This outcome suggests that employing PCA on a broader set of features, beyond those extracted from facial images, can enhance the performance of machine learning models. The preservation of accuracy, sensitivity, specificity, and AUC metrics demonstrates the effectiveness of PCA in capturing relevant information while reducing dimensionality.

SVM performed the best after PCA in this method. These results can be attributed to its ability to handle non-linear decision boundaries through kernel functions. SVM is well-suited for high-dimensional spaces, making it effective after dimensionality reduction by PCA. The algorithm's focus on maximizing the margin between classes and its robustness in imbalanced data sets contribute to its success. Additionally, if the features retained by PCA align well with the data set's characteristics, SVM can benefit from the best feature selection. Overall, SVM's strengths in handling non-linearity, high-dimensional data, and imbalanced data sets make it a robust choice, especially when combined with PCA for feature reduction.

While the 40 feature PCA analysis is useful to see, in the context of our study, using those results in the comparison would skew our data immensely. Since Decision Tree exhibited the highest accuracy for the six feature extraction analysis and QDA exhibited the highest accuracy for the nine feature extraction analysis, they were respectively chosen as the PCA representative for further analysis and comparison when referring to the comparison of six and nine features.

5.8 K-Means Clustering

For our methodology, we had no need to split the data into training and testing. We took the data of 3 of the facial features: Eyebrow Length, Lip Size, and Face Length. We converted the data for a subsection of 500 random faces into vectors, which is seen in the graph below. We then randomly assigned the data into 2 classes, Male and Female, and calculated their centroids. We then simply use the K-means algorithm to recalculate the proximity of nodes to each other and recalculate their centroids many times to minimize the function seen in the Implementation section of this paper. As a result of implementing k-means 100 times on the data, we get this graph:

It is important to note that this observation only yielded a 63% accuracy rounded up so it is not too reliable for a good classification machine learning algorithm in this specific scenario. This may because of an error in one of the images that results in that huge outlier in the bottom left of the graph. Because data is classified based on proximity to centroids, that centroid is in its own section with the outlier, causing it to have its own classification

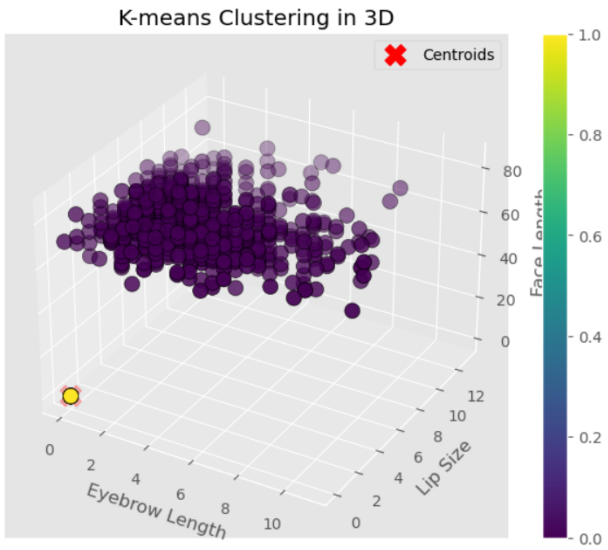


Figure 7: Caption

separate from the rest of the data. Thus, removing it may increase the accuracy of k-means by a lot.

5.9 Analysis Results

In the provided analysis of machine learning models utilizing facial features (eyebrow length, eyebrow height, lip size, eye length, round face, and face height) and additional features (makeup present, necktie present, lipstick present), the models were evaluated based on accuracy, sensitivity, specificity, and the AUC. Among the models, the one that performed the best overall is the model utilizing Principal Component Analysis (PCA) with Decision Tree (DT), achieving an accuracy of 64.29%. This model shows promise in accurately predicting gender based on the combination of facial and additional features.

