Stroke predictions
Final Project, Machine Learning for Biomedical Applications

Judith Camacho, 218863
David Legarre, 218911

June 20, 2021

Contents

Background

We wanted to apply the concepts learnt through the course to some medical "real" examples.

After doing some research on internet, we found the following dataset on kaggle.com:
`https://www.kaggle.com/fedesoriano/stroke-prediction-dataset`

This dataset is used to predict whether a patient is likely to get a stroke based on the input. Each row in the data is relevant information from a patient.

Please take a look at the script Stroke_Prediction to see all the graphs, confusion matrices and detailed code.

1 The Data

Our dataset has the following structure:

1. **Id**: Unique Identifier of the patient, **numerical value**
2. **Gender**: "Male", "Female" or "Other", **categorical value**
3. **Age**: Age of a patient, **numerical value**
4. **Hypertension**: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension, **categorical value**
5. **Heart_disease**: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease, **categorical value**
6. **ever_married**: "No" or "Yes", **categorical value**
7. **work_type**: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed", **categorical value**
8. **residence_type**: "Rural" or "Urban", **categorical value**
9. **avg_glucose_level**: Average glucose level in blood, **numerical value**
10. **BMI**: Body Mass Index, **numerical value**
11. **Smoking_status**: "formerly smoked", "never smoked", "smokes" or "Unknown", **categorical value**
12. **Stroke**: 1 if the patient had a stroke or 0 if not, **categorical value**

In total we have in the dataset: 4 numerical variables and 7 categorical variables.

2 Data visualization

In our notebook we have implemented all visualization of our data, here we are going to show a few since it would take a lot of space to show all of our plots.

First of all, our dataset consists of 5110 entries and 11 features, 12 counting **id**. In the case of the numerical data, we can see that most features have different ranges and distributions, so we'll need to standardize the data, which we'll explain later in section Standardize the data.

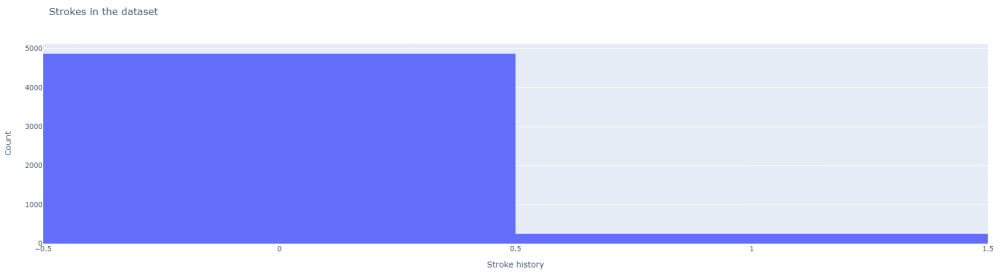

Figure 1: Histogram plot of the stroke column of the dataset

As we can see in figure [1] our data is highly unbalanced, there 4861 no strokes and 249 strokes, we'll explain how we dealt with this in section Data Preprocessing

We also wanted to know characteristics of only people with stroke. We see that 141 are female and 108 are men, which are close numbers, and we were expecting a huge difference between genders.

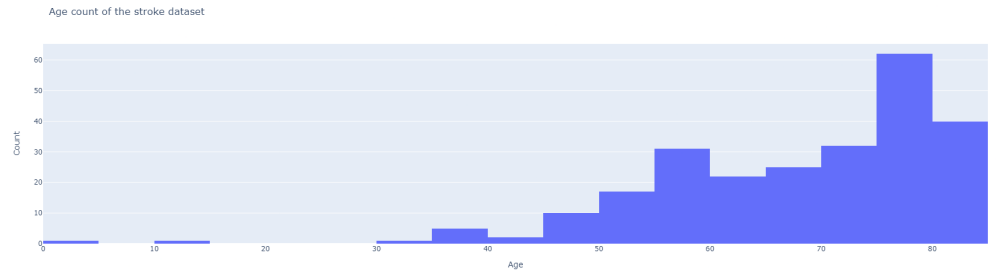

Figure 2: Histogram distribution of ages of people, who have had a stroke

As we can see in [2] mostly people over the age of 70 are the most common age range. There is also a peak at around 80 and then decreases probably because of the average years of life.

We have extracted the average glucose level of people with stroke, and it is 132.54 mg/dL, which is higher than the normal values. Of 249 people, 66 of them have hypertension and 47 have heart disease.

3 Data Preprocessing

.

