DATA AND INFORMATION QUALITY PROJECT REPORT

PROJECT ID: 26

PROJECT NUMBER:1

ASSIGNED DATASETS: cancer, mushrooms

STUDENTS: Giovanni Chiurco 10659071 - Matteo Braceschi 10662497

ASSIGNED TASK: Classification

1. Cancer

The “cancer” dataset comes from the University of Wisconsin–Madison Hospitals and is made up of 9 numerical attributes ('Clump_Thickness', 'Uniformity_of_Cell_Size', 'Uniformity_of_Cell_Shape', 'Marginal_Adhesion', 'Single_Epithelial_Cell_Size', 'Bare_Nuclei', 'Bland_Chromatin', 'Normal_Nucleoli', 'Mitoses') and a target column (‘Class’) with only two classes, 2 and 4. The features have integer values that are already normalized within the interval [1-10].

1.1 Setup choices

For the classification task on this dataset, we chose Logistic Regression and Support Vector Classifier (SVC), which have proven to be the most successful without much hyperparameter tuning. To evaluate them we used the metrics tailored for a classification task, such as Accuracy, Recall, Precision, F1 score and ROC AUC (Area Under the Receiver Operating Characteristic Curve).

Between the standard imputation techniques seen during the lectures we tried to replace the missing values in each column with the mean, the median and the mode, but at the end we chose to use the “mean” method. Instead, for the advanced techniques we tried k-Nearest-Neighbors imputer, MICE and missForest (Random Forest), but we finally used MICE, which performed better in predicting the missing values.

1.2 Pipeline implementation

After doing data exploration and data profiling on the cancer dataset, to better understand the distribution of the values for each feature, we apply on it the provided script to inject random missing values in various percentages, creating a vector of five new versions of the original dataset with 50%, 40%, 30%, 20% and 10% of missing values, respectively. Then we follow two roads: assess the accuracy of the prediction of the missing values from the

imputers and evaluate the performance of the classifiers with the original dataset and the imputed ones.
For the first step we decided to estimate the soundness of the imputation measuring the Mean Square Error (MSE) between the original values, removed by the provided script, and the imputed values, because the cancer dataset have numerical values and imputing the missing values with the mean or with the MICE technique means that the substituted values are real ones, instead of integer from 1 to 10. In the table below there are the results of the accuracy assessment on the cancer dataset. It is important to notice that the MSE computed on the datasets imputed with the so called simple techniques, mean, median and mode, are independent from the percentage of missing values, instead when we use the advanced ones we notice that the MSE decreases with the decrease of the number of missing values. So, techniques such as MICE, KNN imputation and missForest are more precise than replacing with the mean or with the mode according to the MSE.

	50%	**40%**	**30%**	**20%**	**10%**
	MSE	**MSE**	**MSE**	**MSE**	**MSE**
mean	**7.928092**	**7.919333**	**8.306705**	**7.883291**	**7.991496**
median	10.381647	10.191653	11.426139	10.500431	10.76699
mode	12.253243	11.865883	12.902252	12.107666	12.063107
KNNImputer	4.987844	4.674644	4.465524	3.659087	3.131133
missForest	4.506497	4.282508	3.924866	3.723836	3.084952
MICE	**4.223982**	**3.995474**	**3.522652**	**3.078966**	**2.810502**

In the second step, to evaluate the performance of the classifiers we used k-fold cross validation on each dataset .

