Donia Youssef Mokhtar Youssef 10863392

Manuel Peracci 10824742

Mobile Radio Networks project

1. Introduction

For this assignment, the python template provided by professor Pimpinella is used as a baseline and edited according to the requirements. Firstly, the data is imported and plotted, trends are observed and mathematical transformations are applied. Afterwards, features are selected and edited to allow for a more accurate model to be obtained. The technique used for dimensionality reduction is PCA and the algorithm selected is logistic regression. The classifier is optimized through K-fold cross validation and the prediction performance is done through the area under the ROC curve (AUC) value.

2. Features Selection & Engineering

First of all, the original data is processed and filtered removing the following unnecessary columns in both the training and test sets: '*Unnamed: 0*' and '*Unnamed: 0.1*'. Looking at the correlation matrix (seen below in *Figure 1*), the following is noted:

- The data is quite uncorrelated with the 'User_Satisfaction' variable (which from now and on will be referred to as "target"). In fact, the highest correlation, equal to -0.1324, was with *Cumulative_Full_Service_Time_UMTS.*
- There is clearly some severe collinearity between these two pairs:
    - *Cumulative_YoutubeSess_LTE_DL_Volume* and *Cumulative_YoutubeSess_LTE_DL_Time*

    - *Cumulative_YoutubeSess_UMTS_DL_Volume* and *Cumulative_YoutubeSess_UMTS_DL_Time*

- The classes are slightly unbalanced (0.32 and 0.68), but the gap is not too severe. So, we decided to create a new dataset using some oversampling technique on dataset c4 (have a look at *c5_sd*).

