Table of Contents

Describe the Data: 2

Data Summarization 2

Variable/Feature Identification 4

Data Preprocessing 4

Data Transformation 4

- Skew Handling: 4
- Scaling 5
- Descritization 5
- Binarize Data 5
- Encoding 5
- Decomposition 5
- Aggregation 5
- Binning 5
- Dimensionality Reduction 5
- Transforming the Label 6

Missing Value Imputation 6

Preprocessing in Scikit Learn: 6

Handling of Imbalanced Datasets 6

Outlier Handling 7

Basics of Feature Selection and Importance 7

- Feature Ranking by LVQ (Learning Vector Quantization) 7
- Recursive feature elimination 7
- Univariate feature selection methods in ScikitLearn 7

Feature Extraction: 8

Software Tools for EDA and Data Prep 8

1. OPTIMUS 8
2. imbalanced-learn 8
3. PETL 8
4. Scikit-Feature 8

5. Feature Miner 9
6. PyViz: http://pyviz.org/ 9
7. https://datavizproject.com/ 9

Following are the Key Steps involved in Data preparation:

Describe the Data:

- Start out by identifying the "Predictors"(Inputs) and Labels(Outputs) in the dataset.

Data Quality and Validation

The degree to which the measures conform to defined business rules or constraints:

- *Data-Type Constraints* – e.g., values in a particular column must be of a particular datatype, e.g., Boolean, numeric (integer or real), date, etc.
- *Range Constraints:* typically, numbers or dates should fall within a certain range. That is, they have minimum and/or maximum permissible values.
- *Mandatory Constraints:* Certain columns cannot be empty.
- *Unique Constraints:* A field, or a combination of fields, must be unique across a dataset. For example, no two persons can have the same social security number.
- *Set-Membership constraints*: The values for a column come from a set of discrete values or codes. For example, a person's gender may be Female, Male or Unknown (not recorded).
- Regular expression patterns: Occasionally, text fields will have to be validated this way. For example, phone numbers may be required to have the pattern (999) 999-9999.
- Cross-field validation: Certain conditions that utilize multiple fields must hold. For example In a hospital database, a patient's date of discharge from hospital cannot be earlier than the date of admission.

Data Summarization

- **Review various slices of the data**

Check data.head, data.tail and other records from the mid of the dataset. Check 5, 10, 100 records from each slice.

- **Get the dimensions of your Data**
    - In Pandas: Data.shape ==> Prints number of Rows and Columns in data
    - Check if record count is too large which will result in algos taking too long to train. Similarly if too few records than not enough data to train.
    - Check if too many features than may need to perform dimensionality reduction. Too many features and some algorithms can be distracted or suffer poor performance due to the curse of dimensionality. A good way to check is if n>>m where n is the number of

features and m is the number of records. So in essense number of records must be significantly "large" compared to the number of features..

- **Get data Distributions**
    - for classification problems using data.groupby("var").size()
        - This will return the size of each class that exists within each class of the Label variable. It will highlight if imbalance exists as well as define the max possible accuracy of a model
    - Using data.describe() will return following stats:
        - Count
        - Mean
        - Standard Devaition
        - Minimum Value
        - 25th Percentile
        - 50th Percentile (Median)
        - 75th Percentile
        - Maximum Value
        - # missing values.
    - Get measures of dispersion in data such as
        - Range
        - IQR
        -
- Generate Box Plots for each variable
- Generate a summary of the pair-wise attribute correlations using a parametric (Pearson's) and non-parametric (Spearman's) correlation coefficient. This can highlight attributes that might be candidates for removal (highly correlated with each other) and others that may be highly predictive (highly correlated with the outcome attribute).
- Create univariate plots of each attribute.
- Create bivariate plots of each attribute with every other attribute or Feature Feature Relationship.
- Create bivariate plots of each attribute with the class variable or Feature Class Relationship.
- **Get Feature Correlations** which refers to relationship between features and has impact on model accuracy score. Use "Pearson" correlation coefficient to calculate correlation between features: data.corr(method = 'pearson'). Also generate correlation matrices.
    - **Correlation = Covariance(X,Y) / SQRT(Var(X)* Var(Y))**
- **Generate Attribute Histograms and following charts:**
    - Five number summaries (mean/median, min, max, q1, q3)
    - Histogram graphs
    - Line Charts
    - Box and Whisker plots
    - Pairwise Scatterplots (scatterplot matrices)
    - Typical graphical techniques used in EDA are:
    - Box plot
    - Histogram
    - Multi-vari chart
    - Run chart
    - Pareto chart

