Hands-On Machine Learning Cheat Sheet

by Remidy08 via cheatography.com/159206/cs/34123/

Tips

Even though the RMSE is generally the preferred performance measure for regression tasks, in some contexts you may prefer to use another function. For example, suppose that there are many outlier districts. In that case, you may consider using the Mean Absolute Error.

Computing the root of a sum of squares (RMSE) corresponds to the Euclidian norm: it is the notion of distance you are familiar with. It is also called the ℓ2 norm, noted ‖ · ‖2 (or just ‖ · ‖).

Handling Text and Categorical Attributes

Converts classes into numbers	from sklearn.preprocessing import LabelEncoder
	encoder = LabelEncoder()
	housing_cat_encoded = encoder.fit_transform(columns with categories)
Turns an a categorical atribute into a sparse matrix where each column is a class and each row an observation	from sklearn.preprocessing import OneHotEncoder
	encoder = OneHotEncoder()
	housing_cat_1hot = encoder.fit_transform(housing_cat_encoded.reshape(-1,1))

One issue with this representation is that ML algorithms will assume that two nearby values are more similar than two distant values.

Visualizing data

Visualizing data (cont)

places a legend on the axis	plt.legend()
Plot with histograms and scatter plots	from pandas.tools.plotting import scatter_matrix
	scatter_matrix(housing[list of columns], figsize=(12, 8))

some attributes have a tail-heavy distribution, so you may want to transform them (e.g., by computing their logarithm)

Feature Scaling

from sklearn.pipeline import Pipeline	takes a list of name/estimator pairs defining a sequence of steps. All but the last estimator must be transformers
StandardScaler()	

Machine Learning algorithms don't perform well when
the input numerical attributes have very different scales

Training and Evaluating on the Training Set

Correlations

correlation matrix	data.corr()

Data cleaning

Drops rows with NA values	housing.dropna(subset=["total_bedrooms"])
DReturn the data set without a column or row (in this case it is a column)	housing.drop("total_bedrooms", axis=1)
fills NA values with the corresponding values	housing["total_bedrooms"].fillna(value)
Imputer	from sklearn.impute import SimpleImputer

Data cleaning (cont)

Replace missing values using a descriptive statistic (e.g. mean, median, or most frequent) along each column, or using a constant value	imputer = SimpleImputer(strategy="median")
The imputer has simply computed the median of each attribute and stored the result in its statistics_ instance variable.	imputer.fit(data)
Returns values that we computed	imputer.statistics_
Transform the missing value into corresponding value (return numpy array)	X = imputer.transform(housing_num)
Transform it back to a data frame	housing_tr = pd.DataFrame(X, columns=housing_num.columns)

Scatter plot	data.plot(kind="scatter", x="longitude", y="latitude", aplha=0.1 (makes the points transparent, thus allowing the visualization of high density places), s=column (determines size of the points), cmap=plt.get_cmap("jet") (color scheme), colorbar=True (makes a color bar appear), label='pop' (label of the points),c='column'(which column the circles will base its collor off))