CLASSIFICATION

2020-04

Magsbiost

Machine Learning

SUPERVISED

CLASSIFICATION

TARGET:

Discrete
Categorical

SUB-METHODOLOGIES:

Time series
Anomaly detection
NLP

ALGORITHMS:

Logistic Regression
SVM
Decision Tree
Random Forest
KNN
Neural Network

REGRESSION

TARGET:

Continuous
numeric, ordered

SUB-METHODOLOGIES:

time series
anomaly detection

ALGORITHMS:

OLS (linear regression)
GLM (tweedie regressor)
Polynomial Regression
Support Vector Regressor
Decision Tree Regressor
Neural Networks
LASSO, LARS

REINFORCEMENT

SUB-METHODOLOGIES

Anomaly Detection
NLP
Recommender Systems

ALGORITHMS:

Collaborative Filtering
Content Filtering
Neural Networks

UNSUPERVISED

CLUSTERING

SUB-METHODOLOGIES:

Anomaly Detection
NLP

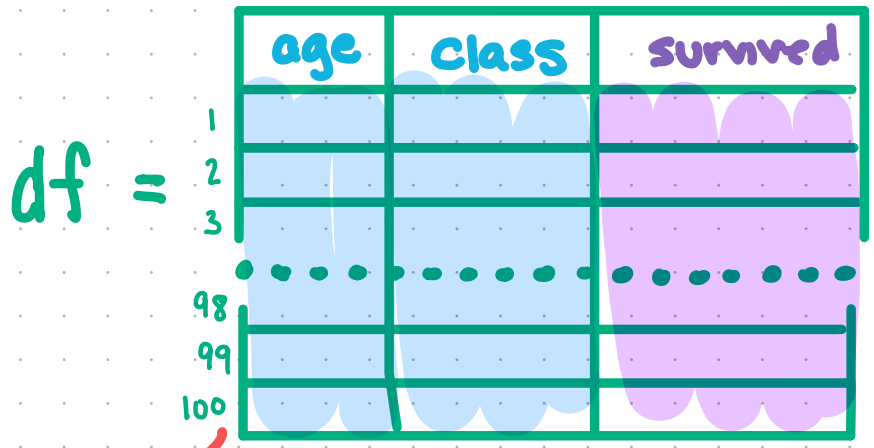
ALGORITHMS:

K-Means
DBSCAN
Hearchical

DIMENSIONALITY REDUCTION

PCA
t-SNE
factor Analysis

Splitting & Sampling Dataframes



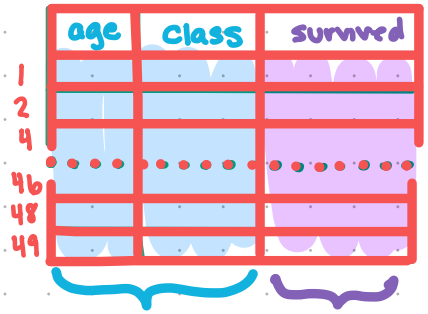
3x100 → observations

2 features, indep. vars, predictors, attributes

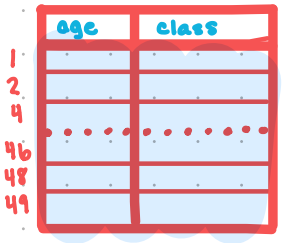
1 dependent var, target, outcome variable.

50 Rows

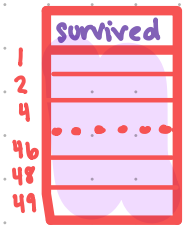
train 3x50



X_train 2x50

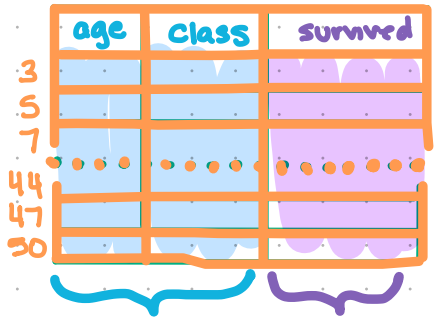


y_train 1x50

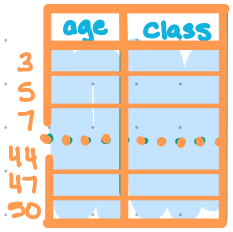


30 rows

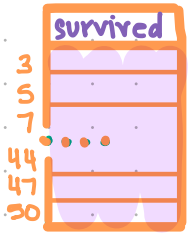
validate 3x30



X_validate 2x30

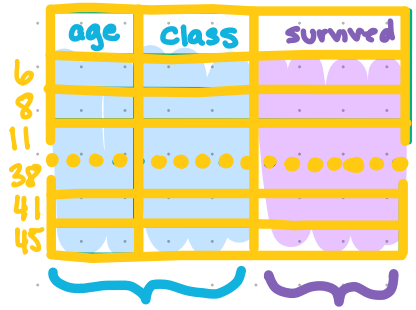


y_validate 1x30

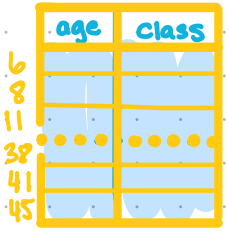


20 rows

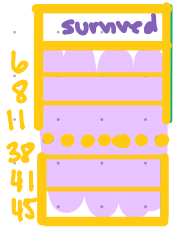
test 3x20



X_test 2x20



y_test 1x20



A Sample Flow

Acquire.py

```
myfile.candas as pd
def acquire_titanic():
    df = pd.read_csv('titanic.csv')
    return df
```