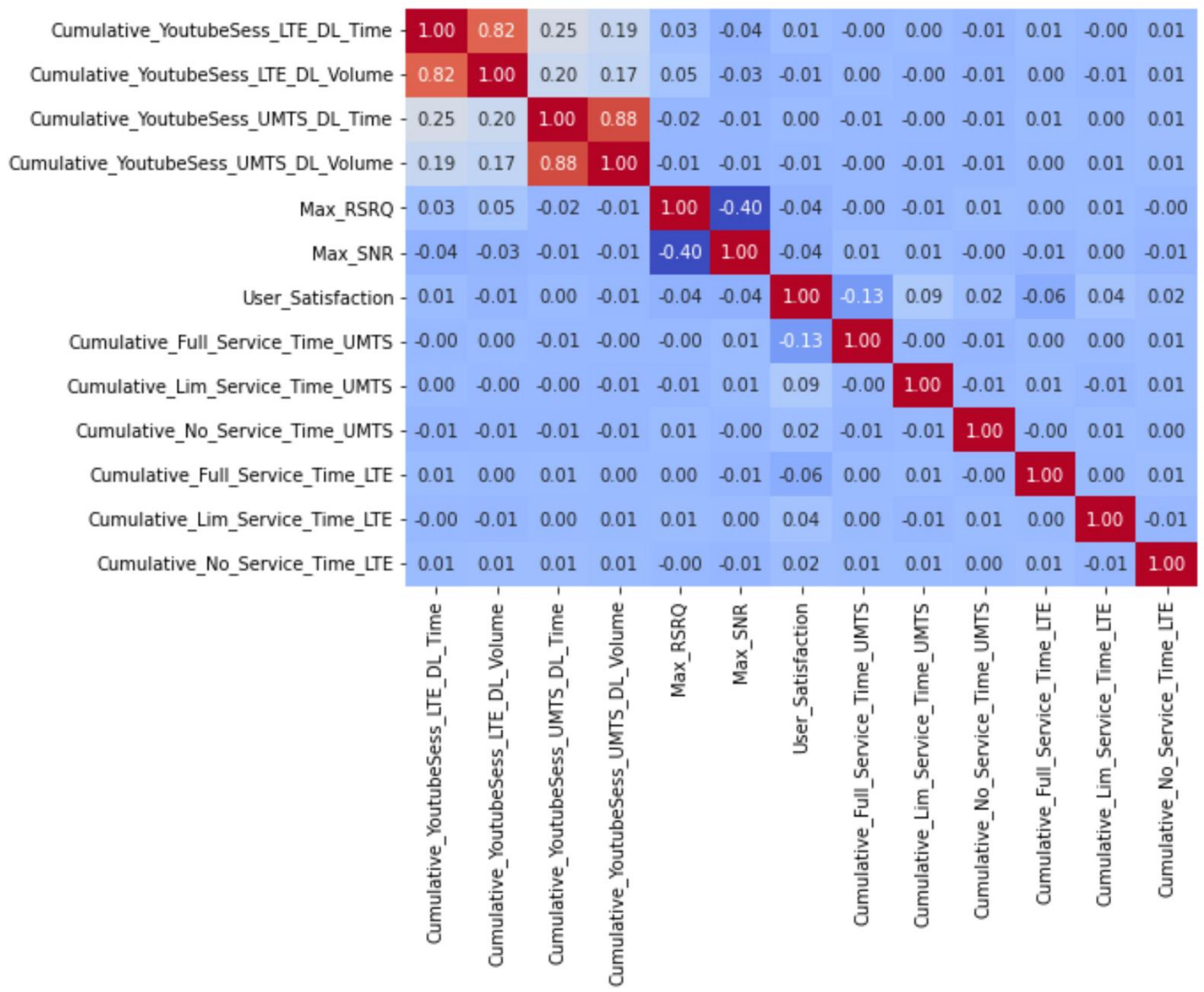


Figure 1 shows the correlation matrix on the original dataset

Taking into account all these facts, we decided to work with many different datasets in order to find the best dataset for predictions. Hence, the data is processed using feature selection, mathematical transformations, standardization and creating new variables from existing ones. In the end, 8 different datasets were created (which were different combinations of new and old features) and they are described below. To avoid redundancy, new features that are used for more than one dataset are only defined once. For all datasets from number 2 to 7, *Max_RSRQ* and *Max_SNR* were not modified and all rows were normalized for both the training and test datasets. Also the statistics of the training set were used in order to normalize the test set.

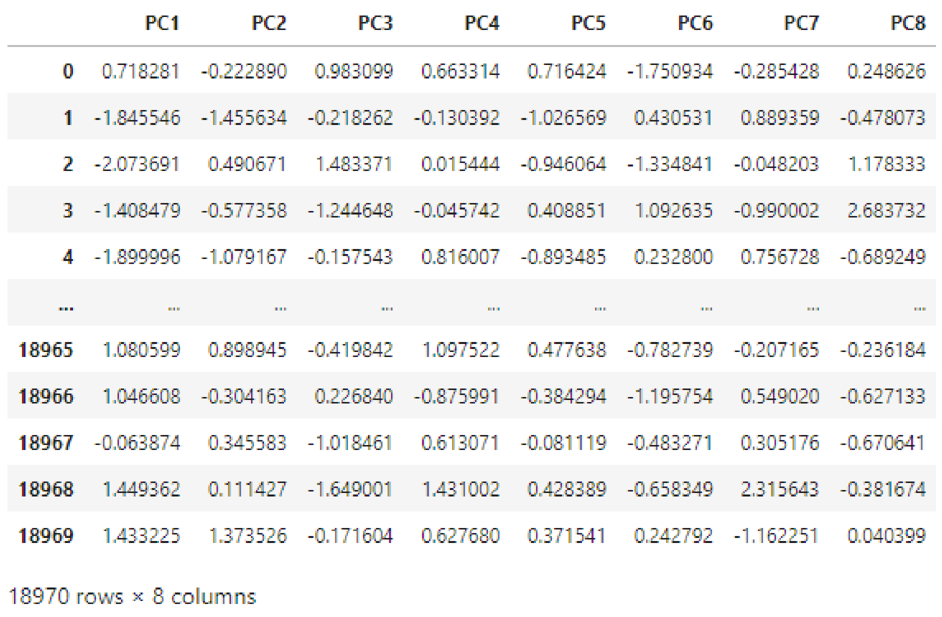
1. The original data was left unaltered.
2. *c_sd* where most of the original columns are eliminated and the following new features are added:
   - *Rate_LTE_DL* which is the *Cumulative_YoutubeSess_LTE_DL_Volume* divided by *Cumulative_YoutubeSess_LTE_DL_Time*. Notice that rows with this last variable equal to zero were modified with 1 (this transformation doesn't impact much on the data, but it helps to avoid an inevitable error). Notice also that *Rate_UMTS_DL* was not added as the correlation was lower than the corresponding Volume that was kept.
   - *Some_Service* which is the sum of all the features when there is some service, ie: *Cumulative_Full_Service_Time_LTE, Cumulative_Full_Service_Time_UMTS, Cumulative_Lim_Service_Time_LTE* and *Cumulative_Lim_Service_Time_UMTS*.
   - *No_Service* which is the sum of *Cumulative_No_Service_Time_UMTS* and *Cumulative_No_Service_Time_LTE*.
3. *c1_sd* is organized mostly in pairs in the following way:
   - *Cumulative_Full_Service_Time* is the sum of *Cumulative_Full_Service_Time_LTE* and *Cumulative_Full_Service_Time_UMTS*.

