What are your contributions to the project

I did all of it.

* Summarize what you did in the project

I tried to follow what you have shown to us in the Avila project, more or less. Starting with exploratory data analysis and then training the models we have learned in the course. In the first section, I did some feature and class analysis to understand the correlation between features. MVG, Logistic Regression, Quadratic Logistic Regression, Linear and Kernel SVM, Gaussian Mixture Models. I selected the two best models for my system and started the experimental validation to compare the performance of models using the validation data.

* How could you implement non-linear models?

Non-linear models like Kernel-SVM and Quadratic Logistic Regression use non-linear separation rules or functions to separate the classes. For classification, the model is linear if you can plot all the n features in n-dimensional space, and there is a dimensional "line" (or plane, or hyperplane) that separates (or mostly separates) different classes. But in non-linear models, the decision function is not linear (quadratic) in x.

QLR:

We define a mapping that expands our feature space. We can train an LR model using expanded features rather than our normal features. We will obtain a model that has a linear separation surface in the space defined by the mapping. This space is also called expanded feature space. The LR model (both binary and multiclass) allows computing linear separation rules for the transformed features which correspond to quadratic forms in the original feature space. We are estimating quadratic separation surfaces in the original space. In general, we can consider a transformation of our feature space such that our classes are (approximately) linearly separable in the expanded feature space.

* Generative vs. Discriminative approach. What are generative approaches?

Discriminative probabilistic model:

* Model class posterior probabilities and assign labels according to posterior probabilities.
* Directly model class posterior probability.
* Class assignment is again based on the highest posterior probability.
* Probabilistic interpretation
* Cannot directly incorporate application-dependent information – The class prior probabilities are embedded in the model.

Discriminative non-probabilistic model:

* The output is a score.
* The score can be taken as a measure of the strength of the hypotheses (possible class assignments) under test.
* The class assignment, in general, is based on the best (usually highest) score.
* Cannot directly account for prior information (in some cases, it’s possible to recover probabilistic interpretations)

Generative probabilistic model:

* Model the joint distribution of features and labels and apply the Bayes theorem to compute posterior probabilities
* The model describes the data generation process in terms of class-conditional distributions.
* The model can incorporate application-dependent prior class information.
* The inference is based on the Bayes theorem.
* Class assignment is based on the highest posterior probability.
* Probabilistic interpretation, optimal decisions depending only on prior information.
* Most demanding — it requires modeling per-class densities of our data.
* Most informative — it allows us to generate new samples.

**PCA and LDA**

* What is dimensionality reduction?

The feature space is often very large and contains a large amount of unwanted and potentially harmful information. Dimensionality reduction techniques compute a mapping from the n-dimensional feature space to an m-dimensional space, with m ≪ n.

* Compress information (we want to retain discriminant information)
* Remove unwanted variability (noise)
* Data visualization
* Simplify classification (reduce the curse of dimensionality, reduce overfitting)
* we want to retain the maximum amount of information for given output size.

Overfitting: An over-complex model fits very accurately the observed data (very small training error) but is not able to provide accurate predictions for unseen data.

Curse of dimensionality: Volumes in high-dimensional spaces grow very fast, and data becomes very sparse.

* What is PCA doing?

PCA is an Unsupervised learning method that computes a mapping from N to M dimensional feature space. The selection of optimal m can be made by cross-validation using a validation set. PCA can be interpreted as the linear mapping that preserves the directions with the highest variance. We compute these principal components of our data based on the minimization of the average reconstruction error.

* How do you compute PCA?

Usually, PCA is applied directly to centered data:

* Compute sample mean
* Center data (Subtract mean from data)
* Compute the sample covariance matrix
* Compute the eigendecomposition of Covariance Matrix

Project the data in the subspace spanned by the m columns of U corresponding to the m highest eigenvalues. The selection of optimal m can be made by cross-validation using a validation set.

* What happens if the dataset is not zero-mean?

P represents a subspace whose axes pass through the origin. If the dataset is far from the origin, the first PCA direction will approximately connect the origin and the dataset mean. This direction (in most cases) is not very interesting.

* Does normalization change PCA directions?

Yes. Normalization is important in PCA since it is a variance-maximizing exercise. It projects your original data in directions that maximize the variance. When data is not normalized, it seems like one component with huge numbers explains most of the variance in the data. The reason for this is that PCA seeks to maximize the variance of each component. The PCA will select to project as much as possible in the direction of a feature whose variance is much greater.

* What is the reconstruction error?

Reconstruction error has been used in PCA as a criterion to estimate the principal components. A reasonable criterion may be the minimization of the average

reconstruction error. The general definition of the reconstruction error would be the distance between the original data point and its projection onto a lower-dimensional subspace.

In PCA, Reconstruction error or loss is the sum of eigenvalues of the ignored subspace. Let's say you have 10 Dimensional data, and you are selecting the first 4 principal components. Your principal subspace has 4 dimensions and corresponds to the 4 largest eigenvalues and respective vectors, So reconstruction error is the sum of 6 eigenvalues of the ignored subspace (the smallest 6).

Minimizing the reconstruction error means minimizing the contribution of ignored eigenvalues which depends on the distribution of the data and how many components we are selecting.

* How does LDA work, assumptions and limitations?