3.1 Cleaning NULL values

First of all, while looking at the data, we noticed that the feature **BMI** is lacking some entries, 201 in total compared to the total of 5110 entries

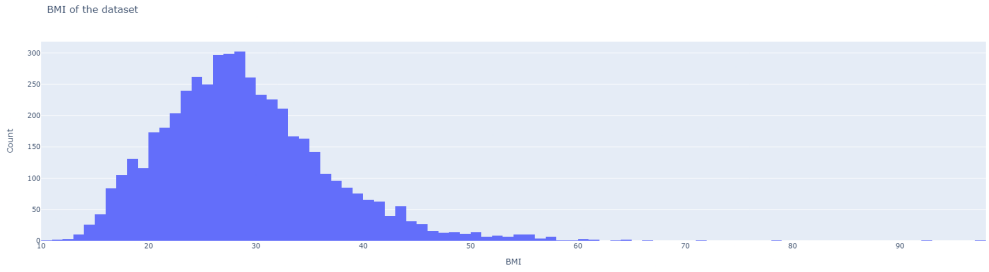

Figure 3: Histogram plot of the BMI column of the dataset

In this distribution, given that BMI is a numerical variable, we can see that most values lie in the range $(20, 40)$ and it only has one mode, so we decided to give this missing values the value of the average BMI in the dataset.

3.2 Categorical data to dummy variables

As we have shown before, there are 5 categorical features that do not use numbers to represent their value, so we have to encode these features with dummy variables.

3.3 Standardize the data

Now we standardize the data. The goal of this is to change the values of the numeric columns to a common scale but without distorting the differences in the ranges of values.

We have decided to use standardscaler() from the Sklearn.preprocessing library, because from the previous plots we know that our features approximate to Gaussian distributions.

We perform data scaling because if we use distance based models, this will prevent features with wide ranges from dominating the distance metric.

3.4 Correlation of the variables

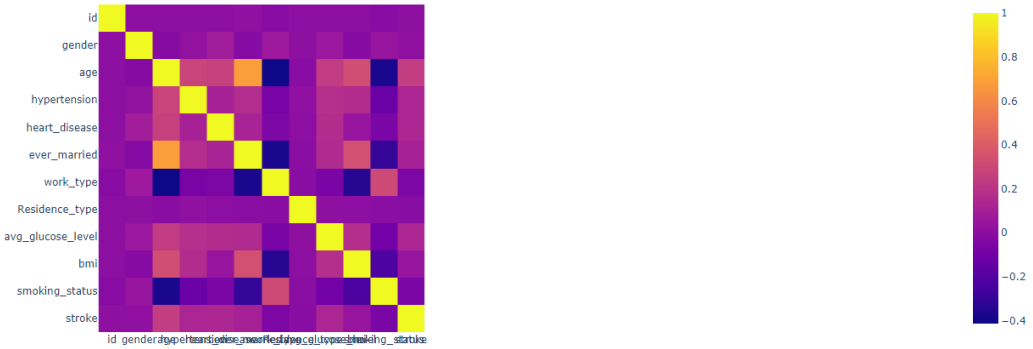

Figure 4: Correlation matrix between the variables

When two features are negatively correlated, it means that the relationship between them is opposite all the time and otherwise when they're positive.

Now we are going to take a look at how stroke is correlated with other features:

- **Age**, as we've seen before the stroke age group is biased towards the largest values.
- **ever_married**, because as age increases, people are more likely to be married.
- **Hypertension**, this is another condition that may have an effect on the chances of having a stroke.
- **Heart_disease**, dangerous condition overall

- **work_type**, seems also to have some of a correlation.

But overall, in our dataset, there's a low correlation between stroke and residence_type, smoking_status, work_type, BMI and gender. Also there are other correlations between other variables such as age and ever-married (older people are more likely to be married).

We also see no strongly correlated features, neither positive nor negatively, with stroke. The biggest correlation value is 0.245257 and it is the correlation between stroke and age.

3.5 Balancing the data

Given the nature of our dataset, generating new samples is complicated, since it was taken from a published one in the internet we can't resample, so we decided to create a new dataset taking a random sample from the non-stroke entries and having in the end the same number of strokes and no strokes in this new dataset. We tested our models with both datasets, **unbalanced** and **balanced** so that we can compare the two and see which one performs better.

4 Modelization without feature selection

First we decided to test three models with the data we have until now. We chose a random forest, a SVM and an ANN for classification. Random forest is chosen because it's an ensemble model easy to understand and avoids overfitting. ANN was chosen because even though they can be as black-boxes they are able to find non-linear relationships between variables, which we think is very useful for medical purposes due to the complexity of biology. Notice that we decided to initially tune the parameters of the models and it was done empirically.

4.1 Results

4.1.1 Unbalanced data

	Accuracy	**Recall**	**Precision**
Random Forest	0.9488	0.0120	0.1579
SVM	0.9513	0.0	0.0
ANN	0.9419	0.0	0.0

Seeing these results for the recall, we immediately decided to balance the data, even before doing other improvements.

