

Machine Learning Workshop Cheat Sheet

Lab 1

Exercise 1

`data.describe()`

Exercise 2

`sns.distplot(data['radius_mean'])`

Lab 2:

Retail: Optimize inventory spacing and placing

Describe approach: group items together that are frequently purchased together and/or have same type of purchaser

Target variable: none

Supervised or unsupervised: unsupervised

Model type: clustering

Data needed: purchasing history

Model evaluation: Time in store goes down while sales go up

Sports: Increase number of season ticket holders

Describe approach: classify customers as likely/unlikely to become a season ticket holder, target advertisement accordingly

Target variable: season ticket purchaser/non-purchaser

Supervised or unsupervised: supervised

Model type: classification

Data needed: profile data and transaction history of season ticket purchaser/non-purchaser

Model evaluation: Number of season ticket holders go up, advertising spend goes down

Airline: Project seasonal flight demand

Describe approach: predict flight demand fluctuations based on previous seasonal data plus current/upcoming conditions

Target variable: tickets sold by time interval

Supervised or unsupervised: supervised

Model type: regression

Data needed: previous flight demand, weather data, flight delay data

Model evaluation: less empty seats in planes, less plane down time

Lab 3:

Exercise 1

```
select_cols = [  
    'smoothness_mean',  
    'compactness_mean',  
    'Symmetry_mean',  
    'fractal_dimension_mean'  
]  
feature_box_plot(select_cols, data)
```

Exercise 2

```
train_df, test_df = train_test_split(final_df, test_size=0.25, random_state=42,  
stratify=final_df['diagnosis'])  
compare_test_train(train_df, test_df)
```

Lab 4:

Exercise 1:

create and fit new model, please fill in missing arguments

```
logregV2 = LogisticRegression(solver='lbfgs', max_iter=10000, C=1570.2901247293776)
logregV2.fit(train[features], train['diagnosis_code'])
```

```
# evaluate model, fill in missing arguments
model_metrics(test, logregV2, features)
```

```
# create auc plot, fill in missing arguments
create_roc_curve(test, logregV2)
```

Exercise 2:

```
# try out different thresholds, please fill in missing arguments
metric_new_threshold = model_metrics_threshold(test, logreg, features, threshold=0.8)
```