1.3 Results

In the table below there are the results of the Logistic Regression and SVC models with the mean and the MICE imputations for the various percentages of missing values:

Performance of Logistic Regression with mean imputation

	10%	**20%**	**30%**	**40%**	**50%**
Accuracy	0.971	0.968	0.958	0.947	0.941
F1	0.968	0.965	0.953	0.942	0.934
Precision	0.968	0.965	0.955	0.944	0.941

ROC AUC	0.995	0.995	0.992	0.988	0.988
Recall	0.969	0.966	0.953	0.940	0.930

Performance of Logistic Regression with MICE imputation

	10%	**20%**	**30%**	**40%**	**50%**
Accuracy	0.974	0.968	0.959	0.934	0.939
F1	0.972	0.965	0.955	0.934	0.933
Precision	0.971	0.966	0.956	0.937	0.938
ROC AUC	0.996	0.992	0.989	0.979	0.979
Recall	0.972	0.964	0.955	0.932	0.929

Performance of SVC with mean imputation

	10%	**20%**	**30%**	**40%**	**50%**
Accuracy	0.968	0.966	0.950	0.956	0.94
F1	0.965	0.963	0.946	0.952	0.934
Precision	0.963	0.962	0.944	0.953	0.937
ROC AUC	0.989	0.989	0.980	0.983	0.979
Recall	0.968	0.965	0.948	0.951	0.932

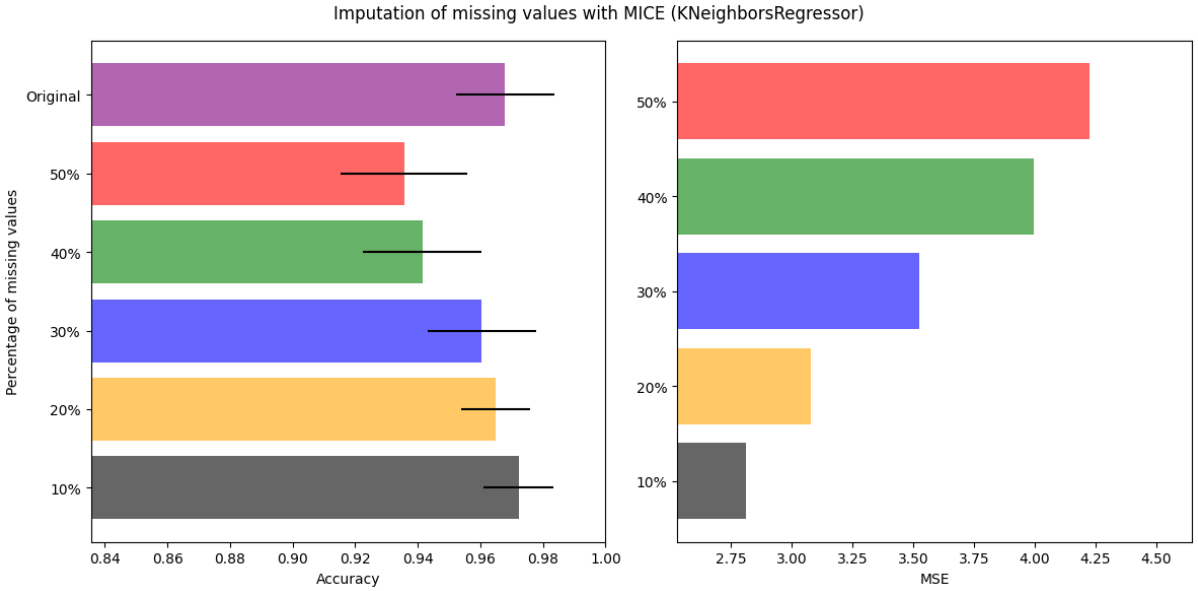
Performance of SVC with MICE imputation

	10%	20%	30%	40%	50%
Accuracy	0.966	0.963	0.965	0.95	0.941
F1	0.963	0.959	0.962	0.945	0.935
Precision	0.962	0.96	0.961	0.945	0.939
ROC AUC	0.991	0.985	0.979	0.966	0.963
Recall	0.964	0.96	0.963	0.946	0.932

Results of the classifiers for the original dataset

	Accuracy	Recall	Precision	F1	ROC AUC
Logistic Regression	0.968	0.965	0.965	0.965	0.996
SVC	0.965	0.962	0.961	0.961	0.989

As we notice from the above tables the Logistic Regression model gave the best results with respect to more complex algorithms, such as SVC, with the smaller percentages of missing values, like 10% and 20%. While the SVC performed better when the dataset had more missing values, from 30% to 50%. Certainly, in these last cases the accuracy of the classifiers decreased compared to the accuracy with the original dataset, i.e. even if the dataset had few features and data, when it was missing more than the 30% of values, it was more difficult for the ML models to predict. As in the case of the accuracy, also with the other metrics the performances were reduced as the number of missing values increased.



