Data Science: Principles and Practice
Final Assignment A

Kamilė Stankevičiūtė (ks830)
Gonville & Caius College

2178 words

1 Data exploration

I examine the dataset of medical care records of diabetic patients over a period of 10 years (1999-2008) [1], as presented in a given sample *diabetic_data_balanced.csv*. This analysis has been done on a subset with at most one record per patient, as described later in the Preprocessing section.

1.1 Patient demographics

First I examine patient demographic data. The first part of Figure 1 shows the positively-skewed patient age distribution, where the majority of patients are from 50 to 90 years old.

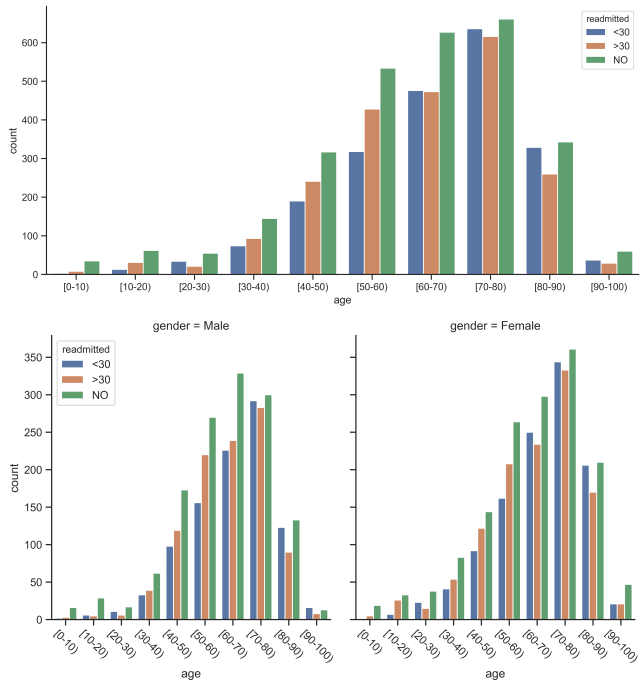

Figure 1: Patient age histograms categorised by outcome and gender.

Generally, more adult patients are readmitted than not, and we can observe that patients from the age of 60 have higher occurrence of '<30' label compared to '>30', indicating that at that age the patients are likely to be readmitted sooner. Further categorising by gender (bottom part of Figure 1), the readmission frequency increases earlier for female patients, where the relative occurrence of '<30' and '>30' changes at 60-70 rather than 70-80 years old.

1.2 Relationship between the type of treatment and readmission

For this particular dataset, distribution and normalised kernel density estimate plots seemed to represent most of the key trends of the full scatter plot matrix while being much easier to read (Figure 2). Here are some of the insights I gained from the data.

Most of the admissions to hospitals are related to emergency situations and serious medical problems. This can be seen from the peaks at admission_source_id=7 and admission_type_id=1 (where both identifiers correspond to emergency admission). The next highest source of admission is physician referral (admission_source_id=1), which may relate to urgent (but not emergency) and elective (planned) admissions (admission_type_ids 2 and 3). The relative distributions of readmission outcomes are similar across admission types (where the 'NO' case is the most common and the '<30' and '>30' outcomes are less common and closer in count), although there are some deviations from this. Similarity between distributions indicates that some particular identifier is uninformative for the readmission outcome, while the deviations are more important. For example, a single green peak at discharge_disposition_id=11 shows the patients who expired at that medical encounter – we can infer with certainty (given the current power of medicine to bring people back from the dead) that they are not going to be readmitted.

There is a proportion of patients with null and missing values (admission_source_id=17, discharge_disposition_id=18, admission_type_id=6). Their relative values (especially only '<30' patients having no discharge information) are generally different from the other distributions, which might indicate bias in how the records are filled.