Prepare.py

```
Import pandas as pd
from sklearn.modelselection import train_test_split
def clean_titanic():
        <handle nulls>
        <handle data errors>
        <handle anomalies>
        <new features>
        <other things to prepare your data before splitting>
        return df

def split_titanic():
        train_validate, test = train_test_split()
        train, validate = train_test_split()
        return train, validate, test

def prep_titanic():
        df = clean_titanic()
        train, validate, test = split_titanic()
```

Explore.ipynb

```
Import pandas as pd
Import acquire
Import prepare
Import matplotlib.pyplot as plt
Import seaborn as sns

df = acquire.acquire_titanic()
train, validate, test = prepare.prep_titanic()
```

You Are Here!

#Explore

1. Univariate
2. Bivariate
3. Multivariate / Q&A

Draw Conclusions
Select Features

#Model

1. Fit on TRAIN
2. Evaluate on train, validate
3. Repeat with new Model (features, Algorithm, hyperparams)
4. Select Best Model
5. Evaluate on test.

#Deliver

1. Create a final report with highlights, conclusions, recommendations
2. Create a slide presentation to deliver of analysis results, highlights, conclusions, recommendations.

EXPLORE

Univariate Exploration - Exploration of 1 variable at a time

WHY?

1. ID outliers — Anomalies / Errors
2. ID imbalanced data
   - target
   - Features with little-No Entropy

WHAT?

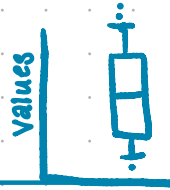
1. Document takeaways, findings
2. Document action needed, like "Replace data error with _"
3. Document new Questions to ask the data
4. Return to prepare as needed.
5. Progress to Bivariate Exploration.

HOW?

Numeric DTypes

1. descriptive stats - df.describe()
2. histogram - Series.hist()

3. boxplot - plt.boxplot() / sns.boxplot()



Categorical/Discrete Dtypes

1. Frequencies - series.value_counts()
2. Bar Plots - sns.countplot()

EXPLORE

Bivariate Exploration - Exploration of 2 variables at a time

WHY?

1. IDentify Variable Interdependence (You don't want that!)
2. IDentify relationships between

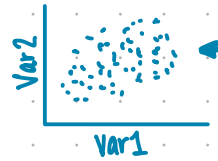
[Independent Variables: predictors, features] ⇔ [dependendent variable, outcome, target]

WHAT?

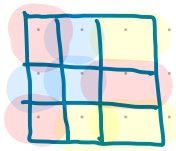
1. Document takeaways, findings
2. Document action needed, like "Drop var1 due to interdependence"
3. Document new Questions to ask the data
4. Return to prepare as needed.
5. Progress to Multivariate Exploration.

HOW?

Numeric x Numeric Dtypes

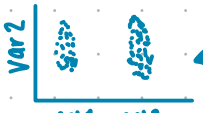
Scatterplot 

← sns.scatterplot()

heatmap 
← sns.heatmap()
sns.lineplot()
sns.lmplot()
sns.pairplot()

Statistical tests → pearson's R (linear corr.) & spearmans (non-linear)

Discrete x Numeric Dtypes

Swarmplot 

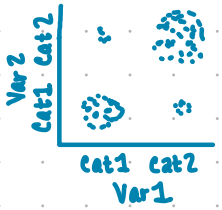
← sns.swarmplot()
sns.barplot()
sns.stripplot()

Violinplot 

← sns.violinplot()
sns.boxenplot()

Statistical tests → t-test, ANOVA, Mann-Whitney U

Discrete x Discrete

Swarmplot 

sns.swarmplot()

Crosstab pd.crosstab()

Statistical test → χ^2 (chi-square)

EXPLORE

Multivariate Exploration - Explore 3+ variables at a time

WHY?

Identify relationships between

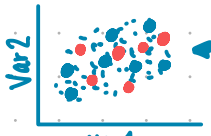
[Independent variables: predictors, features] & [dependent variable: outcome, target]

WHAT?

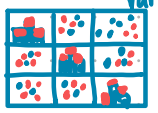
1. Document takeaways, findings, conclusions
2. Document initial recommendations
3. Document new questions to ask the data
4. finalize features to move forward into modeling

HOW?

Numeric x Numeric Dtypes

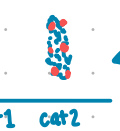
Scatterplot 

← sns.scatterplot()

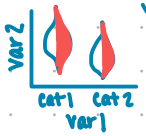
Pairplot  sns.pairplot()

Statistical tests → control for 1 group & Run
pearson's R (linear corr.) & spearmans (non-linear)

Discrete x Numeric Dtypes

Swarmplot 

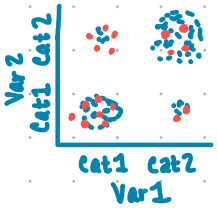
← sns.swarmplot()
sns.barplot()
sns.stripplot()

Violinplot 

← sns.violinplot()
sns.boxenplot()

Control for 1 cat. variable, test the other.
Statistical tests → t-test, ANOVA, Mann-Whitney U

Discrete x Discrete

Swarmplot 

sns.swarmplot()

Statistical test → Control for 1 categorical variable & test within the other 2.
X^2 (Chi-Square)