	LDA	QDA	SVM	DT	KNN	Log Reg	PCA with DT
Accuracy	61.71 %	57.86%	59%	57.86%	55.29%	60.14%	64.29%
Sensitivity	0.8280	0.6360	0.8480	0.7160	0.7560	0.8560	0.000
Specificity	0.5000	0.5467	0.4489	0.5022	0.4400	0.4600	1.000
AUC	0.7565	0.5913	0.6484	0.6100	0.5980	0.7600	0.5000

Table 6: Comparison of analysis results of multiple machine learning models of facial features(eyebrow length, eyebrow height, lip size, eye length, round face, and face height) and additional features (makeup present, necktie present, lipstick present)

6 Conclusion

In conclusion, a variety of classification, clustering, and dimensionality reduction techniques were used in our thorough investigation of gender detection in face images, which produced insightful findings and useful performance indicators. We used PCA for dimensionality reduction, K-means for clustering, SVM, KNN, Decision Tree, Logistic Regression, QDA, and LDA for classification.

Achieving high accuracy in gender detection proved challenging due to factors such as class imbalance, non-distinctiveness of numeric values in facial features, noisy data, and limitations in the Dlib package. These limitations in the Dlib package stem from face images having too dark or light of a background, or face being turned, causing the data point to be values of 0. The lack of reproducibility in certain scenarios and the inability of Dlib to identify faces in challenging conditions led to misclassifications, adding complexity to the task.

Our study emphasizes the significance of taking into account various methodologies when addressing gender detection in face images, despite these obstacles. In order to improve overall performance, future work should concentrate on reducing the limitations that have been

found, investigating different feature selection techniques, fine-tuning model parameters, and possibly combining the advantages of various approaches.

7 Roles

7.1 Ta'Destiny Geiger

My primary responsibility in the group project was to design and code the Feature Extraction process for the data frame. In addition, I played an important role in laying the groundwork for the project by significantly contributing to the introduction, methodology, and data sections. In addition, I was in charge of incorporating the processed data into the Support Vector Machine, Decision Tree, and K-nearest neighbors. This included not only applying the data to these algorithms but also fine-tuning parameters and ensuring seamless integration.

7.2 Ramya Nivedha Raja

Completed PCA, LDA, QDA formulation, implementation, and analysis. Compiled all code to work from same subset of data on the same document, and obtained results. Incorporated additional feature implementation with each algorithm, and performed the additional feature extraction (lipstick, makeup, necktie). Completed "number of feature comparison" analysis as well as coding and enacting the feature correlation analysis. Wrote introductions for sections of the paper, the abstract, and thoroughly proofread and organized the document.

7.3 Michael Ani

My responsibility in the group project was to incorporate the data into Logistic Regression and analyze it by detailing its accuracy, sensitivity, sensibility, and area under the curve. I also was tasked with clustering the data using the k-means algorithm and describing the significance and accuracy of it.

References

[1] Nanyuan Zhao David Zhang Changshui Zhang, Jun Wang. Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction. *Pattern Recognition Volume 37, Issue 2, February 2004, Pages 325-336*, 2004.

[2] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.

[3] Trevor Hastie Robert Tibshirani Daniela Witten, Gareth M. James. *An Introduction to Statistical Learning: With Applications in R.* Springer, 2013.

[4] Emon Kumar Dey, Mohsin Khan, and Md Haider Ali. *Computer vision based gender detection from facial image.* Citeseer, 2013.

[5] Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. Applications of support vector machine (svm) learning in cancer genomics. *Cancer genomics & proteomics*, 15(1):41–51, 2018.

[6] Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, April 2023.

[7] Alberto Ferrer José Camacho, Jesús Picó. Data understanding with pca: Structural and variance information plots. *Chemometrics and Intelligent Laboratory Systems Volume 100, Issue 1, 15 January 2010, Pages 48-56*, 2019.

[8] Upmanu Lall and Ashish Sharma. A nearest neighbor bootstrap for resampling hydrologic time series. *Water resources research*, 32(3):679–693, 1996.

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[10] Bruno Marcos. Image recognition - gender detection - inceptionv3, 2023.

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

[12] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[13] S Ravi and S Wilson. Face detection with facial features and gender classification based on support vector machine. *International Journal of Imaging Science and Engineering*, pages 23–28, 2010.

[14] Sandro Sperandei. Understanding logistic regression analysis. *Biochem Med (Zagreb)*, 24(1):12–18, February 2014.

