Data Cleaning Wheel

Load Data

load dataframe

function loaddata(dataset,labellist,IDlist)

> ***dataset***: str, the path of data file
>
> ***labellist***: list, contain all the target label
>
> ***IDlist***: list, contain all the ID feature which are useless

return quantitative,qualitative,df_train,targetdf_list

Missing Value detect and process

missing value percentage plot

function **Missing_Table()**

return a tabel contains missing ratio

---

show missing value table

function **Missing_Plot()**

return a plot about missing features and missing ratio

---

drop and fill missing value

function **Missing_Process**(drop_threshold=0.90,filllist=[],fillnum='mean',fillcat='Missing')

> ***drop_threshold***: 0~1,int, default is 0.9, missing ratio over the threshold will be dropped, if you don't want drop any features, set it to 1
>
> ***filllist***: list, contains the features with missing value you want to fill, default is [], if set it to [], the program will fill all the features
>
> ***fillnum***: str,the method you use to fill numerical features, default is mean, option:mean,median,mode,0,-999 or usedefined integer value
>
> ***fillcat***: str,the method you use to fill categorial features, default is Missing, option:Missing, None, or

usedefined string

return a list of which feature to drop

Outlier detect and process

outlier plot

function **Outlier_Plot**(feature,label,k=4,show_grid=True,plot_outlier=True)

feature: str,the feature you choose, option:all, if you select all,will plot all features with target label

label: str, the target label

k: int, a positive integer, the larger the number is, the lower the detect standard is, default is 4

show_grid: boolean, if show grid,default is True

Plot_outlier: boolean, if mark outliers, default is True

return a 2Dplot of single feature with target label, which shows outliers as red points

outlier collect

function **Outlier_collect**(feature,label,k=4)

feature: str,the feature you choose, option:all, if you select all,will plot all features with target label

label: str, the target label

k: int, a positive integer, the larger the number is, the lower the detect standard is, default is 4

return the outliers position

outlier drop

function **Outlier_Drop**(feature,label,k=4,method='',droplist=[])

feature: str,the feature you choose, option:all, if you select all,will plot all features with target label

label: str, the target label

k: int, a positive integer, the larger the number is, the lower the detect standard is, default is 4

method: str, control the method to drop, default '', option:dropall,'',dropsome

droplist list, control which outliers to drop, only be used when method=dropsome, default:[]

return the dataset after process

single value and duplicated value process

single value process

function **single_process**(drop_threshold=0.99,single_ratio=0.95,collect=True,show_plot=False,show_table=False,drop=False)

drop_threshold: float,drop the feature over this value, default:0.99

single_ratio:float,if a feature over this value, the feature will be collected and plotted

collect: boolean, if collect the single value, default True

show_plot: boolean, if show the single value plot, default False

show_table: boolean, if show the single value table, default False

drop: boolean, if drop the feature over drop_threshold, default False

return the dataset after process

duplicated value process

function **duplicate_process**():

drop the duplicated row, return update dataset

Univariable Visulization Analytic

function **Distplot**(feature,fitmethod='norm',show_QQplot=True,show_skew=False,show_kurt=False,color='m',kde=True,rug=False)

feature: str, select a feature

fitmethod: str, {norm | lognorm | johnsonsu} option

show_QQplot: Boolean, default True, if True, generate a QQ plot

show_skew: Boolean, default False, if True, generate a skewness distplot of all features

generate a skewness distplot of all features

show_kurt: Boolean, default False, if True, generate a kurtosis distplot of all features

color : str, color

kde: Boolean, default True

rug: Boolean, default False

return figures for analytic

---

function **Describe**(feature,show_plot=True):

feature: str, select a feature

show_plot: Boolean, show describe bar plot

return figures for analytic

---

function **Countplot**(feature,show_plot=True)

feature: str, select a feature

show_plot: Boolean, show count bar plot, default True

return figures for analytic

---

function **Kurtosis**(show_plot=True)

show_plot: Boolean, show kurtosis plot and table, default True

return figures for analytic

---

function **Skewness**(show_plot=True)

show_plot: Boolean, show skewness plot and table, default True

return figures for analytic

Multivariable Visulization Analytic

function **Scatterplot**(xlabel,ylabel,reg=True,color=None,logx=False,robust=False)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

reg: Boolean, if show regression fit line, default True

color: str,default None

logx: Boolean, default False

robust: Boolean, default False

return figures for analytic

---

function **Jointplot**(xlabel,ylabel,kind='reg',color='b',add_scatter=False)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

kind: default reg{ scatter | reg | resid | kde | hex } optional

color: default None

add_scatter: Boolean, if True, add scatter to the plot

return figures for analytic

---

function **Pairplot**(xlabel,ylabel,columns,kind='scatter',diag_kind='kde',size=3)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

columns: list, a list contains all the faetures you want to plot

kind: str, {scatter, reg} optional, default scatter

diag_kind: str, {hist, kde} optional, default kde

size: integer, the size of figures, default 3

return figures for analytic

---

function **Correlation**(xlabel,ylabel,zoom=0.2,show_cm=True,show_heatmap=True)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

zoom: float, zoom the correlation feature through zoom ratio, default 0.2, avalible when show_cm=True

show_cm: Boolean, default True, show correlation with the label

show_heatmap: Boolean, default True, show the correlation heatmap

return figures for analytic

---

function **Barplot**(xlabel,ylabel,hue)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

hue: str, the third value to be plotted, default None

return figures for analytic

---

function **Boxplot**(xlabel,ylabel,hue)

xlabel: str, the feature in x axis

ylabel: str, the feature in y axis

hue: str, the third value to be plotted, default None

return figures for analytic

---

function **Violinplot**(xlabel,ylabel,hue)

> ***xlabel***: str, the feature in x axis
>
> ***ylabel***: str, the feature in y axis
>
> ***hue***: str, the third value to be plotted, default None

return figures for analytic

---

function **Pointplot**(xlabel,ylabel,hue)

> ***xlabel***: str, the feature in x axis
>
> ***ylabel***: str, the feature in y axis
>
> ***hue***: str, the third value to be plotted, default None

return figures for analytic

---

function **Stripplot**(xlabel,ylabel,hue)

> ***xlabel***: str, the feature in x axis
>
> ***ylabel***: str, the feature in y axis
>
> ***hue***: str, the third value to be plotted, default None

return figures for analytic

---

function **Swarmplot**(xlabel,ylabel,hue)

> ***xlabel***: str, the feature in x axis
>
> ***ylabel***: str, the feature in y axis
>
> ***hue***: str, the third value to be plotted, default None

return figures for analytic

---

function **Factorplot**(xlabel,ylabel,hue,col,color=None,kind='point')

> ***xlabel***: str, the feature in x axis
>
> ***ylabel***: str, the feature in y axis
>
> ***hue***: str, the third value to be plotted, default None
>
> ***col***: str, the fourth value to be plotted, default None
>
> ***color***: str, default None
>
> ***kind***: str, including point plots,box plots, violin plots, bar plots, or strip plots, default point