Accuracy of the Logistic Regression model on the original and the imputed dataset (on the left) and the MSE between the original and imputed values (on the right)

2. Mushrooms dataset

The dataset is structured by some nominal categories, and the goal of the project is to correctly predict the class of the imputed dataset (previously modified by removing values). The classes of this dataset are only ‘Poisonous’ and ‘Edible’.

2.1 Setup choices

For the imputation part, we selected the standard technique **Mode Imputation**, that is the most-frequent element to substitute the missing value with the most-frequently occurring element in the feature. It is important to consider the limitations of the most-frequent imputation technique when using it to handle missing values in a categorical dataset. This technique may introduce bias if the missing data is not randomly distributed and the most frequent value is not representative of the population. Hence, in order to improve the score, we also tried to use other advanced imputation techniques such as KNN (K-Nearest Neighbors) and Miss Forest. We select **Miss Forest** because it provides the best result in the accuracy assessment.

The chosen ML algorithms for the classification are **SVC (Support Vector Classifier)** and **GradientBoostingClassifier** from scikit-learn library. Choosing these algorithms can allow us to do comparison and evaluation of their performance on the same dataset, and moreover they have shown the best results in terms of accuracy.

After that, we evaluate them with other metrics. Some common evaluation metrics for classification include accuracy, precision, recall, f1 score and ROC AUC (Area Under the Receiver Operating Characteristic Curve).

2.2 Pipeline implementation

As in the previous dataset, we do data exploration and data profiling, to visualize the distribution of the values for each feature. Then we inject the missing values in various percentages, creating a vector of five new versions of the original dataset with 50%, 40%, 30%, 20% and 10% of missing values, respectively.
Afterward, we need to use an encoder in order to correctly manage the categorical dataset, we use an Ordinal Encoder from the scikit-learn library.
Once the preprocessing is finished, we start to assess the accuracy of the prediction of the missing values, done by different imputation techniques and evaluate the performance of the classifiers with the original dataset and the imputed ones.

2.3 Results

For the accuracy assessment, we compute the accuracy score as:

$$count\_of\_correct\_values \ / \ total\_count\_of\_values$$

and from that we notice that different imputation techniques may yield different results, for that reason it is important to evaluate the performance of the model. It is important to consider the trade-offs between this evaluation and choose the imputation technique that best fits the needs of the analysis.

	50%	**40%**	**30%**	**20%**	**10%**
	Accuracy score	**Accuracy score**	**Accuracy score**	**Accuracy score**	**Accuracy score**
mode	76.732%	83.530%	84.491%	91.694%	95.925%
KNNImputer	83.924%	90.170%	90.294%	95.547%	97.709%
MissForest	86.373%	91.253%	90.975%	95.825%	97.919%

Intuitively, the standard technique performs worse than the other advanced techniques, and also this fact is highlighted as the percentage of missing values increases. On the other hand, with a low level of percentage, the accuracy score is similar among all the classifier.

In addition, comparing the performance of two machine learning algorithms on the same imputed dataset can provide insight into which algorithm is more suitable for the task. Understanding the strengths and weaknesses of different algorithms and imputation techniques can help in selecting the most appropriate approach for a classification task.

Another evaluation is about the performance of the same ML algorithm with dataset imputed with different imputation techniques. Below there are the tables with all the evaluations.

