Notes on Exploratory Data Analysis and Data Preprocessing

Sameer Kesava PhD

Contents

Exploratory Data Analysis

- Unsupervised learning must be carried out as part of EDA.
- How the features are represented, i.e. feature engineering, makes a huge difference in the performance of ML algorithms.

Correlations between Features

2.1 Variance Inflation Factor

- Implemented in StatsModel.
- Rule of Thumb: if VIF>5 or 10, this indicates significant collinearity between features.

2.2 Correlation Matrix

Pearson correlation coefficient

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{2.1}$$

Scaling

Scaling is not needed in classifiers using Decision trees such as RandomForest.

Dimensionality Reduction

Two main categories: 1). feature selection 2). feature extraction

4.1 ML Algorithms

The algorithms for reducing the number of features in the input data.

4.1.1 Random Forest

- Feature importances can be used to select features without making an assumption if the features are linearly separable or not.

4.1.2 Principal Component Analysis

Popular applications of PCA:

1. EDA
2. Denoising of signals, e.g. stock market data, genome data and gene expression levels.

The features should be standardized to a σ=1 and, preferably, μ=0 before using PCA.

Kernels:

1. Linear
2. Radial (RBF): for non-linear datasets.
    - Gaussian kernel function: γ is the hyperparameter.

PCA tends to perform well in cases where most of the information is captured by the first few principal components. This can be obtained as variance explained ratios implemented in sklearn as pca.explained_variance_ratio_, and plotted as a **Scree plot** (elbow points in this plot gives an estimate for the choice of no. of PCs to choose for further processing).

Biplots: for visualizing PC scores and loading vectors.

4.1.3 Linear Discriminant Analysis

LDA classifier assumes that the observations are drawn from classes with a normal distribution of predictors while sharing a common variance σ^2.

4.1.4 Quadratic Discriminant Analysis

QDA classifier assumes that the observations are drawn from classes with a normal distribution of predictors but not sharing a common variance σ^2.

4.1.5 UMAP

- Feature extraction method
- Non-linear dimensionality reduction
- Other such methods are **triMAP**.
- t-SNE is only for visualization and not for data preprocessing.

4.1.6 Dendogram

- For visualization of high-dimensional data.

4.2 Small Dataset

Also see Augmenting Data section 5.1.
n: number of samples
p: number of features/predictors
Use Leave-One-Out-CV (LOOCV) for ***very small datasets***.

4.2.1 When p is comparable to n

1. Use shrinkage methods such as Lasso or L1 norm.

4.2.2 $\mathbf{p \ll n}$

1. Use k-fold CV. And Stratified k-fold to preserve class proportions.

4.3 Large Dataset

4.3.1 Out-of-core learning

- For handling large datasets which cannot be accommodated in the memory.
- Partial fit function using stochastic gradient descent is used here.

4.3.2 Caching

- Performance could be improved with a sophisticated caching strategy, e.g. by sharding the images to reduce random access disk I/O.
- tqdm module for caching.

Data Transformation

The various data transformation methods available for preprocessing and for addressing non-linear decision boundaries.

5.1 Augmenting Data

Accuracy can be improved by augmenting to the data and then fitting the model with both the original and augmented datasets. Some examples are

- Adding random noise to the data.
- In case of images, random rotations of the images.

5.2 Embedding

Embedding of characters and words in character/language modeling is a popular way of encoding in DL.

- Efficient, dense represention
- A hyperparameter - 8 for small datasets to 1024 for large.
- A high dimensional embedding can capture fine-grained relationships between words but takes more data to learn.
- A simple model could yield more interpretable embeddings.

5.3 Bag-of-Words

To convert text to numerics.

1. n-gram model (Tokenizer in Keras)
2. CountVectorizer in sklearn
3. In case of large datasets, Hashing Vectorizer can be used. See Out-of-core learning subsection 4.3.1.

5.3.1 NLP

- When handling multiple documents, a different parameter accounting for the frequency of occurrences is ***term-frequency - inverse document frequency***.
- In **NLP**, punctuation marks and emoticons could reveal useful information, and hence, should not be discarded.
- Use of ***word stemming*** and ***lemmatization*** helps obtain the grammatically correct form of the word.
- Another important step is ***stop-word removal***, i.e. removal of words such as 'and', 'is', 'has' and 'like' which contain little or no useful information that can be used to distinguish between different classes of documents.
- Start and End tokens added to each sentence to handle variable input sizes before training with Encoder-Decoder (see Forecasting document).

5.4 Log Transformation

This transformation could result in the distribution appearing bell-like or normal.

5.5 Differencing

In the case of time-series forecasting, differencing to remove a trend or seasonality should be carried out, after which, scaling and transformations should be implemented.

5.6 Power Transformation

Definition (from wiki) : In statistics, a power transform is a family of functions that are applied to create a monotonic transformation of data using power functions. This is a useful data transformation technique used to stabilize variance, make the data more normal distribution-like, improve the validity of measures of association such as the Pearson correlation between variables and for other data stabilization procedures. One example is Box-Cox transformation.

Applications (from wiki): Power transforms are ubiquitously used in various fields. For example, multi-resolution and wavelet analysis, statistical data analysis, medical research, modeling of physical processes, geochemical data analysis, epidemiology and many other clinical, environmental and social research areas.

5.7 Using DL

Neural networks can be used for feature extraction which then can be used as inputs to the traditional ML algorithms, e.g. used in statistical machine translation system.

Curse of Dimensionality

The ML algorithms significantly affected by high dimensions.

1. KNN
2. KMeans
3. DBSCAN
4. SVM