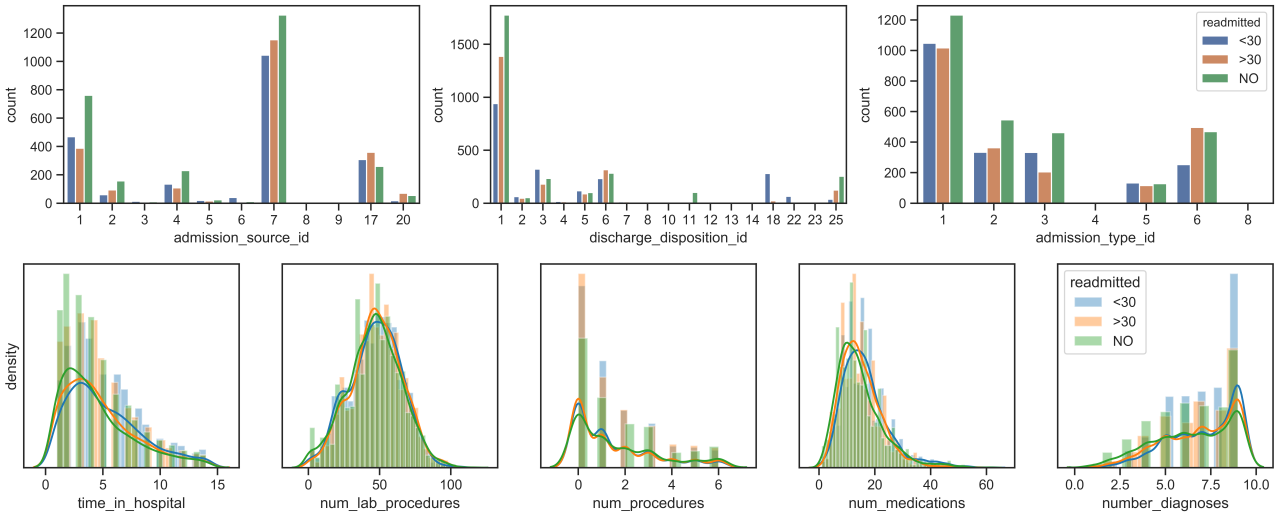

Figure 2: Histograms of admission and discharge identifiers; kernel density estimates for numerical features.

The length of stay (time_in_hospital) distribution is more skewed to the left for the patients who have no record of readmission, whereas the soon readmitted patients tend to stay for longer, and a somewhat similar trend can be seen in num_medications distribution.

With the increasing num_diagnoses, there patients are more likely to be readmitted within a month (the blue peak for '<30' being the highest at the maximum number of diagnoses, followed by the orange peak). On the other hand, patients with low number of diagnoses (e.g. 2-3) are more likely to have the 'NO' outcome (with the green curve higher than others).

Interestingly, patients with low number of procedures (num_procedures) are more likely to be readmitted (e.g. at num_procedures=0, '<30' and '>30' densities are higher), while the higher number of procedures has a higher 'NO' outcome density. This might be explained assuming the higher number of procedures gives the better estimate and therefore treatment of the patient, which helps to avoid later complications and readmission.

1.3 Relationship between diabetes conditions and readmission

This dataset focuses on the readmission outcomes of diabetic patients in particular.

Some set of interesting histograms is in Figure 3, where the patients are split by readmission outcome, the importance of diagnosis (primary or secondary), and the diagnosis type (which is based on the ICD numbers as described in Strack et al. [1]).

By definition of the dataset, *all* patients are supposed to have diabetes as one of their diagnoses; however, the relative number of diabetic patients in those distributions is low – only 46.2% of patients have diabetes listed as one of their first three diagnoses. This leaves the majority of admitted patients as having at least 4 diagnoses (which corresponds to the right-skewed distribution in the num_diagnoses panel of Figure 2) where diabetes is not among the first three. From the histograms, we can see that circulatory problems are very common – indeed, over 62% of the admitted patients had a circulatory problem as one of their first three diagnoses, and 58% of those had it listed as primary diagnosis. Another common set of conditions is related to respiratory issues, where 12% of all patients had it as their primary diagnosis. Given that most patients admitted had more serious primary health problems (and that many of those were probably an emergency), it might explain why only 18.2% and 9.2% of patients had their A1c and glucose serum tests taken respectively (with only 0.4% having both), even though all of the admitted patients had a diabetes condition.

2 Machine learning algorithms implementation

In the following section I implement the various machine learning algorithms covered in the course and report the mean training set cross-validation accuracy scores. I will further evaluate the algorithm that had the best mean accuracy in the Evaluation section.