PERFORMANCE OF ML ALGORITHMS WITH **MODE IMPUTATION**

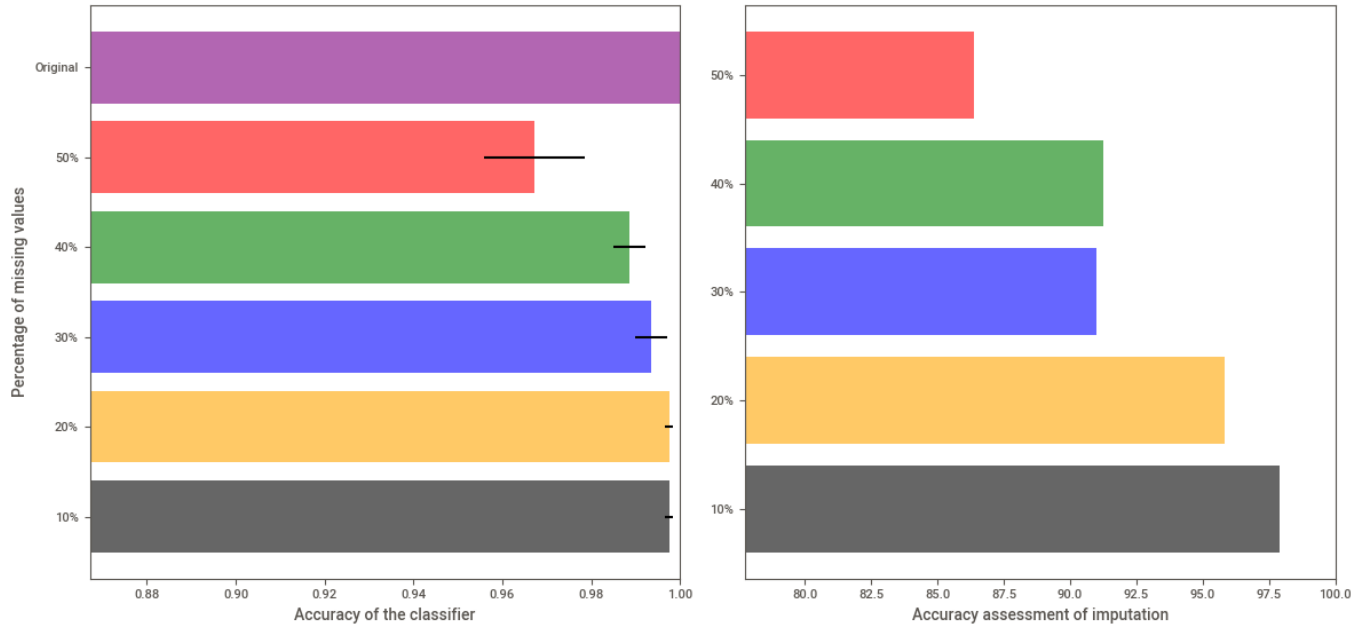
	50%	**40%**	**30%**	**20%**	**10%**
	Accuracy score	**Accuracy score**	**Accuracy score**	**Accuracy score**	**Accuracy score**
Gradient Boosting	0.919 ± 0.008	0.960 ± 0.005	0.980 ± 0.002	0.985 ± 0.007	0.994 ± 0.001
Logistic Regression	0.831 ± 0.023	0.859 ± 0.010	0.870 ± 0.018	0.893 ± 0.010	0.919 ± 0.008
SVC	0.849 ± 0.016	0.879 ± 0.012	0.895 ± 0.017	0.924 ± 0.011	0.954 ± 0.003

PERFORMANCE OF ML ALGORITHMS WITH **MISS FOREST IMPUTATION**

	50%	40%	30%	20%	10%
	Accuracy score	**Accuracy score**	**Accuracy score**	**Accuracy score**	**Accuracy score**
Gradient Boosting	0.967 ± 0.011	0.989 ± 0.004	0.994 ± 0.004	0.998 ± 0.001	0.998 ± 0.001
Logistic Regression	0.929 ± 0.010	0.940 ± 0.012	0.937 ± 0.008	0.932 ± 0.007	0.934 ± 0.006
SVC	0.944 ± 0.011	0.963 ± 0.009	0.963 ± 0.007	0.969 ± 0.005	0.966 ± 0.010

In our case the Gradient Boosting algorithm provides the best results, but as we can see in the table above, the advanced techniques of imputation Miss Forest improve the accuracy, particularly in the case where there is a high percentage of missing values.

