# Classification of NS data

Marina Berbel, Miquel Miravet-Tenés
(Dated: April 22, 2022)

## I. RANDOM FOREST

### A. Using only independent recovered variables

Training and testing using: $m1_{rec}, m2_{rec}, chi1_{rec}, chi2_{rec}, snr$. Standard deviation of score during crossvalidation: 0.000277942225285724 . Mean: 0.9744487408123469 Score 0.9748495808263471 . Optimum forest found: 200 trees, entropy criteria and sqrt max features

Score on testing: 0.9747829130485508

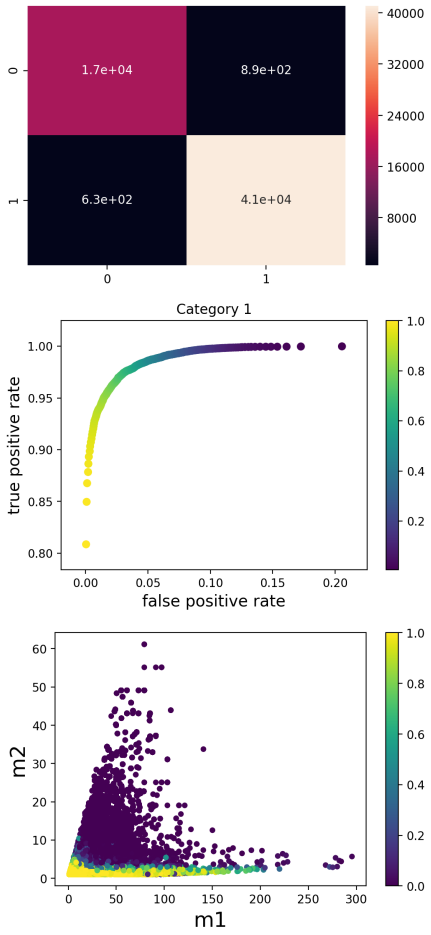


FIG. 1. Results with optimum model after crossvalidation. Using just independent recovered values (masses, spins and SNR).

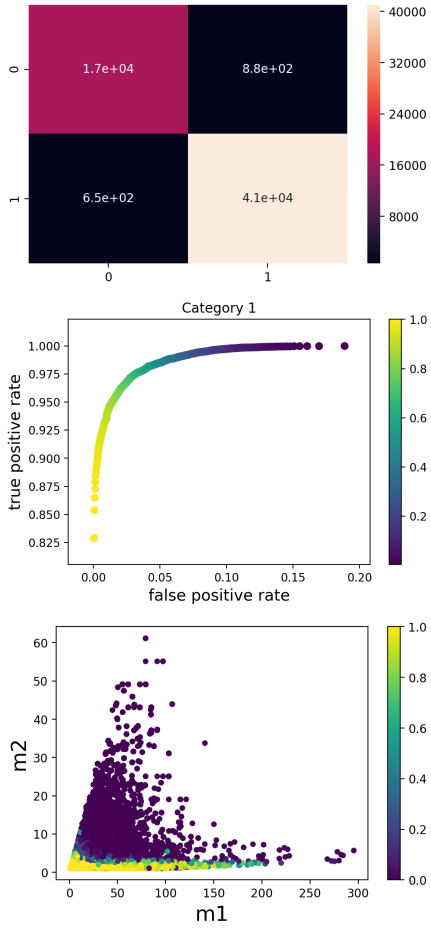


FIG. 2. Results with optimum model after crossvalidation. Using all recovered values (masses, spins and SNR, mass ratio, chirp mass, compactness and Risco).

### B. Using recovered variables

Training and testing using: $m1_{rec}, m2_{rec}, chi1_{rec}, chi2_{rec}, mc_{rec}, q_{rec}, R_isco_{rec}, Compactness$ Standard deviation of score during crossvalidation: 0.0002807499052054315 . Mean: 0.9744466574442907 Score 0.975032917215287 . Optimum forest found: 400 trees, entropy criteria and sqrt max features