2.1 Preprocessing and design choices

Anonymisation Some patients have multiple encounter records (up to 15 per patient). This might

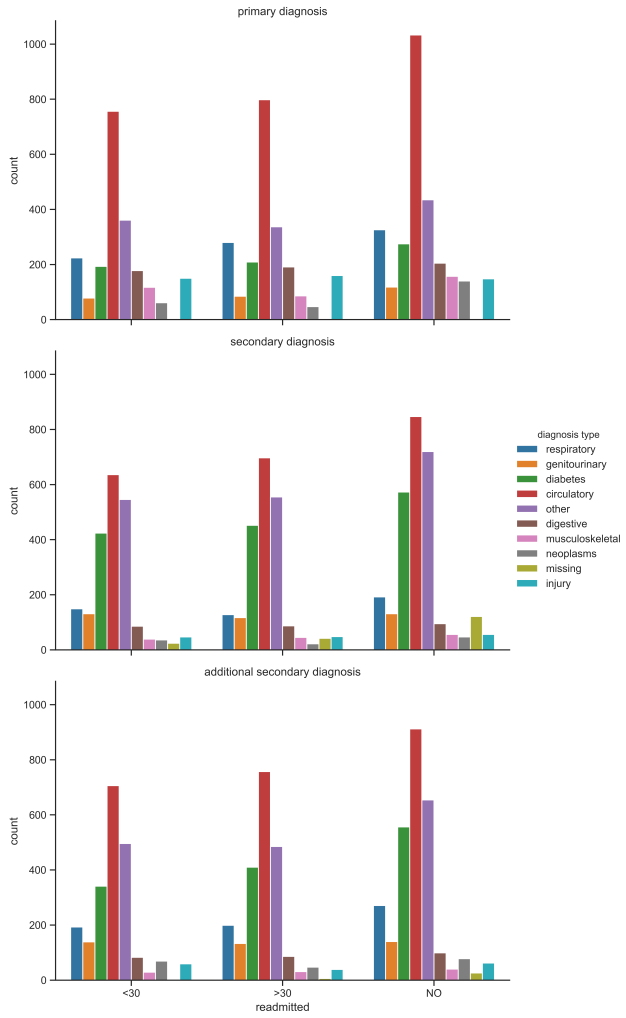

Figure 3: Histogram of readmission outcomes by diagnosis.

skew the results of further analysis (whether exploration or classification) as models might learn to identify particular patients (through patient number or otherwise). For this reason at most one randomly sampled encounter per patient will be included in further analysis, and patient and encounter numbers removed. This leaves 7944 unique instances.

Missing values The dataset has several features with missing values, most notably weight (97.0%), payer code (97.6%), and medical specialty (36.3%). The first two will be excluded from further analysis. I will assume that the values are missing at random and replace them (as well as the other missing categorical feature values) with a separate category. There seems to be no numerical feature values missing.

Unknown values It is possible that some categories present in the testing set are not present in training set, which may cause errors in one-hot-encoding pipeline. I set the parameters of one-hot encoder to ignore such errors, which returns zero- rather than one-hot vector, so that the number of features stays the same.

Numerical features Some features, such as admission source, type, and discharge identifiers should be categorical as those values should not have any ordering associated with them. I will convert those features to one-hot-encoded vectors. Other numerical features will be normalised.

Categorical features For most categorical features I will be using one-hot encoding. However, to represent the relative order of age categories, I will encode the age feature with consecutive integers and normalise them.

Feature sets I will analyse two feature sets: *full* feature set with 198 total features (after preprocessing), and *reduced* feature set with 93 features, the latter excluding medication and medical specialty data. I based this design choice on the assumption that most of the patients will not be taking all 24 drugs, so the features add little information most of the time while accounting for around half of the total features after preprocessing.

Train and test split I chose 90%/10% stratified train/test split with 7149 train and 795 test instances.

Hyperparameters I generally tried to use the default hyperparameters. For the best performing classifiers I then used grid search and similar techniques to further improve the accuracy. The hyperparameters used for each classifier can be found in `training.py` script.

2.2 Simple multi-class classifiers

Prediction of three readmission outcome labels is an instance of a multi-class classification task. I implement the multi-class *Naive Bayes* (NB), *stochastic gradient descent* (SGD) and *logistic regression* (LogReg) classifiers (using *one-vs-all* strategy where appropriate).