Moreover, in the plot below there are all the horizontal bars for each percentage of the accuracy of Gradient Boosting Classifier with Miss Forest Imputation. On the left there is the accuracy of the classifier compared to the original one (purple), while on the right there is the accuracy assessment.



Once we find the best classifier and imputation method for our case, it's also important to evaluate the performance of the model using appropriate evaluation metrics.

In a classification task performed on an imputed dataset using two different imputation techniques, the resulting model performance may differ. One technique may yield a higher accuracy, while the other may have a higher precision or recall. It is important to consider the trade-offs between these evaluation metrics and choose the imputation technique that best fits the needs of the analysis. Understanding the strengths and weaknesses of different algorithms and imputation techniques can help in selecting the most appropriate approach for a classification task.

Here we can see all the results with some different evaluation metrics.

PERFORMANCE OF GRADIENT BOOSTING WITH MODE IMPUTATION

	10%	**20%**	**30%**	**40%**	**50%**
Accuracy	0.994031	0.98507	0.979602	0.960204	0.919409
F1	0.994026	0.985044	0.979581	0.960104	0.919237
Precision	0.994029	0.985515	0.979722	0.961478	0.920436
ROC AUC	0.999798	0.998123	0.995775	0.992225	0.975738
Recall	0.994027	0.984831	0.979511	0.959591	0.918894

PERFORMANCE OF GRADIENT BOOSTING WITH MISS FOREST IMPUTATION

	10%	**20%**	**30%**	**40%**	**50%**
Accuracy	0.998507	0.997511	0.996021	0.988063	0.96617
F1	0.998506	0.997508	0.996017	0.988045	0.96612
Precision	0.998554	0.997572	0.996063	0.988402	0.966646
ROC AUC	0.999988	0.999988	0.999456	0.997351	0.991009
Recall	0.998463	0.997466	0.995991	0.987848	0.965891

PERFORMANCE OF SVC WITH MODE IMPUTATION

	10%	20%	30%	40%	50%
Accuracy	0.95373	0.92388	0.894533	0.878596	0.848763
F1	0.953632	0.923371	0.894042	0.87781	0.848417
Precision	0.954705	0.92937	0.897936	0.883591	0.850256
ROC AUC	0.990939	0.976173	0.957913	0.94003	0.910437
Recall	0.953246	0.922366	0.893425	0.87705	0.848333

PERFORMANCE OF SVC WITH MISS FOREST IMPUTATION

	10%	20%	30%	40%	50%
Accuracy	0.982584	0.980596	0.982087	0.975126	0.948268
F1	0.982563	0.980578	0.982059	0.975093	0.948171
Precision	0.982818	0.980653	0.982472	0.975403	0.94911
ROC AUC	0.998461	0.997627	0.997822	0.994543	0.978384
Recall	0.982414	0.980567	0.981845	0.974904	0.947799

3. Conclusion

To conclude, the data imputation is an important technique for dealing with missing or incomplete data. In our two cases, we must decide the correct imputation for a specific dataset because each technique can be applied depending on the specific characteristics of the dataset and on the task. To respect this, we have used different methods in order to choose the best one and report the uncertainty of the imputed values in the final analysis. Our experiments demonstrated that more advanced imputation techniques, such as MICE or missForest, give better results compared to simpler techniques like imputing missing values

with the mean or the mode, although the more advanced ones are more time consuming and costly.