        - Scatter plot
        - Stem-and-leaf plot
        - Parallel coordinates
- **Generate Relationship Plots**
        - Stacked Column Charts
- Perform chi sqr test which signifies the relationship between two variables. also infers whether the results of relationship may be generalized to the global population.
- For Categorical & Continuous variables Perform
        - Z-test/T test: Checks whether the mean of the two variables is statistically different form each other.
        - ANOVA: The same as z test but applied for more than 2 groups.

Variable/Feature Identification

Identify Feature types that exist within the data. In Pandas: df.dtypes

- **Numerical**
    - Continuous
    - Discrete
    - Ordinal
- **Categorical**
    - Nominal
    - Dichotomous
- Nominal
- Binary
- Boolean
- Text
- Time
- Ordinal

Data Preprocessing

This step Involves following Key aspects:

1. **Data Transformation**
2. **Data Cleaning**
3. **Data Sampling**

Data Transformation

The objective of this section is to expose hidden structure of the dataset which is otherwise invisible in the raw data.

Below is a list of some univariate (single attribute) transforms that may be used

- **Skew Handling:** Skewed data is data that has a distribution that is pushed to one side or the other (larger or smaller values) rather than being normally distributed. Some methods assume

normally distributed data and can perform better if the skew is removed. Try replacing the attribute with the log, square root or inverse of the values. Skew can be checked using **data.skew()** in pandas.

- Square and Cube
- Square root
- Box-Cox: A Box-Cox transform or family of transforms can be used to reliably adjust data to remove skew.
- Logirthm

- **Scaling_Rescaling**
  - Standardize (e.g. 0 mean and unit variance)
  - Normalize (e.g. rescale to 0-1).
    - Normalization may be performed on the basis of:
    - Zscore, Logistic normalization, Log Normal Normalization, TanH Normalization
    - Note that some ML algos have a pre requisite for Normalization
  - Rescaling such as Min-Max Normalization, Mean Normalization, Scale to Unit Length,
- **Descritization** (e.g. convert a real to categorical)
- **Binarize Data** (Make Binary)

- **Encoding**
  - Integer Encoding
  - One Hot Encoding
- **Decomposition:** There may be features that represent a complex concept that may be more useful to a machine learning method when split into the constituent parts. An example is a date that may have day and time components that in turn could be split out further. Perhaps only the hour of day is relevant to the problem being solved. consider what feature decompositions you can perform.
  - **Decompose** Categorical Attributes
  - **Decompose a Date-Time**

- **Aggregation**: There may be features that can be aggregated into a single feature that would be more meaningful to the problem you are trying to solve. For example, there may be a data instances for each time a customer logged into a system that could be aggregated into a count for the number of logins allowing the additional instances to be discarded. Consider what type of feature aggregations could perform.
- **Binning**: Numeric data can be made discrete by grouping values into bins. This is typically called data Descritization. Binning can be performed using Quantiles, Percentiles or using categorization of the numerical features.

- **Dimensionality Reduction**

  Unsupervised dimensionality reduction
  - PCA: principal component analysis
  - Random projections
  - Feature agglomeration
- **Random Projection**
  - The Johnson-Lindenstrauss lemma
  - Gaussian random projection
  - Sparse random projection

- ### Transforming the Label

  Label binarization

  In scikit learn use **LabelBinarizer** to help create a label indicator matrix from a list of multi-class labels

  Label encoding

  Use **LabelEncoder** to help normalize labels such that they contain only values between 0 and n_classes-1.

