Train In Data

Imputation with pandas

Pandas suppors basic imputation

- Mean / median imputation
- Arbitrary number imputation
- Frequent category imputation
- Arbitrary category imputation, i.e., “missing”
- Missing indicators

fillna

pandas.DataFrame.fillna

```
DataFrame.fillna(value=None, *, method=None, axis=None, inplace=False,
limit=None, downcast=None)
```

[source]

Fill NA/NaN values using the specified method.

Parameters: **value** : ***scalar, dict, Series, or DataFrame***

Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). Values not in the dict/Series/DataFrame will not be filled. This value cannot be a list.

pandas - mean

```
imp_dict = X_train.mean().to_dict()

X_train.fillna(imp_dict, inplace=True)
X_test.fillna(imp_dict, inplace=True)
```

- On covariance.

pandas - median

```
imp_dict = X_train.median().to_dict()

X_train.fillna(imp_dict, inplace=True)
X_test.fillna(imp_dict, inplace=True)
```

pandas - mode

```
imp_dict = X_train.mode().iloc[0].to_dict()

X_train.fillna(imp_dict, inplace=True)
X_test.fillna(imp_dict, inplace=True)
```

pandas – arbitrary values

```
imp_dict = {
        “Age”: 99, “Income”: -1,
        “Color”: “black”, “Make”: None,
}
X_train.fillna(imp_dict, inplace=True)
X_test.fillna(imp_dict, inplace=True)
```

pandas - advantages

- Convenient if working with dataframes
- Can impute feature subsets

pandas - limitations

- Does not store learned parameters
- Does not store the names of the variables to impute

Pandas: missing indicators

pandas.isna

pandas.isna(*obj*) [source]

Detect missing values for an array-like object.

This function takes a scalar or array-like object and indicates whether values are missing (`NaN` in numeric arrays, `None` or `NaN` in object arrays, `NaT` in datetimelike).

Parameters: **obj : *scalar or array-like***

Object to check for null or missing values.

Returns: **bool or array-like of bool**

For scalar input, returns a scalar boolean. For array input, returns an array of boolean indicating whether each corresponding element is missing.

Pandas: missing indicators

```
indicators = [f"{var}_na" for var in X_train.columns]


X_train[indicators] = X_train.isna().astype(int)
X_test[indicators] = X_test.isna().astype(int)
```

Accompanying Jupyter Notebook

- Jupyter Notebooks in **pandas folder**
- Demo of different imputation methods

Train In Data

THANK YOU

www.trainindata.com