4.2 Balanced data

	Accuracy	**Recall**	**Precision**
Random Forest	0.7590	0.7148	0.7448
SVM	0.4899	0.3012	0.4839
ANN	0.4800	1.0	0.4800

As we can see the accuracy has decreased. One might think at the beginning that in fact, the model is worse, but we have to take a look at the recall and precision. They have significantly increased, which means that the models are learning something. Before they predicted almost all input as no stroke due to the lack of stroke samples.

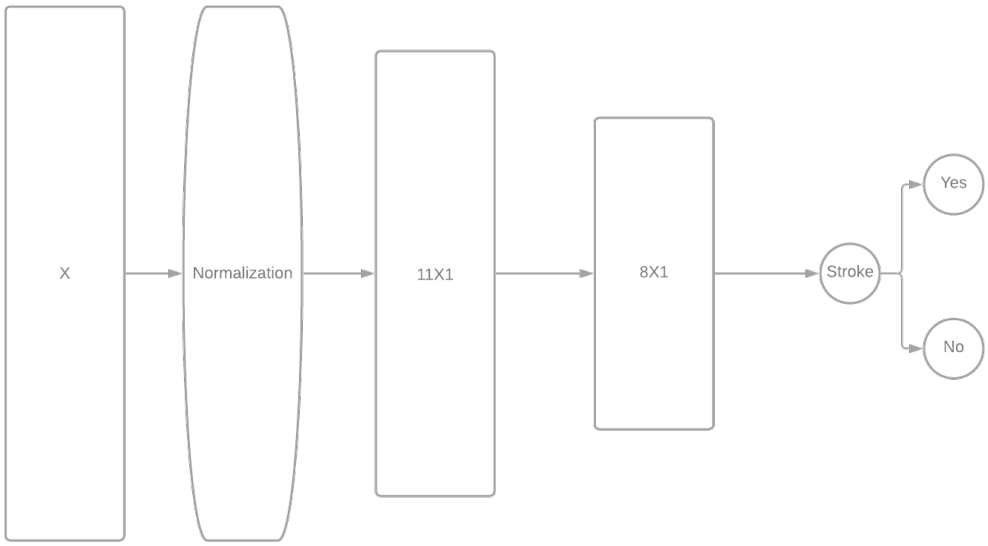

Figure 5: Structure of our ANN

As we can see in [5] our ANN is formed by a normalization layer, that normalizes the data so that the loss function uses normalized values, otherwise our loss function would be returning values in the range of $(100, 400)$. Followed by two dense connected layers of sizes 11 and 8 respectively. The first layer is formed by 11 since our data consists of 11 features in this case. Then a single neuron is used to output the prediction, "Yes" or "No" stroke.

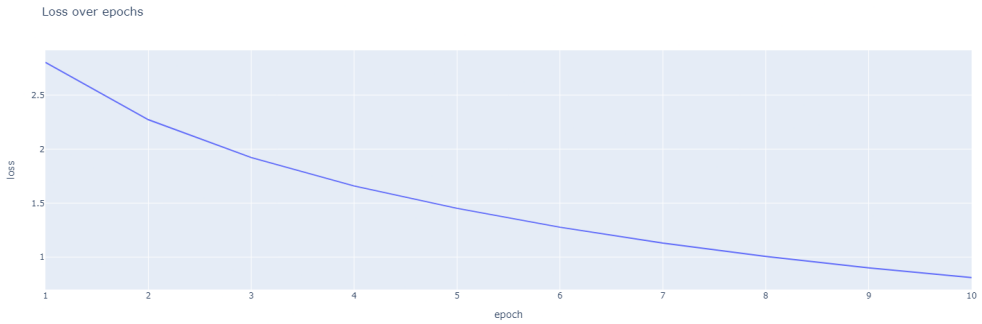

Figure 6: Loss over epoch during training

5 Feature selection

In order to improve our model, we tried see how many variables we needed to explain most of the variance in the dataset.