- *Cumulative_Lim_Service_Time* is the sum of *Cumulative_Lim_Service_Time_LTE* and *Cumulative_Lim_Service_Time_UMTS*.
- *Cumulative_No_Service_Time* is the sum of *Cumulative_No_Service_Time_LTE* and *Cumulative_No_Service_Time_UMTS*.
- *Rate_LTE_DL*

4. *c2_sd* is organized mostly in pairs in the following way:
   - *Cumulative_Full_Service_Time*
   - *Cumulative_Lim_Service_Time*
   - *Cumulative_No_Service_Time*
   - *Total_Rate* is the sum of the two volumes (LTE and UMTS) divided by the sum of the two download times (LTE and UMTS). Notice that rows with this last variable equal to zero were modified with 1 (to avoid an error).
5. *c3_sd* contains the following edited features which were described above:
   - *Some_service*
   - *No_service*
   - *Total_Rate*
6. *c4_sd* is composed of the following features:
   - *Cumulative_YoutubeSess_UMTS_DL_Volume*, *Cumulative_Lim_Service_Time_UMTS*, *Cumulative_Lim_Service_Time_LTE*, *Cumulative_No_Service_Time_LTE***.** Notice that some features were kept and others removed based on the results of the correlation with *User_satisfaction*.
   - *Rate_LTE_DL*

- *log_Cumulative_Full_Service_Time_UMTS*, *log_Cumulative_No_Service_Time_UMTS* and *log_Cumulative_Full_Service_Time_LTE* are the transformations of *Cumulative_Full_Service_Time_UMTS*, *Cumulative_No_Service_Time_UMTS* and *Cumulative_Full_Service_Time_LTE*, respectively. All transformations applied are log(x + 1). Notice that the log works nicely because there are a lot of observations with low value and few observations with a high value.

7. *X_std_pca* is created using the PCA technique on the correlation matrix (we are exploiting the geometry of the covariance matrix after the dataset has been normalized. In fact, computing the covariance matrix on standardized data, we will obtain the correlation matrix. It's also possible to exploit the geometry of the covariance matrix, but the magnitudes among the features are too different and it would be wrong to exploit it in this specific case). The used dataset was c4_sd. Noticed that we obtained the same number of new features, but we looked at the explained variability and we cut the number of features using the rule of thumb of 80%: we kept all the features up to the first one that had a cumulative explained variability of at least 80% and we removed the rest of them. We ended up with 8 features. Notice that we used the loadings to transform the testset (in any case, we have never used any statistics coming from the test set). In *Figure 2* it is possible to look at the structure of the new dataset.



	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
0	0.718281	-0.222890	0.983099	0.663314	0.716424	-1.750934	-0.285428	0.248626
1	-1.845546	-1.455634	-0.218262	-0.130392	-1.026569	0.430531	0.889359	-0.478073
2	-2.073691	0.490671	1.483371	0.015444	-0.946064	-1.334841	-0.048203	1.178333
3	-1.408479	-0.577358	-1.244648	-0.045742	0.408851	1.092635	-0.990002	2.683732
4	-1.899996	-1.079167	-0.157543	0.816007	-0.893485	0.232800	0.756728	-0.689249
...	...	...	...	...	...	...	...	...
18965	1.080599	0.898945	-0.419842	1.097522	0.477638	-0.782739	-0.207165	-0.236184
18966	1.046608	-0.304163	0.226840	-0.875991	-0.384294	-1.195754	0.549020	-0.627133
18967	-0.063874	0.345583	-1.018461	0.613071	-0.081119	-0.483271	0.305176	-0.670641
18968	1.449362	0.111427	-1.649001	1.431002	0.428389	-0.658349	2.315643	-0.381674
18969	1.433225	1.373526	-0.171604	0.627680	0.371541	0.242792	-1.162251	0.040399

18970 rows × 8 columns

Figure 2. Components of the PCA

8. *c5_sd* is a dataset with the same features of dataset *c4* (which is *c4_sd* denormalized) with normalization applied after oversampling the data. Therefore, this dataset is perfectly balanced.

