Cheatography

Python - Logistic Regression Model Cheat Sheet

by DarioPittera (aggialavura) via cheatography.com/83764/cs/19936/

TO START

```
# IMPORT DATA LIBRARIES
import pandas as pd
import numpy as np
# IMPORT VIS LIBRARIES
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
# IMPORT MODELLING LIBRARIES
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

PRELIMINARY OPERATIONS

df.pd.read_csv('data.csv')	read da
df.head()	check head df
df.info()	check info df
df.describe()	check stats df
df.columns	check col names

VISUALISE DATA

sns.heatmap(df.isnull())*	check null values
sns.set_style('whitegrid')	set different style
sns.countplot('col',df)	countplot
sns.countplot('col',df,palette='')	countplot
sns.countplot('col',df,hue='',palette='')	countplot
sns.distplot(df['col'].dropna(),bins=30)	distribution plot

sns.heatmap(): can take more useful parameters; yticklabels=False,cbar=False,cmap='viridis'

DATA CLEANING

create a personalise function*	impute values
apply the personalised function*	apply function
dummy_var = **pd.get_dummies(df['col'],drop_first=True)***	convert categorical features
df.drop(['old.col1',...''])	drop old columns
df= pd.concat([dummy_var],axis=1)	add the new dummy var into the df

See imputing and apply section.

drop.first=True: without this command, we would have two specular columns, leading to issues of multicollinearity.

IMPUTING AND APPLY

```
# EXAMPLE OF A POSSIBLE FUNCTION TO IMPUTE MISSING VALUES
def impute_age(cols):
     Age = cols[0]
     Pclass = cols[1]

     if pd.isnull(Age):
          if Pclass == 1:
             return 37
          elif Pclass == 2:
             return 29
          else:
             return 24
     else:
         return Age
# EXAMPLE OF HOW TO APPLY THE FUNCTION
train['Age'] = rain[['Age','Pclass']].apply(impute_age,axis=1)
```

You can impute using mean, median, etc. If you are interested in using Bayesian Estimation, you can see here: https://github.com/jeweinberg/Pandas-MICE or https://pypi.python.org/pypi/fancyimpute

TRAIN and EVALUATE MODEL

CREATE X and y

X = df[['col1','col2',etc.]]	create df features
y = df['col']	create df var to predict

SPLIT DATASET

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)	split df in train and test df

FIT THE MODEL

log = LogisticRegression()	instatiate model
log.fit(X_train,y_train)	train/fit the model

MAKE PREDICTIONS

predictions = log.predict(X_test)	make predictions

EVAUATE MODEL

print(classification_report(y_test,predictions))	useful measures
confusion_matrix(y_test, predictions)	