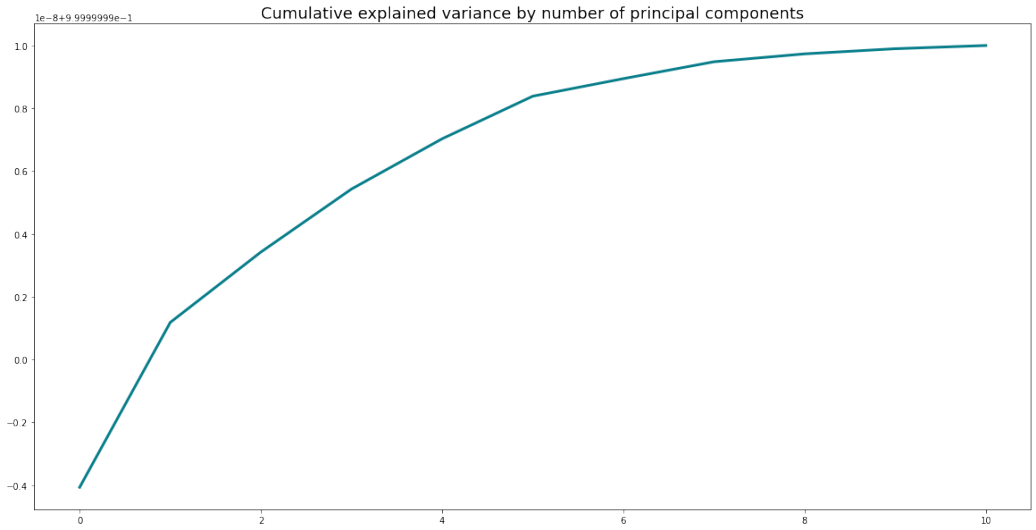

Figure 7: % Variance explained over number of variables

This way, we can see how many features are really meaningful. We can see that, with the first 7 principal components, we get almost 90% of variance of the data. Hence, now we are going to transform the data to the projection of these 7 components and see the new performance of the models. As we have shown before most likely the 4 components are the ones with a correlation above or under 0.0 with respect to stroke.

6 Final models

We retrained our models with class imbalance and with the balanced dataset this time using only the 7 features with the most explained variance.

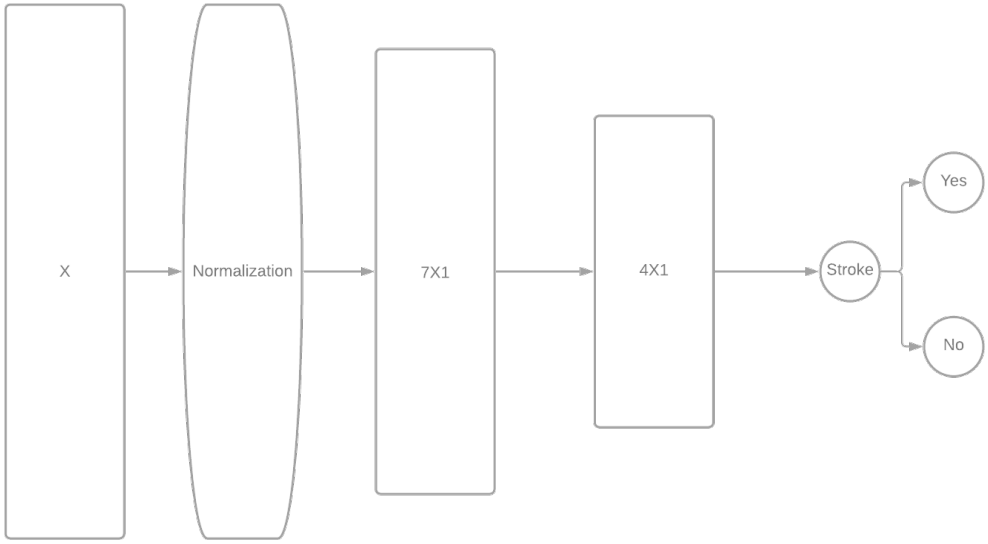

Figure 8: Structure of our ANN after PCA

When using the reduced data by PCA, we had to adapt the model to the new dimensions of the input data, as explained before, from 11 features to 7.

6.1 Results

6.1.1 Unbalanced data

	Accuracy	**Recall**	**Precision**
Random Forest	0.9491	0.0080	0.1333
SVM	0.9513	0.0	0.0
ANN	0.9531	0.0	0.0

6.1.2 Balanced data

	Accuracy	**Recall**	**Precision**
Random Forest	0.6887	0.6546	0.7026
SVM	0.5020	0.4859	0.5021
ANN	0.4720	0.4839	0.4688

Comparing to our models before using PCA, we can't see any improvement, we have even lost some recall score in our ANN model. So in general we

consider the use of PCA a failure in our models.

7 Conclusions

In conclusion, we have decided that this dataset is not good enough to create a classifier of future strokes.

We thought that it could be a lack of important biomedical features (proteins/hormones levels, other diseases, etc). Perhaps the current features are not the main cause of strokes in patients. And also we lack data for patients who have suffered strokes, because even if we balanced the data the models still had problems classifying this data.

Perhaps the differences in this dataset between people who have suffered a stroke and those who did not is too small for our models to appreciate, at least as our dataset explains them.

Nevertheless, we consider that none of these models are adequate for this particular biomedical problem due to the large number of false negatives it predicts.A false negative in medicine is really dangerous since it leaves a patient that needs treatment without one. Also notice that in all the scenarios the model with the best performance is the Random Forest (a model easy to understand that usually gives good results very fast). It was able to achieve a precision of 0.7448. Despite this, it could still be leaving at around a $\frac{1}{4}$ of the patients without treatment.