Score on testing: 0.9745329088818147

### C. All injected values

No crossvalidated Score on testing: 0.9999833330555509

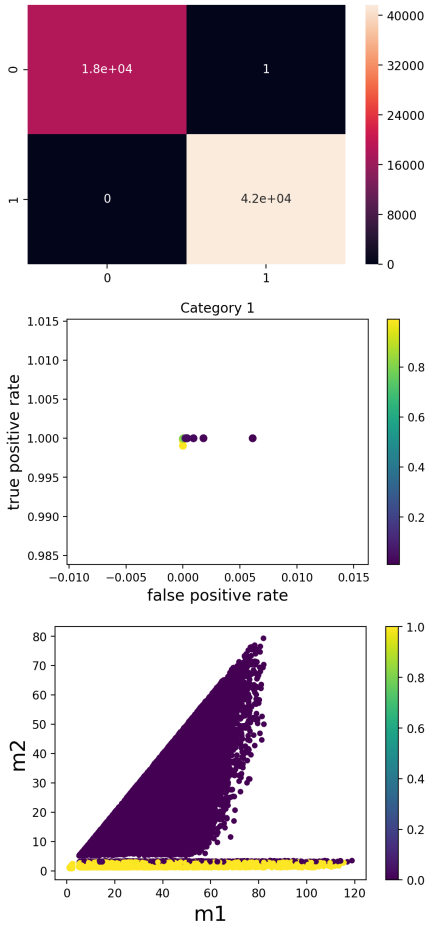


FIG. 3. Results using all the injected values

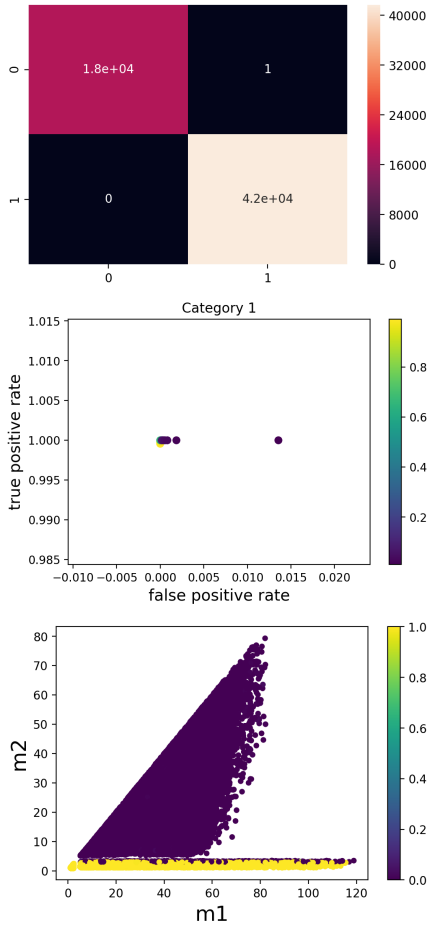


FIG. 4. Results using the independent injected values

### D. Independent injected values

Score on testing: 0.9999833330555509

## II. K-NEAREST NEIGHBORS

In order to compare with Chatterjee et al (2020), we are only using 5 features, the independent variables:

$$\left[m_1, m_2, \chi_1, \chi_2, \mathrm{SNR}\right].$$

The study is made over the recovered values from GstLAL. The mean score is computed by training the algorithm on the 90% of the dataset and testing it on the remaining 10%, cycling the train/test combination over the full dataset. To do that, we are going to use the training dataset, sicne it's the larger one. In order to train and test the model and create the different plots, we are going to use the training and testing files. First of all, let's check the performance of the algorithm using the same parameters as in Chatterjee et al.

### A. Using paper's hyperparameters

In this case, the number of neighbors is fixed to be $K = 11$ (twice the number of features plus one). The metric usd is the Mahalanobis metric, but it is said that

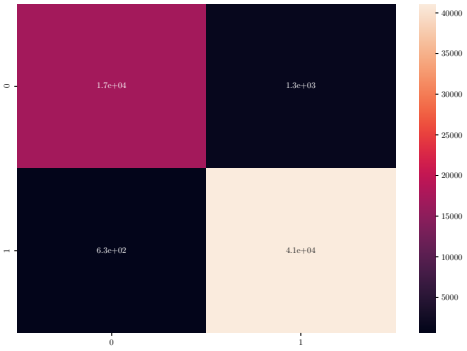


FIG. 5. Confusion matrix for the model used in Chatterjee et al (2020), using the independent recovered values.

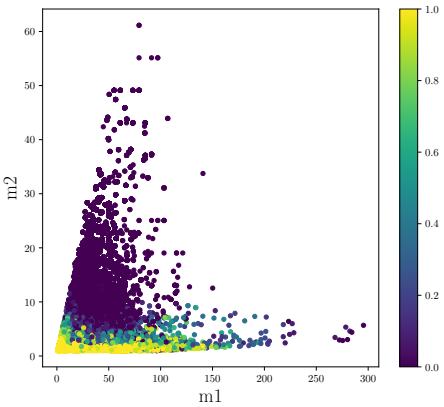


FIG. 6. Probability of having a remnant as a function of the values of the masses.

the metric used doesn't make too much difference. Points are weighted by the inverse of their distance. The mean score is:

$$\mathrm{s_m} = 0.9668068999826518\,,$$

and the score on testing:

$$\mathrm{s_t} = 0.9673161219353655\,.$$

The confusion matrix can be seen in Fig. 5. We have also plotted the $m_1$-$m_2$ plot with the probability of having a remnant, in Fig. 6, also with $q$ and $m_1$ in Fig. 7, and the true vs false positive rates with the corresponding threshold value for the probability in Fig. 8.

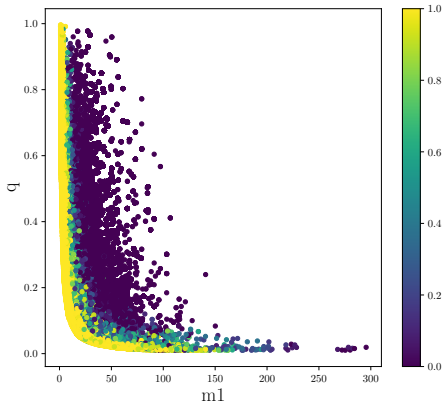


FIG. 7. Probability of having a remnant as a function of $m_1$ and the mass ratio of the binary, $q$.