[15] Tahmina Akter Sumi, Mohammad Shahadat Hossain, Raihan Ul Islam, and Karl Andersson. Human gender detection from facial images using convolution neural network. In *Applied Intelligence and Informatics: First International Conference, AII 2021, Nottingham, UK, July 30–31, 2021, Proceedings 1*, pages 188–203. Springer, 2021.

[16] Shan Suthaharan and Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[17] Dujuan Zhang, Jie Li, and Zhenfang Shan. Implementation of dlib deep learning face recognition technology. In *2020 International Conference on Robots & Intelligent System (ICRIS)*, pages 88–91. IEEE, 2020.

A Appendix

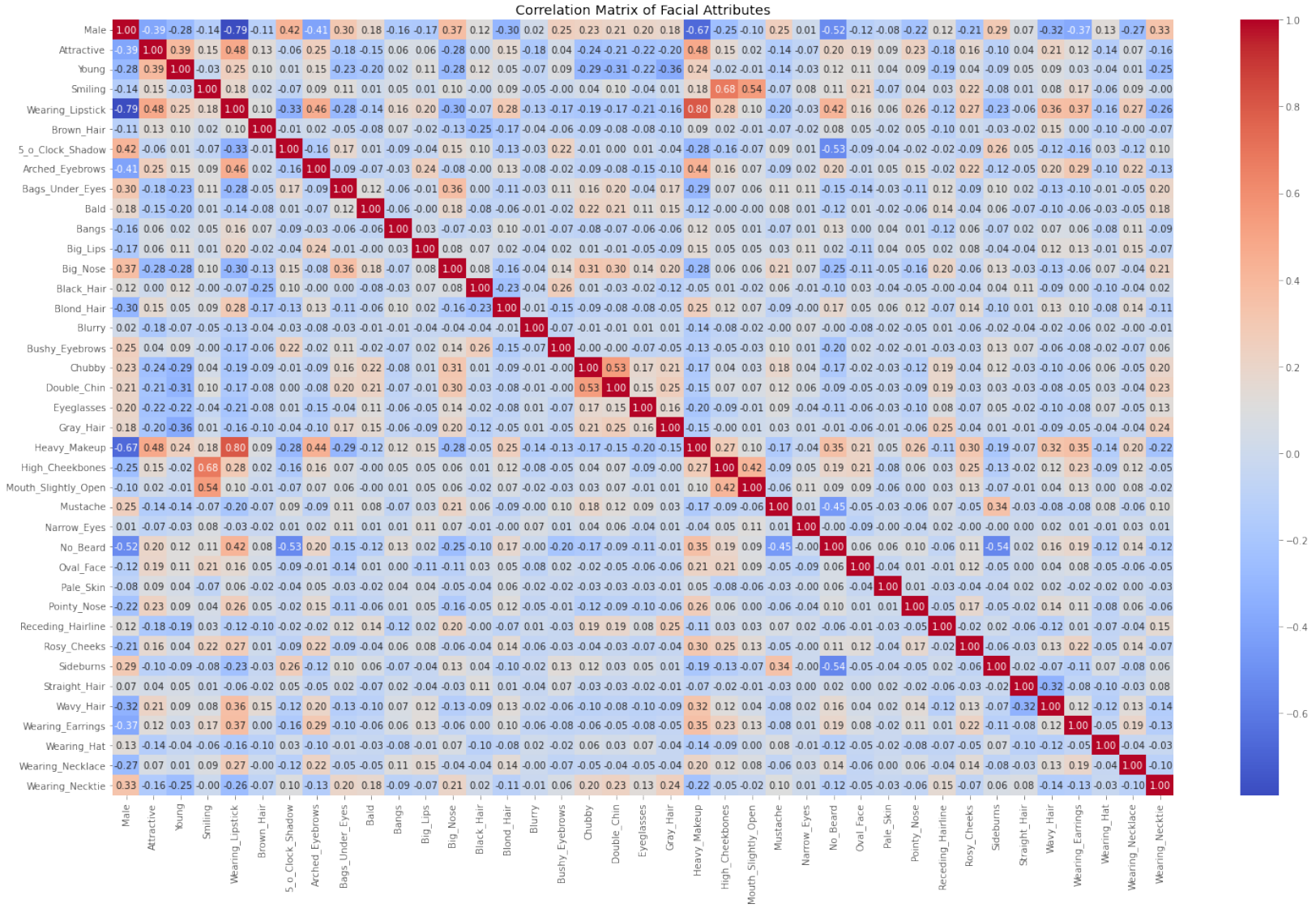


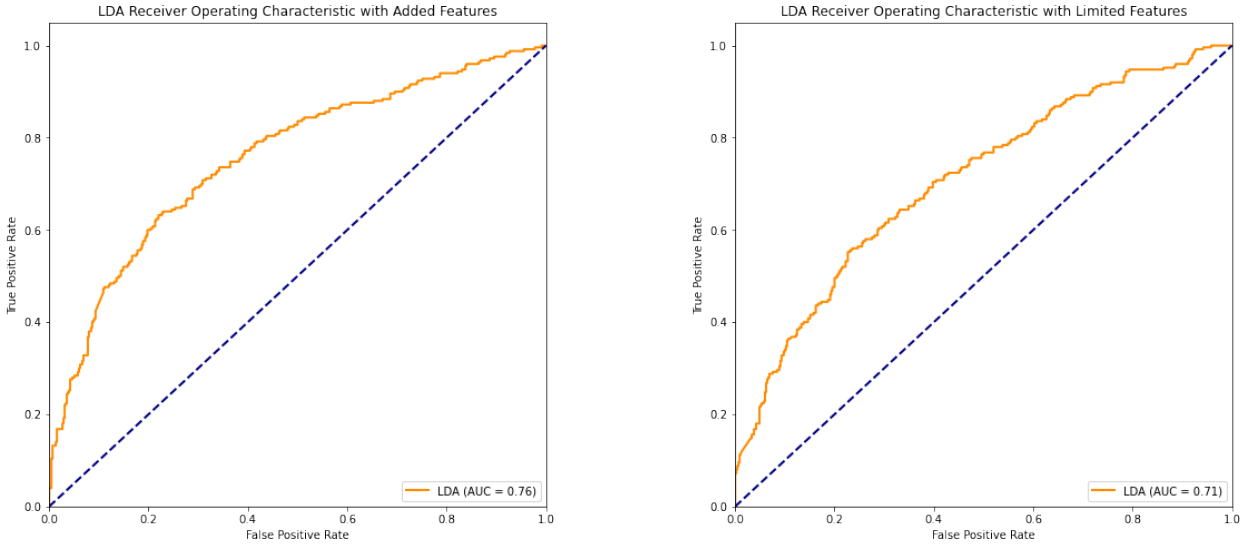
Figure 8: Correlation matrix of the 40 attributes from the csv file that accompanied the data set to help identify optimal features the optimal features to extract from images in data set.

	LDA	QDA	SVM	DT	KNN	Log Reg	PCA with DT
Accuracy	61.71 %	57.86%	59%	57.86%	55.29%	60.14%	64.29%
Sensitivity	0.8280	0.6360	0.8480	0.7160	0.7560	0.8560	0.000
Specificity	0.5000	0.5467	0.4489	0.5022	0.4400	0.4600	1.000
AUC	0.7565	0.5913	0.6484	0.6100	0.5980	0.7600	0.5000

Table 7: Comparison of analysis results of multiple machine learning models of facial features(6) and additional features (makeup present, necktie present, lipstick present)

	LDA	QDA	SVM	DT	KNN	Log Reg	PCA with QDA
Accuracy	55.14 %	61.43%	53.00%	58.00%	61.43%	55.43%	53.43%
Sensitivity	0.8160	0.7760	0.8360	0.7000	0.7840	0.8240	0.7960
Specificity	0.4044	0.5244	0.3600	0.5133	0.5200	0.4044	0.3889
AUC	0.7075	0.6502	0.5980	0.6100	0.6520	0.7100	0.5924

