....

Data Analysis with Python

Cheat Sheet: Model Development

Crieat Street, Moder Development			
Process	Description	Code Example	
Linear Regression	Create a Linear Regression model object	1 from sklearn.linear.model import LinearRegression 2 le - LinearRegression()	@
Train Linear Regression model	Train the Linear Regression model on decided data, separating input and Output attributes. When there is single attribute in input, then it is simple linear regression. When there are multiple attributes, it is multiple linear regression.	1 X = df[['attribute_2', 'attribute_2',]] 2 Y = df['target_attribute'] 3 lr.fit(X,Y)	@
Generate output predictions	Predict the output for a set of input attribute values.	1 Y_hat - 1r.predict(X)	@
Identify the coefficient and intercept	Identify the slope coefficient and intercept values of the linear regression model defined by $\hat{y}=mx+c$ Where m is the slope coefficient and c is the intercept.	1 coff = ln.coef 2 intercept = ln.intercept_	@
Residual Plot	This function will regress y on x (possibly as a robust or polynomial regression) and then draw a scatterplot of the residuals.	<pre>import seabon as sns sns.residplot(r=eff[[stribute_2']], y=eff[('attribute_2']])</pre>	(9)
Distribution Plot	This function can be used to plot the distribution of data w.r.t. a given attribute.	1 import seaborn as sns 2 sns.distplot(df['attribute_name'], hist-false) 3 # can include other parameters like color, label and so on.	@
Polynomial Regression	Available under the numpy package, for single variable feature creation and model fitting.	f = op.polyfit(x, y, n) Kereates the polymonial features of order n p = op.polyfit(f) B becomes the polymonial model used to generate the predicted output Y_Nat = of(x) V_Nat = of(x) V_Nat is the predicted output	8
Multi-variate Polynomial Regression	Generate a new feature matrix consisting of all polynomial combinations of the features with the degree less than or equal to the specified degree.	1 from sklearm.preprocessing import PolynomialFeatures 2 Z = df[[stributez V.', strribute 2.',]] 3 pr.=NaynomialFeatures(Segreen) 4 Z_pro-pr.fit_transform(2)	@
Pipeline	Data Pipelines simplify the steps of processing the data. We create the pipeline by creating a list of tuples including the name of the model or estimator and its corresponding constructor.	1 from sklearn.pipeline import Pipeline 2 from sklearn.preprocessing import Standardscaler 3 Impute("scale", Standardscaler()), ("obj.(scale",)), ("obj.(sca	@
R*2 value	RP2, also known as the coefficient of determination, is a measure to indicate how close the data is to the filted regression line. The value of the R-squared is the percentage of variation of the response variable (y) that is explained by a linear model. a. For Linear Repression (single or multi attribute) b. For Polynomial regression (single or multi attribute)	1	@
MSE value	The Mean Squared Error measures the average of the squares of errors, that is, the difference between actual value and the estimated value.	1 from sklearn.metrics import mean_squared_error 2 nse = mean_squared_error(Y, Yhat)	@