Results The 5-fold cross-validation on the training set gave the best results for the logistic regression classifier (mean accuracy 53.4%). All three classifiers performed better than the baseline mean accuracy (random guessing based on label distribution) of 33.3% (Table 1). The reduced feature set generally worked better, although not by much – these results, especially for logistic regression, are likely to be subject to noise.

	Baseline	NB	LogReg	SGD
full	33.3%	37.2%	53.6%	50.2%
red.	33.3%	38.7%	53.8%	52.8%

Table 1: Mean training set cross-validation accuracies for baseline, Naive Bayes, logistic regression and stochastic gradient descent classifiers, trained on full and reduced feature sets.

	Voting	Bag.	Past.	AB	GB
full	54.7%	52.9%	53.3%	54.1%	52.9%
red.	54.3%	53.7%	53.6%	53.9%	52.3%

Table 2: Mean training set cross-validation accuracies for voting, bagging, adaptive boosting and gradient boosting ensembles, trained on full and reduced feature sets. *Out-of-bag scores are reported for the bagging classifier.*

Grid search Further grid search on regularisation parameters and regularisation strength increased the mean cross-validation accuracy of logistic regression classifier to 53.9% for both full and reduced feature sets.

Kernel trick Transforming the reduced feature set using the `RBFSampler` with default parameters resulted in accuracies between 34% and 37%, therefore worse predictive power.

2.3 Ensemble models

For ensemble models, I have implemented the following:

(*i*) A *voting classifier* based on logistic regression, random forest and C-support vector classifiers;

(*ii*) The *bagging* and *pasting* techniques on a *decision tree* classifier (implemented as *random forest*);

(*iii*) The *adaptive boosting* (AB) technique on a *decision tree* classifier;

(*iv*) The *gradient boosting* (GB) technique on a decision tree classifier.

Results The results are presented in Table 2, with out-of-bag scores for the bagging classifier (since these give a better test dataset performance estimate). Generally results are similar between ensemble techniques and feature sets since the differences in accuracies are small and vary depending on the technique chosen. The highest result in this comparison was for the voting classifier which gave the mean cross-validation accuracy of 54.7% for the full feature set.

Tuning Further hyperparameter search using automated machine learning tools [2] boosted the training set cross-validation accuracy of the **gradient boosting classifier** to 54.8% on the full feature set and **55.4% on the reduced feature set**. Since this is the best cross-validation accuracy among all classifiers I will use this model for further evaluation.

3 Further evaluation of the best performing classifier

In this section I report the test set performance of the best classifier.

Test accuracy The final test set accuracy of the best performing classifier was **55.5%**.

Precision, recall and F_1 score The overall precision and recall, combined using *macro-averaging* (which treats all classes equally by averaging the metrics computed for each class), are 0.571 and 0.525 respectively. If we average F_1 scores in a similar manner (rather than combining the previous two scores directly), the overall multi-class F_1 score is 0.513.

Receiver operating characteristic The ROC curves can be seen in Figure 4. The individual AUC scores are 0.784, 0.636, and 0.758 for '<30', '>30' and 'NO' cases respectively. The macro-averaging strategy gives the combined AUC score of 0.726.

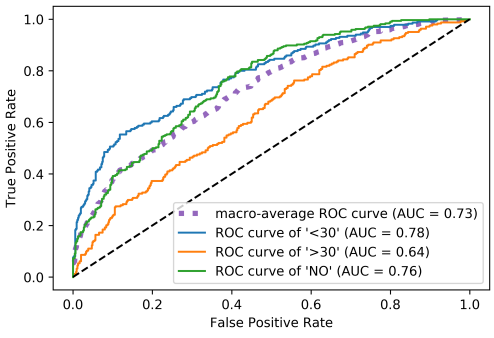

Figure 4: ROC curves for the best classifier.

Interpretation The confusion matrix is shown in Figure 6. The instances where the patient was not readmitted were distinguished the best, followed by the instances where the patient was readmitted the soonest ('<30'). The classifier was the worst at distinguishing the '>30' readmission outcome (corresponding to the worst ROC curve and lowest AUC for that class). This makes sense intuitively – if the time until readmission

spans many years, the initial encounter record may not contain enough information to foresee this and therefore the patient is misclassified as healthy (with no known readmissions).