3. Cross-Validation and Logistic Regression

For the machine learning part, we decided to rely on the *Logistic Regression* technique to solve the binary classification problem and the *K-fold Cross-Validation* (CV) to tune the hyperparameters. These two techniques were applied to all the previous training sets: first the CV and second the logistic regression with the best parameters. Both the techniques are taken from the library sklearn.

For the CV we used *GridSearchCV*, which is the main function to apply this technique. The number of splits was 10. The most important inputs for this function are:

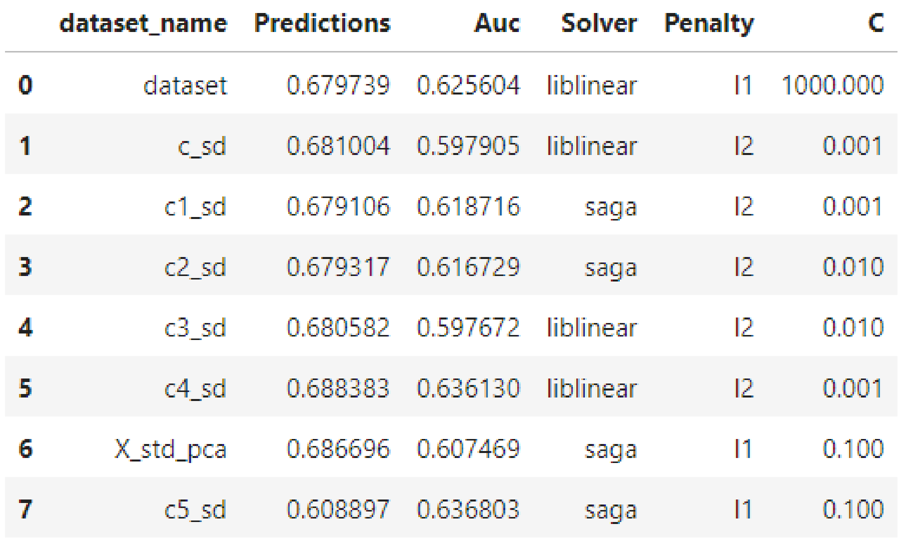
1. The model, which is the logistic regression
2. The grid, which is the space of possible hyperparameters for the CV
3. The scoring, which was set to "*roc_auc*"

The used hyperparameters are:

1. The solver used and it could be "*liblinear*" and "*saga*"
2. The penalty, which could be "*l1*" or "*l2*"
3. The hyperparameter *C*, which could be: *1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1, 10, 100* or *1000.*

   C is the inverse of the regularization strength.

This function was applied to all the previous dataset and *Figure 3* reports the best hyperparameter for each dataset.



	dataset_name	Predictions	Auc	Solver	Penalty	C
0	dataset	0.679739	0.625604	liblinear	l1	1000.000
1	c_sd	0.681004	0.597905	liblinear	l2	0.001
2	c1_sd	0.679106	0.618716	saga	l2	0.001
3	c2_sd	0.679317	0.616729	saga	l2	0.010
4	c3_sd	0.680582	0.597672	liblinear	l2	0.010
5	c4_sd	0.688383	0.636130	liblinear	l2	0.001
6	X_std_pca	0.686696	0.607469	saga	l1	0.100
7	c5_sd	0.608897	0.636803	saga	l1	0.100

Figure 3. Grid of the results of the k-fold Cross-Validations applied to all the available datasets. Only the best models are reported.

Then, logistic regression was applied on each dataset with the best computed hyperparameters. Finally, predictions on the test sets were performed and the results are reported in the column "prediction" of *Figure 3*. The predictions were also expressed in terms of the area under the curve (AUC) for each model and the relative ROC curve was drawn. The AUC values are reported for each model in the column "AUC" of *Figure 3*.

4. Results

It is interesting to note that the PCA didn't help in the dimensionality reduction. It was possible to reduce the dimensions from 11 to 8, but other better strategies for dimensionality reduction were possible. In fact, having some knowledge on the data it was possible to create new useful features such as the downlink rate. To conclude on PCA, even if all the AUC results were all around the same values, the AUC on the dataset with PCA was one of the worst.