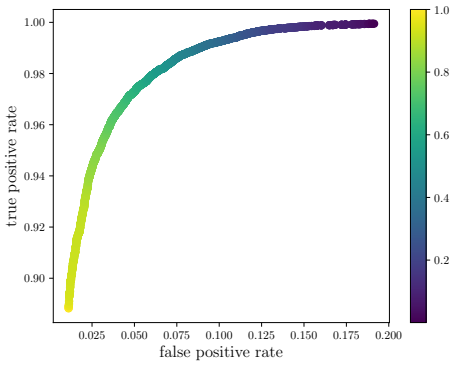


FIG. 8. Relation of the true and false positive rates as a function of the threshold applied to make the decision betweenhaving or not having a remnant.

### B. Using our hyperparameters

In our case, the metric used to compute the distance between points is the *Manhattan* metric, weighting the points uniformly. After applying cross validation, we get that the optimal number of neighbors is $K = 10$, with a mean score of:

$$\mathrm{s_m} = 0.9718355224352762\,,$$

and the score of testing

$$\mathrm{s_t} = 0.9723828730478842\,.$$

In Fig. 9 you can find how the mean score changes with the number of neighbors of the algorithm. The confusion matrix appears in Fig. 10, the probability as a function of $m_1$ and $m_2$ and $m_1$ and q is shown in Figs. 11, 12, respectively, and the true and false positive rates in terms of the threshold probability appear in Fig. 13.

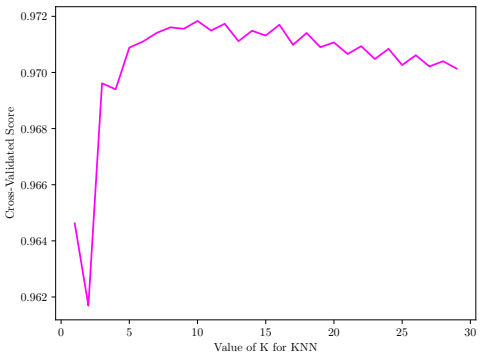


FIG. 9. Score of our model as a function of the number of neighbors.

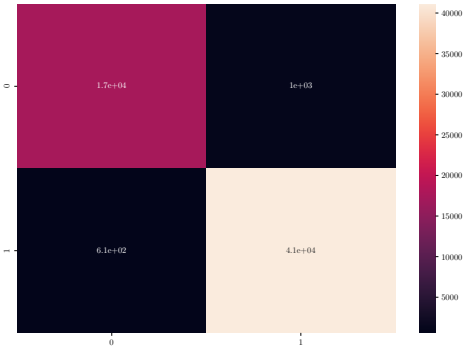


FIG. 10. Confusion matrix for our model, using the independent recovered values.

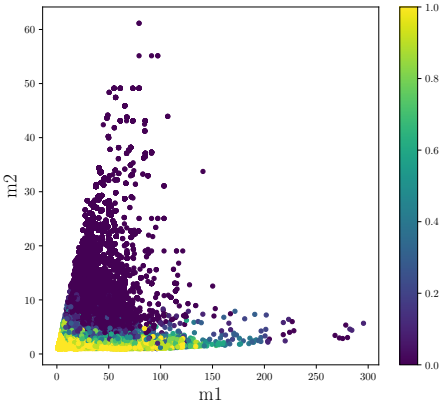


FIG. 11. Probability of having a remnant as a function of the values of the masses.

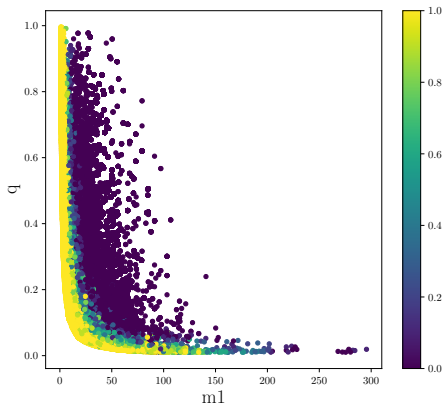


FIG. 12. Probability of having a remnant as a function of $m_1$ and the mass ratio of the binary, $q$.

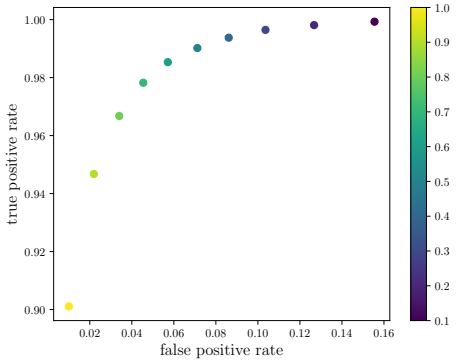


FIG. 13. Relation of the true and false positive rates as a function of the threshold applied to make the decision between having or not having a remnant.