Individual Reports

In this document, each member details his contributions to the project. Any further detail is in the main report file.

D'Evangelista Nicolò - Gradient Ascent method

I worked about the part that concerns Gradient Ascent method, After the implementation I tested the method without a features rescale, but I was not satisfied with the results, as it was slow to converge and low learning rate was required. After the features rescale (in range 0 - 1) I could immediately observe a great improvement, it converges pretty fast and accuracy is 83%, within 1000 iterations with a learning rate of 0.3. In conclusion, from the results obtained, I can say this method is a good metric for deciding if a customer will be satisfied or not.

Choma Patryk - Newton Method

I implemented the Newton method algorithm for the project. It was relatively difficult to implement, because it was very common for the hessian matrix to be singular. Combined with the size of the dataset and the many features, it was also very hard to compute. To manage to obtain results, I settled on using just 1% of the dataset for training. Surprisingly, the method produced pretty good predictions, managing to achieve ~80% accuracy and a nice area under the precision/recall curve. The result was so surprising, it required further tests, and I figured out that different train/test splits produce very different results, which is understandable enough (might get lucky on the sampling), and further solidified after some research, when I found that *saddle points*, to which Newton's method tends to converge to, become very numerous in high-dimensional spaces, such as our feature-space. In conclusion, the method does not seem particularly suited for the problem, but it did produce decent results with some luck in our initialization.

Hysaj Rigels - Gaussian Discriminant Analysis

My work consisted in dealing with Gaussian Discriminant Analysis. I have attempted to run many tests, for example, by trying different splitting of the dataset. When using GDA we achieve the best accuracy, with a rate of 83.3% correct predictions when training with 70% of the dataset. However, one of the most relevant detail i noticed it was that this method it is not significantly influenced by change of dataset. For example, the accuracy tends to remain high when using 1% or 0.01% of dataset. Anyway the higher the percentuage of usage of the dataset the higher will be the accuracy. Another thing that i noticed was that not every feature has shown to have a Gaussian distribution, so if all the features had a Gaussian distribution or if we removed all the features without it, the accuracy may vary significantly. This method is faster than any other methods that we used but, unfortunately, has the smallest area under RoC curve.

Chicca Lorenzo - Evaluation

I was tasked to do the evaluation part of the project. The first thing I did was, in agreement with all other members, reducing the training set of the data we decided to use for the project. This decision was made because such a large dataset would need a considerable amount of time to be processed and we were able to remove all rows containing missing values from it, also not wanted features because they weren't related with our goal. Such features were: "flight distance", "departure/arrival tim convenient", "gate location", "departure delay in minutes", "arrival delay in minutes". With the reduction of the dataset, the work of the group began and I had to wait for their results. After receiving

the work done by the other members and discussed their results together, my job was to make the graphs to represent them using the plot function already seen in the first assignment. From the graphs made by the function, it is visible that the GA method is the one which delivers the best result, in contrast to the Newton method that creates a strange fragmented line and the GDA that makes a fairly similar curve as the GA but slightly less precise. We can see a very interesting rivalry between the GA and GDA methods, because one it's more precise but the other it's faster: since we value more precision than velocity, GA appears to be truly the best method. After evaluating the results, together we decided that it would have been better to have all the plots in the same graph, and so the method plot_all_rpc was made. This method uses plot_rpc to do its job by iterating over all the possible thresholds and plotting all the different precision and recall values.