We noticed that the first 4 experiments (*c_sd*, *c1_sd*, *c2_sd* and *c3_sd*) obtained poor scores for the AUC, even worse than the original dataset. The only explanation is that the feature selections were not helpful for the datasets.

On the other hand, *c4_sd* performed better than the previous ones, but only slightly better than the original one.

Finally, *c5_sd* which has the same shape of *c4_sd*, but with a balanced dataset, performed slightly better than c4_sd. This difference is so small that we can say the oversampling is not necessary. The ROC curve for dataset *c5_sd* is shown in *Figure 4* and *5*.

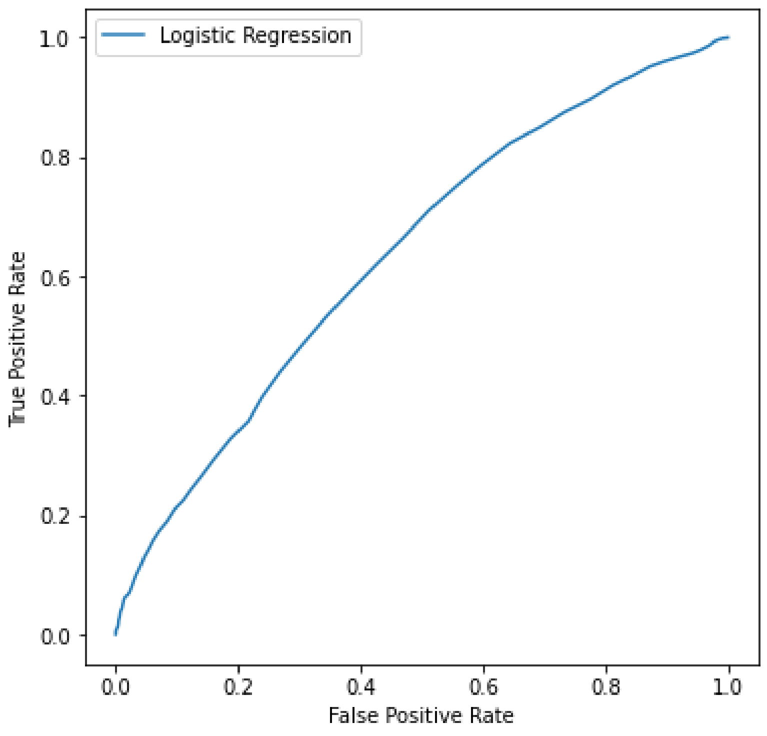


Figure 4. ROC curve of the best model

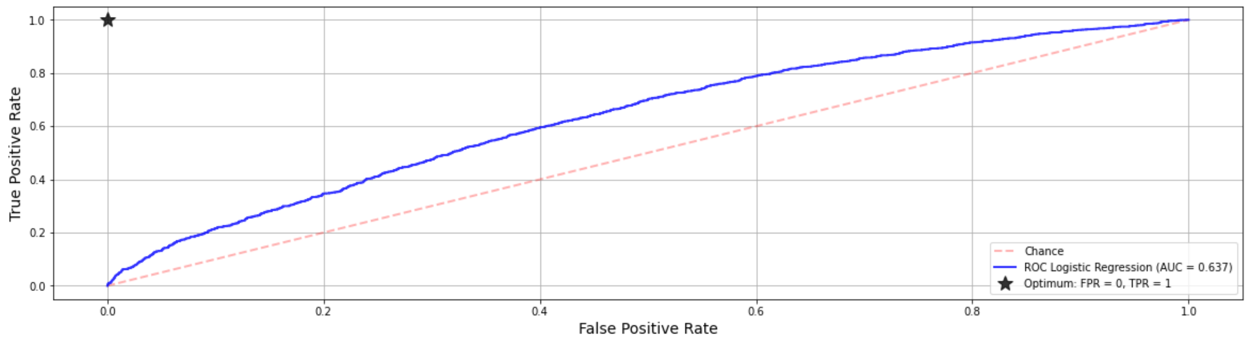


Figure 5. Roc curve of the best model

5. Conclusion

In this project, we have dealt with the complex problem of predicting user satisfaction from a set of parameters. Using logistic regression yields acceptable values (AUC equal to 0.636818); however, a possibility is to further implement a more robust machine learning classifier, such as random forest or neural networks and see if this could improve our results. Moreover, we could try adding different features and trying different combinations of those. Having more data samples collected could also lead to a more accurate model.