Interview Assignment: Factors Influencing Stroke

Introduction

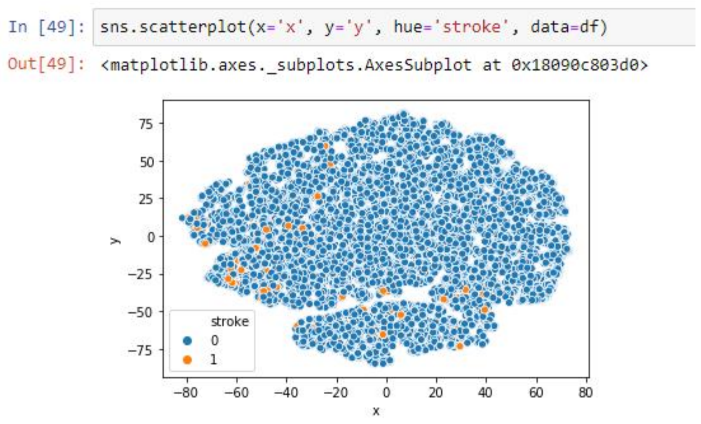
A predictive model is trained on a healthcare dataset containing features such as gender, the presence of heart disease and occupation in order to predict the likelihood of an individual having a stroke.

Exploratory Data Analysis

The dataset was inspected to familiarise the structure of the data. The raw dataset contains 43400 entries including numerical and non-numerical data types. Some of the features contain continuous data, for example: *avg_glucose_level* and *age;* while some of the features contain categorical data such as *gender* and *work_type.* The categorical data is sometimes inherently binary, such as the presence of *hypertension* or *heart_disease*, whereas some are fall into a broader range of categories such as *smoking_status.*

Analysis Selection

Seaborn was used to plot a t-SNE graph to reveal the presence of any natural clusters in the dataset:



```
In [49]: sns.scatterplot(x='x', y='y', hue='stroke', data=df)
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x18090c803d0>
```

As is observed, the datapoints for stroke do not cluster in an obvious way. Because of this, a supervised approach was adopted.

The next question to answer was which type of supervised analysis? This was a categorisation problem, the two categories being *stroke* and *no stroke,* some estimators that perform well in this scenario are SVM, k-Nearest Neighbour or ensemble classifiers such as a random forest. The selection was made ultimately by comparative elimination; random forest is suited to multiclass problems whereas SVM is intrinsically two-class, and k-NN can be sensitive to bad features. Since the classification problem was binary and there were a large number of features, SVM was chosen as the estimator.

Pre-processing

The dataset underwent a dimensional reduction by feature selection.

Gender

Investigation of the data showed that there were three gender classifications: *male, female and other*. Whilst perhaps insensitive, data points containing the *other* classification were dropped for three reasons:

1. There are only 11 data points
2. Each of those datapoints has never had a stroke, so the model may fit to *other* being a significant factor in predicting stroke (or lack thereof)
3. *Other* is ambiguous from a statistical analysis viewpoint and could fit a number of descriptions that we cannot discern from the data.

After this, an OrdinalEncoder was used to convert the Object-type data into integers, where 1 is male and 0 is female.

BMI

3.4% of *BMI* entires were unfilled. This is a small percentage of an already large dataset, so these entries were dropped from the analyses as medical judgement suggests BMI could be a significant measure in predicting stroke.

Another point to consider for BMI is the fact that a 'healthy' BMI isn't necessarily as higher BMI. Rather, one's divergence from a healthy BMI can go one of two ways: lower (underweight) or higher (overweight). Which way you go is likely to affect the kind of health conditions you develop. For example, a person who is obese is more likely to develop Type 2 diabetes than a person who is underweight, who might be more likely to suffer from malnutrition. Because of this, it makes sense to split up the BMI feature into *underweight* and *overweight*, so that the model can assess which of the two is more linked to stroke.

To do this, a 'healthy' BMI of 25 was chosen and for each data point the difference between the actual BMI and the healthy BMI was calculated and labelled 'underweight' and 'overweight' accordingly. An example is seen below:

BMI	**overweight**	**underweight**
35.9	10.9	0.0
25.0	0.0	0.0
23.0	0.0	2.0

A similar result could have been achieved by simply normalising BMI around 25, but since a many of the categorical features were going to be one-hot encoded between 0 and 1, it made sense to avoid having any negative numbers in the BMI feature.

Smoking Data

30.6% of *smoking_status* entries are unfilled, which is a much larger percentage than the previous case. However, intuition here suggests that smoking could be a significant contributor. As stated before, this dataset is already quite large, so dropping 30% of the data points still leaves a large dataset. With this in mind, the decision was taken to drop the unfilled data points.

Smoking data is classified into three discrete categories: never smoked, formerly smoked and smokes. The data may be discrete, but there is an inherent hierarchy to it when considering health implications: smoking is worse for you then formerly smoking which is worse than never smoking. This suggests that the data is ordinal in nature. Because of this, an OrdinalEncoder from the scikit-learn Python package was used to encode the data so that never smoked is ranked lowest at 0 and currently smokes is ranked highest at 2. This ensures that when fitting the model, the 'healthiness' hierarchy of smoking is preserved.

