1. **The name or where the dataset is from** = Hospital Israelita Albert Einstein, at São Paulo, Brazil.

# **Albert Einstein Hospital, Brazil Dataset**

This dataset contains anonymized data from patients visited at the Hospital Israelita Albert Einstein in So Paulo, Brazil, who had samples collected to perform the SARS-CoV-2 RT-PCR and other laboratory tests during their visit.

All data were anonymized in accordance with international best practices and standards.

## **Our Modeling**

The Albert Einstein Hospital dataset consisted of 5644 samples and, 111 variables. The variables are divided into 4 dependent variables and 107 independent variables, the dependent variables are ‘SARS-Cov-2 exam result’, ‘Patient admitted to regular ward’, ‘Patient admitted to semi-intensive unit’ and ‘Patient admitted to intensive care unit’. In our implementation we only used 1 dependent variable ‘SARS-Cov-2 exam result’ to predict COVID-19 infection.

Points analyzed from the dataset:

* Large number of missing values.
* Most features are Imbalanced
* Imbalanced positive and negative cases.

**Missing Values**

Other than ‘Patient Age quantile’ and ‘SARS-Cov-2 exam result’, all features have missing values. The least feature with missing values is 76.01% empty, that is 1354 out of 5644. See Figure 1 for a chart of how much values are there in each feature. Our approach into dealing with missing values, 1) we first shrunk the dataset approximately to the least feature with missing values ‘Parainfluenza 1’ which results into 1352 samples dataset. Figure 2 presents the new features value count. 2) then we dropped features with more than 75% missing values and deleted 1 feature with only 1 unique value (zero-variance feature), which leaves us a new dataset with 1352 samples and 31 features. 15 of the 31 features are categorical and have no missing values. 3) finally, we used KNN imputation with k=2 to fill the missing values, for numerical features using the mean strategy.

Also, after the imputation, we transformed categorical features using dummy encoding, and standardized all features to have a mean of 0 and a variance of 1.

**Imbalanced features**

As for imbalanced features, we in this implementation didn’t use any techniques regarding this issue. But as illustration of how the features are imbalanced, we created Table 1 for categorical features. As for numeric features they have a strange value distribution for instance the ‘Red blood cell distribution width (RDW)’ feature has 55 different values and, from the 1352 samples 986 of them are for one value, this issue is present in all of the numeric features except ‘Patient age quantile’ which has a fair value distribution.

**Imbalanced Target Variable**

The shape of the dataset after the above preprocessing is now 1352 samples and 47 features. The target variable has a class distribution of 1240 negative samples and 112 positive samples, which is very imbalanced, and training a model on a highly imbalanced set will only lead to a model that predicts the minority class poorly or simply a biased-model.

So, in this implementation we created 11 splits of the dataset every set has all the 112 positive samples and a unique 112 negative samples. We ran 7 classifiers and used 10-fold cross validation to test and train the models. Each model is trained and tested on each split of the dataset, and every split produces 10 results using 10-fold cross validation, from the 10 results we take the average which then gives us the average result for one split, this is done for every split which means 11 results will be there, we take the average of the 11 splits to produce the finale result for a specific classifier. See Table 2 for models results.

A picture containing chart

Description automatically generated

Figure : Bar chart showcasing features value count.

Chart

Description automatically generated

Figure : Bar chart of the features value count after shrinking the dataset.

Table 1: Summary of features and their value distribution.

|  |  |  |
| --- | --- | --- |
| Feature | Detected | Not detected |
| Respiratory Syncytial Virus | 52 (3.8%) | 1300 (96.2%) |
| Influenza A | 18 (1.3%) | 1334 (98.7%) |
| Influenza B | 76 (5.6%) | 1276 (94.4%) |
| Parainfluenza 1 | 3 (0.2%) | 1349 (99.8%) |
| CoronavirusNL63 | 45 (3.3%) | 1307 (96.7%) |
| Rhinovirus/Enterovirus | 379 (28%) | 973 (72%) |
| Coronavirus HKU1 | 20 (1.5%) | 1332 (98.5%) |
| Parainfluenza 3 | 10 (0.7%) | 1342 (99.3%) |
| Chlamydophila pneumoniae | 9 (0.7%) | 1343 (99.3%) |
| Adenovirus | 13 (1%) | 1339 (99%) |
| Parainfluenza 4 | 19 (1.4%) | 1333 (98.6%) |
| Coronavirus229E | 9 (0.7%) | 1343 (99.3%) |
| CoronavirusOC43 | 8 (0.6%) | 1344 (99.4%) |
| Inf A H1N1 2009 | 98 (7.2%) | 1254 (92.8%) |
| Bordetella pertussis | 2 (0.2%) | 1350 (99.8%) |
| Metapneumovirus | 14 (1%) | 1338 (99%) |

Table 2: Models performance

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Average Precision | Average Recall | Average F1-Score |
| Logistic Regression (LR) | 79% | 76% | 75% |
| Support Vector Machine (SVM) | 81% | 77% | 77% |
| K-Nearest Neighbors (KNN) | 76% | 74% | 73% |
| Gaussian Naive Bayes (GNB) | 76% | 60% | 53% |
| Decision Tree (DT) | 74% | 72% | 72% |
| Random Forest (RF) | 78% | 75% | 75% |
| Extreme Gradient Boosting (XGBoost) | 78% | 76% | 75% |

Table 3 below presents some results of related work done using the same dataset, compared with ours.

Table 3: Comparison of our model with other existing models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reference | Features | Models used | Precision | Recall | F1-Score |
| [1] | 25 | Stacked Ensemble Machine Learning (See Table 4) | 100% | 100% | 100% |
| [2] | 24 | SVM, MPL, RF,  Naive Bayes,  Bayesian network | 93.77% | 96.76% | 93.80% |
| [3] | 14 | CR | 83.70% | 84.20% | 83.70% |
| Couldn’t find the paper, but was cited by [1] | 15 | SVM, RF, DT | - | - | 92.00% |
| Our model | 31 | LR, SVM, KNN, GNB, DT, RF, XGB | 78% | 74% | 73% |

Table 4: Performance Metrics of Stacked Ensemble Model. [1]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Layer-1 | | | | Layer-2 |
| Performance Metrics | KNN | SVM | XGB | RF | AdB |
| Precision | 98.73% | 96.84% | 97.47% | 99.37% | 100% |
| Recall | 77.23% | 90.00% | 93.33% | 95.15% | 100% |
| F1-Score | 86.67% | 93.29% | 95.36% | 97.21% | 100% |

# **Italy Dataset**

Qfesflskrl

# References

|  |  |
| --- | --- |
| [1] | M. M. U. K. L. A. A. B. S. Md. Mohsin Sarker Raihan, "https://arxiv.org/," [Online]. Available: https://arxiv.org/pdf/2108.05660v2.pdf. |
| [2] | J. C. G. M. A. d. S. J. E. d. A. A. R. G. d. S. R. E. d. S. W. P. d. S. Valter Augusto de Freitas Barbosa, "medrxiv," [Online]. Available: https://www.medrxiv.org/content/10.1101/2020.05.14.20102533v1.full.pdf. |
| [3] | S. H. M. A.-E. M. N. A.-K. M. P. Ibrahim Arpaci, "PubMid," [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33437173/. |