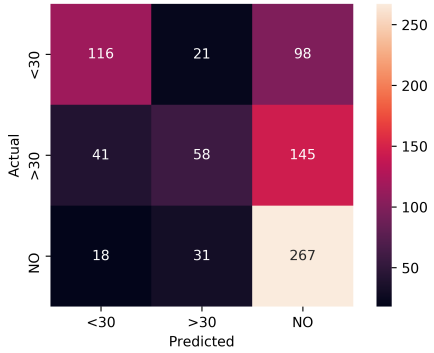

Figure 6: Confusion matrix for the best classifier.

4 Dimensionality reduction and embeddings

The following dimensionality reduction techniques work on numerical features only. For further analysis I will consider features that have clear numerical meaning (i.e. I will exclude features which are encoded as numbers but are categories, and will encode ordinal features, such as age, as numbers).

4.1 Principal component analysis

The projection of the first two PCA components is displayed in Figure 7. The patients who are readmitted are more spread out along the second component range while the 'NO' patients are displayed closer together at the bottom of the figure where the second component is around 0, which suggests the second component uses features that are helpful in distinguishing the 'NO' outcome. Checking the `components_` attribute of `sklearn`'s `PCA` object, the second component mostly uses the number of inpatient, outpatient and emergency visit data. Indeed, Figure 8 shows some distinct peaks for every readmission outcome across the three plots, which I surprisingly overlooked in the initial data exploration section!

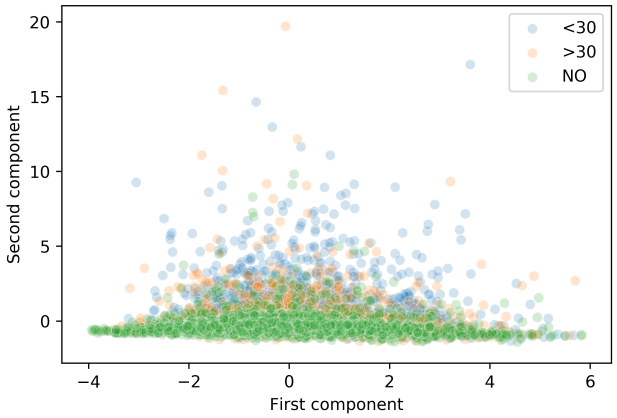

Figure 7: Principal components analysis of the diabetic patient dataset.

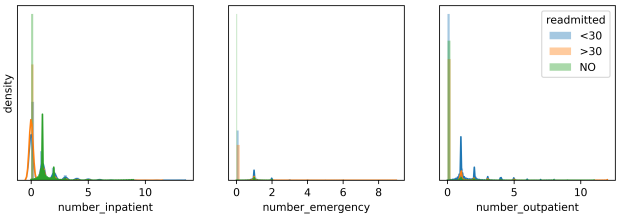

Figure 8: Distribution plots for inpatient, outpatient and emergency visits.

4.2 *t*-distributed stochastic neighbour embedding

The t-SNE embeddings are shown in Figure 5. Those embeddings do not separate the readmission outcomes well, but at all perplexities four main clusters can be seen (one containing the majority of the values, a "triangular" cluster and two smaller well-separated clusters).

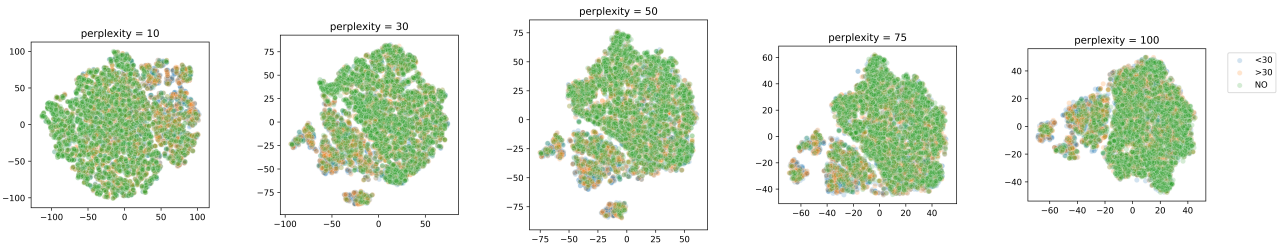

Figure 5: t-SNE embeddings of the diabetic patient dataset for selected perplexities.

References

[1] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014.

[2] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 485–492, New York, NY, USA, 2016. ACM.