LDA Finds a direction that has a large separation between the classes and a small spread inside each class. We measure spread in terms of class covariance.

LDA: Maximizes the between-class variability over within-class variability ratio for the transformed samples.

The between-class covariance matrix can be interpreted as a covariance matrix for the class means, where each class is weighted by the corresponding sample size. The within-class covariance matrix can be seen as an (also weighted) average of the covariance matrix of each class.

We assume that Sw is full rank. The method was originally introduced to solve binary problems. Notice that, from the definition of SB, the number of non–zero eigenvalues is at most C−1. Therefore, LDA allows estimating at most C−1 directions.

Assuming that SW is positive definite (all eigenvalues are greater than 0), we can use a function that solves the generalized eigenvalue problem.

* In LDA, why do we first whiten SW and then SB?

We have seen that the LDA solution can be implemented as a first transformation that whitens the within-class covariance matrix, followed by a projection on the leading eigenvectors of the transformed between-class covariance. We first do the whitening because we need to estimate P1, which comes from the SVD decomposition of SW. The transformed SB thus can be computed with P1.

A whitening transformation is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables whose covariance is the identity matrix, meaning that they are uncorrelated, and each has variance 1.

* Relationship between LDA and the Tied Gaussian Covariance Classifier?

The decision rules are the same and both linear. Both use the same covariance matrices, within-class covariance matrix, and tied covariance matrix are the same. LDA looks for the direction which maximizes between-class covariance while minimizing the within-class covariance matrix. This corresponds to the classification rule of the Gaussian model with tied covariances. Indeed, LDA assumes that all classes have the same within–class covariance.

* What rank is the within covariance matrix? (SW)

SW is full rank. Assuming that SW is positive definite (all eigenvalues are greater than 0). So, all rows (or columns) are linearly independent.

**Probability and Distribution Functions**

What are models that describe discrete data?

How do we model discrete values?

Categorical or multinomial distributions. The frequency with which we observed value I in the class. We can adopt a naive Bayes approximation again and assume that features are independent. We have seen that occurrences can be modeled by multinomial distributions. We can compute model parameters (usually the frequency of occurrence) and compute the log-likelihood ratio as a decision rule. We can also compute class posterior once we choose class priors.

We can model a dataset of categorical samples as n independent categorical R.V.s. Each variable represents a token. The distribution is described by a vector of probabilities π

that allows computing P(X = j) = πj

Alternatively, we can model the dataset in terms of occurrences of events. We have a random vector Y = (Y1 . . . Ym) whose components are R.V. Yi corresponding to the number of occurrences of the event I in the dataset. Again, the distribution is described by a vector of probabilities π that represents probabilities of single events.

**MVG**

What do we use to estimate the parameters for a gaussian distribution in an MVG?

We assume that our data have been generated by Gaussian R.V.s. We use the Maximum Likelihood estimator to estimate the model parameters. The ML is the estimator is the value that maximizes the log-likelihood. The solution corresponds to the empirical mean and empirical covariance matrix of the data.

What is the loglikelihood ratio?

It represents the ratio between the likelihood of observing the sample given that it belongs to h1 or to h0. The log-likelihood ratio acts as a score, with a probabilistic interpretation.

Greater scores values imply our system favors class h1. Lower values mean it favors class h0

The decision requires comparing the LLR to a threshold T that depends on the application through the class prior probability.

How do you train the gaussian classifier model?

* Compute model parameters for each class.
* The empirical mean and empirical covariance matrix of the data.
* Compute densities for each sample considering the model parameters for each class.
* Compute the LLR that shows if sample X belongs to class h1 or h0.

**Decision Functions and Model Evaluation**

How do you compute DCF?

How accuracy and DCF are related?

How did you apply K-fold to compute the minimum DCF?

How would you compute optimal decisions?

Explain the differences between the results with validation and evaluation in the report.

How do you calculate actual DCF, and what do we use it for?

How do we make decisions?

How do you obtain the threshold from the training data?

Qualitative definition of Bayes risk (not the formula)

**Logistic Regression**

What is the assumption made from the Logistic Regression model?

How does the logistic regression model work, and what is the function we try to minimize?

When we calibrate scores using prior-weighted Linear regression, which data should we give as input to the function to estimate W and B? (I'm not speaking about the "x" that are the scores that we have to calibrate)

Hinge loss vs. Logistic loss

Speak about multiclass Logistic Regression

**SVM**

SVM, how to use it non-linearly?

SVM, in general, how do they work?

What is the meaning of SVM primal and dual solutions?

What do the hyperparameters in SVM represent?

What is the RBF kernel?

What does the kernel function represent?

What do Lagrangian multipliers represent?

Relationship between SVM and Logistic Regression + differences

**GMM**

GMM, what are they?

How would you do a Naive Bayes GMM?

How do you train a GMM, what do they achieve, and how do they work?

What do we do when we apply the EM algorithm?

What are responsibilities?

We can compute cluster (component) posterior probabilities. Observe that the responsibilities depend on the mean. If we knew the responsibilities, we could compute

We can interpret Uc as a weighted empirical mean. Given the responsibilities, we re-estimate the GMM parameters θ using the previous expressions.

What is KNN?

What is K-Means?