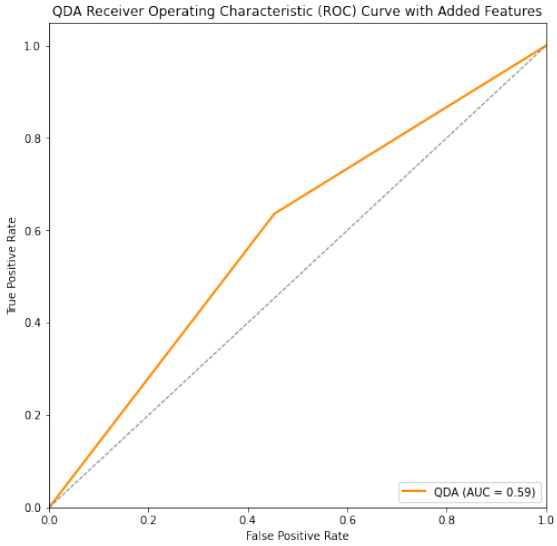
Table 8: Comparison of analysis results of multiple machine learning models, specifically facial features (6) that were extracted from photos



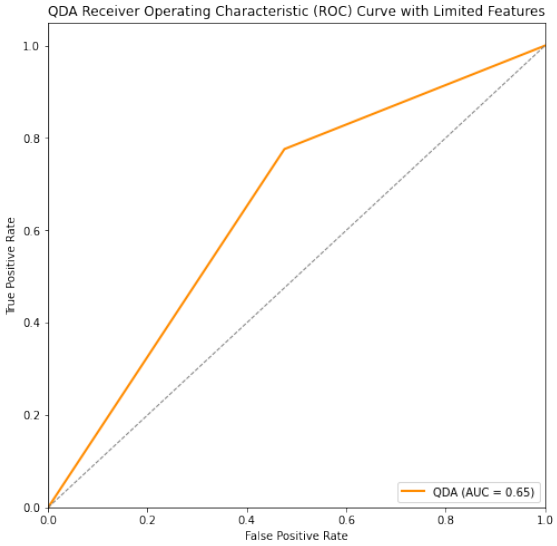
(a) LDA with Only Added Features

(b) LDA with Facial Features

Figure 9: LDA Feature Comparison

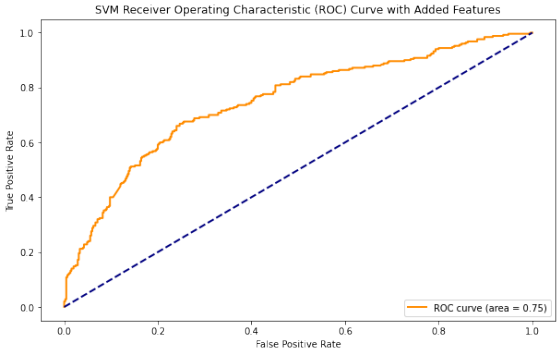


(a) QDA with Added Features

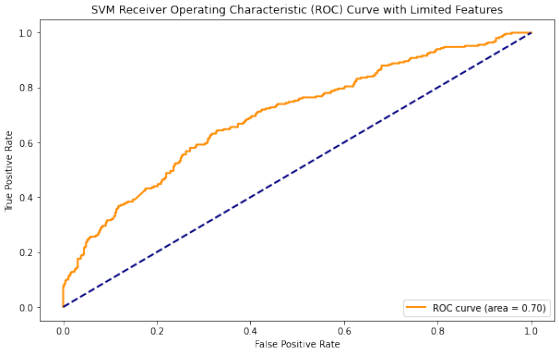


(b) QDA with Facial Features

Figure 10: QDA Feature Comparison

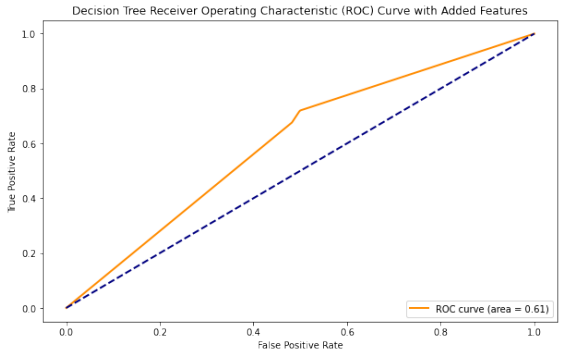


(a) SVM with Added Features

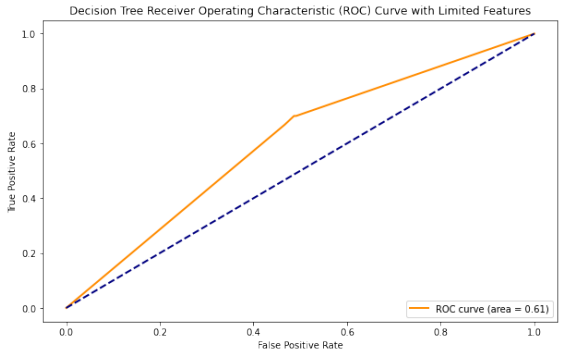