Other Categorical Features

Residence and *work_type* were then the only two remaining features in need of pre-processing. Since both are categorical and there is no obvious hierarchy, the data is Nominal. To get this in a form that could be understood by a machine learning model, the features were one-hot encoded. This was done with the *.get_dummies()* method for Pandas DataFrames. Since there were only two residence types, one of the one-hot encoded columns were dropped, since a 1 in the *urban* column implies a 0 in the *rural* column and vice-versa. *Ever_married* was also dropped from the feature set, as this is unlikely to be a related factor.

Dealing with Imbalance

After all other pre-processing, the data set contained 28517 entries without stroke and 548 entries with stroke, meaning only 1.92% of the data is associated with stroke. This is a massive imbalance in the data set which needed to be accounted for.

Two methods were considered: upsampling the stroke entries to the same count as the no stroke entries and downsampling the no stroke entries to the same count as the stroke entries. In either case, the imbalance will still prove to be a problem. Up sampling the data will cause the model to overfit to the training set and down sampling means throwing away about 92% of the data. However, for the purposes of generalisation, it is likely that downsampling would be the better of the two options. Both approaches were carried out and the results can be seen in the code.

Model Training

The model was fitted using the SVC class from scikit-learn. A train/test split of 70/30 was used to generate a labelled test batch for model evaluation. The data was then rescaled between 0 and 1 to ensure an equal importance of each feature.

Since the dataset had non-linear relationships between some of the features, SVC was chosen over linear SVC in order to improve the fit. However, SVC models cannot be interpreted by the ELI5 package which was used to explain model weights after fitting. Linear SVC models can though, so both estimators were used in parallel: the SVC model to get the best fit and linear SVC, which will result in a poorer accuracy metric but allow the debugging of model weights.

Predictions were then made using the test data split and evaluated against the actual test labels, yielding the following accuracy scores:

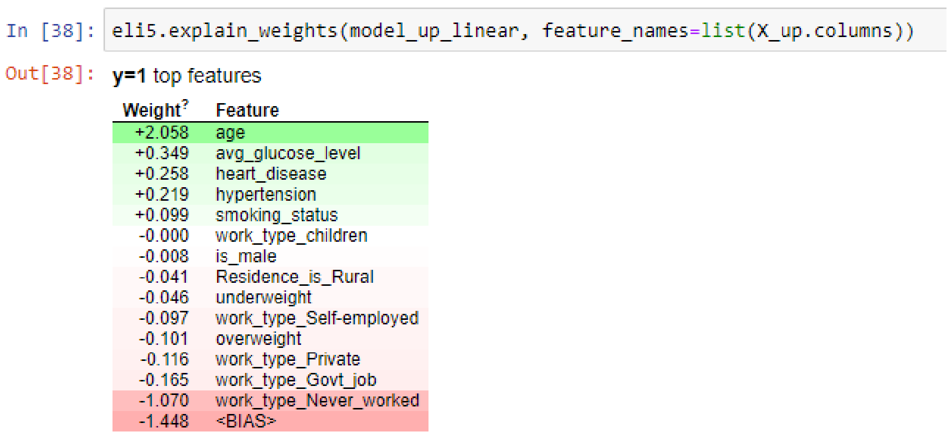
	SVC	**Linear SVC**
Upsampling	0.793	0.762

Downsampling	0.751	0.763

The model giving the best accuracy metric is the upsampled SVC model, however this is likely because the test data set is also upscaled (the train/test split occurs after upsampling) so the model evaluates on the same data points numerous times. In any case, all models seem to show good accuracy for unseen test data.

Interpreting the Model

ELI5 was used to reverse-engineer the model and see which features have the biggest effect. The results of this can be seen below, where positive (green) weights show the features contributing to a stroke labelling of 1:



```
In [38]: eli5.explain_weights(model_up_linear, feature_names=list(X_up.columns))
```

```
Out[38]:
```

y=1 top features

Weight?	Feature
+2.058	age
+0.349	avg_glucose_level
+0.258	heart_disease
+0.219	hypertension
+0.099	smoking_status
-0.000	work_type_children
-0.008	is_male
-0.041	Residence_is_Rural
-0.046	underweight
-0.097	work_type_Self-employed
-0.101	overweight
-0.116	work_type_Private
-0.165	work_type_Govt_job
-1.070	work_type_Never_worked
-1.448	<BIAS>

As is seen, *age* seems to be the highest contributing factor towards stroke. In fact, the output suggests that this is almost six times as important than the second highest contributor: *avg_glucose_level.* Other features such as *heart_disease, hyptertension* and *smoking_status* seem to be somewhat indicative of stroke, whereas features such as *work_type_children*, *residence* and *gender* seem to be less important. Interestingly, it seems that being overweight has a marginally positive effect on reducing stroke.

Many of the *work_type* categories seem to have an effect on reducing stroke, including *never_worked* which is the feature contributing the most to a negative stroke label. However, this is likely because of the inverse relationship that this feature has with *age*; people who are younger are more likely to have never worked, so there is bound to be correlation between these two features. In this case, correlation between stroke and never working does not seem to imply causation.

Refs:

Good DataCamp resource on the SVM pipeline: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

Used to help with the encoding of categorical data: https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd

A nice explanation of different ways of dealing with unbalanced data, although there is an error in the assessment of random trees: https://elitedatascience.com/imbalanced-classes