Missing Value Imputation

Following techniques are frequently used:

- **Record Deletion**: List wise or Pair Wise Deletion.
  - Deletion can only be used if missed records are random in nature and iid else bias will be introduced in the data.
- Mean based Imputation
- KNN imputation

Preprocessing in Scikit Learn:

Check out the "Preprocessing class" in scikit learn for functions related to preprocessing**.**

- 4.3.1. Standardization, or mean removal and variance scaling
- 4.3.1.1. Scaling features to a range
- 4.3.1.2. Scaling sparse data
- 4.3.1.3. Scaling data with outliers
- 4.3.1.4. Centering kernel matrices
- 4.3.2. Non-linear transformation
- 4.3.2.1. Mapping to a Uniform distribution
- 4.3.2.2. Mapping to a Gaussian distribution
- 4.3.3. Normalization
- 4.3.4. Encoding categorical features
- 4.3.5. Discretization
- 4.3.5.1. K-bins discretization
- 4.3.5.2. Feature binarization
- 4.3.6. Imputation of missing values
- 4.3.7. Generating polynomial features
- 4.3.8. Custom transformers

Handling of Imbalanced Datasets

Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally.

Following methods may be used to balance out the dataset:

- Try to collect more data

- Perform Data re-sampling or re-balancing. This entails adding or deleting copies or some samples of under/over represented classes respectively. Use both interchangeably.
    - Under sampling.. When a lot of data is available
    - Oversampling .. When data availability is sparse..
    - Perform sampling using Stratified/Random/Simple schemes.
- Generate Synthetic samples from under represented class using SMOTE . In Python, SMOTE implementation is available in " UnbalancedDataset" module.
    - Random forest with SMOT boosting
    - cost sensitive training on SVM.
- Try creating a test harness with atleast a couple dozen ML models not just a single algo.
- May be problem is better suited for anomaly detection algos etc.
- Use metrics other than accuracy such as AUC/ROC, FI, Kappa etc.
- Besides SMOTE ADASYN can be used for Oversampling
- Cluster Centroids and "Tomek Links" are techniques for Undersampling.

Outlier Handling

Extreme or Outlier values may be handled using following techniques:

- Extreme Value Analysis. Z scores may be used on uni variate data. Scatterplots, Box plots and Histograms reveal generic outlier values.
- Linear models such as PCA, SOM and ICA.
- Density based models such as clustering and KNNs
- LOF or Local outlier factor
- simple statistical tests such as classifying values as outliers if value 2* SD

Basics of Feature Selection and Importance

For Tree based algos, feature importance is provided out of the box by the algo itself. For other algos, following approaches may be used.

- **Feature Ranking by LVQ (Learning Vector Quantization)**
- **Recursive feature elimination**

Caret (R) package can be used to perform both the above steps conveniently.

- **Univariate feature selection methods in ScikitLearn**
    - SelectKBest
    - SelectPercentile
    - selectFPR
    - selectFDR
    - selectFWE
    - GenericUnivariateSelect
    - If data is Sparse(features are sparse) then use Chi^2, mutual_info_regression, mutual_info_classif
- Fisher Linear Discriminant Analysis

- Other methods
    - Pearson's correlation score
    - Mutual information score
    - Kendall's correlation
    - Remove features with low Variance using "Variancethreshold" in scikiL. It will remove all features that have very similar values in all samples/records.
    - Perform feature selection using automated tools such as:
        - Featureselection: http://featureselection.asu.edu/
        - **Featureminer:** http://featureselection.asu.edu/featureminer.php

Feature Extraction:

Reducing the number of features by creating lower-dimension, more powerful data representations using techniques such as PCA, ICA, hashing etc.

Software Tools for EDA and Data Prep

1. ## OPTIMUS

Optimus is a tool to prepare messy data using Python and Spark.

2. ## imbalanced-learn

https://github.com/scikit-learn-contrib/imbalanced-learn

Imbalance helps in dealing with data with class balance issues. It offer nearly 40 algorithms for working with imbalanced data.

http://imbalanced-learn.org

3. ## PETL

petl - Extract, Transform and Load (Tables of Data)

petl is a general purpose Python package for extracting, transforming and loading tables of data.

Pasted from <https://petl.readthedocs.io/en/latest/>

4. ## Scikit-Feature

http://featureselection.asu.edu/index.php

https://github.com/jundongl/scikit-feature

5. ## Feature Miner

http://featureselection.asu.edu/featureminer.php

6. ## PyViz: http://pyviz.org/

7. ## https://datavizproject.com/

Use as inspiration for plotting, visualization and to choose chart types.