(b) SVM with Facial Features

Figure 11: SVM Feature Comparison

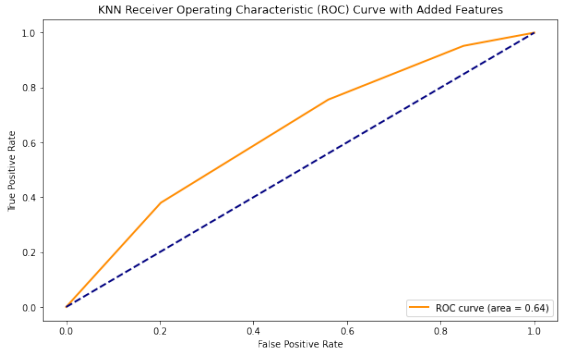


(a) Decision Tree with Added Features

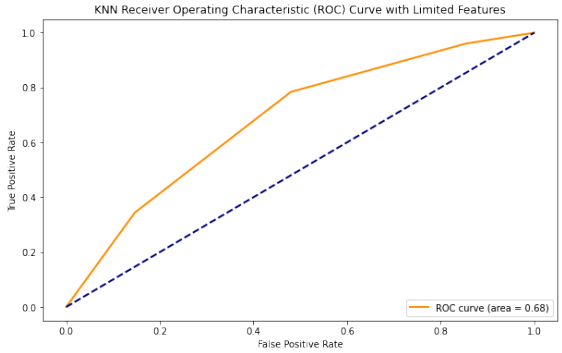


(b) Decision Tree with Facial Features

Figure 12: Decision Tree Feature Comparison

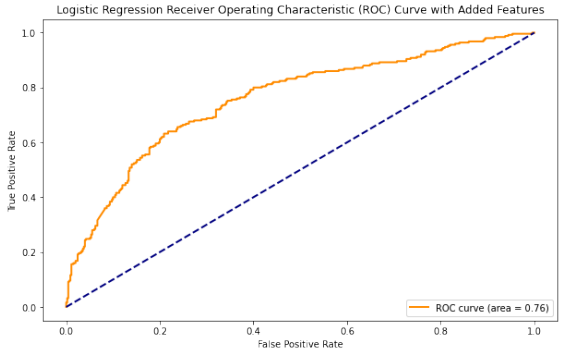


(a) KNN with Added Features

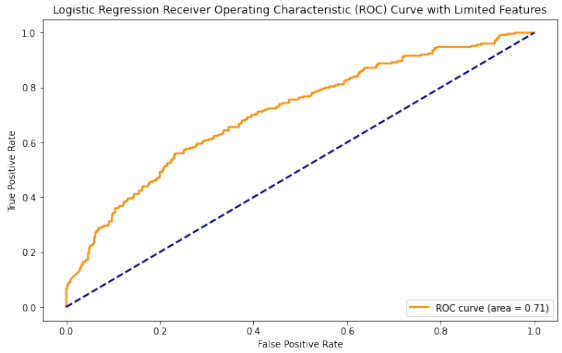


(b) KNN with Facial Features

Figure 13: KNN Feature Comparison



(a) Logistic Regression with Added Features



(b) Logistic Regression with Facial Features

Figure 14: Logistic Regression Feature Comparison