תהליך הכנת הנתונים

ייבוא ספריות

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay, classification_report
```

Penguin Dataset

```
#load dataset
df=pd.read_csv('https://raw.githubusercontent.com/arielb30/datasets/main/penguins.csv')
df
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female
...	...	...	...	...	...	...	...
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female

344 rows × 7 columns

```
df['species'].unique()
```

```
array(['Adelie', 'Gentoo', 'Chinstrap'], dtype=object)
```

```
df['island'].unique()
```

```
array(['Torgersen', 'Biscoe', 'Dream'], dtype=object)
```

```
df['sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

טיפול בערכים חסרים

איתור ערכים חסרים

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
df.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

טיפול בערכים חסרים ברמת העמודה

```
#מחיקת עמודה/ות
df.drop('Cabin',axis=1,inplace=True)

#מחיקת שורות עם ערכים חסרים, בעמודה/ות
df.dropna(subset=['Cabin'], axis=0, inplace=True)
```

טיפול בערכים חסרים ברמת העמודה

```
#מלוי ערכים חסרים בעמודה נומרית
()mean_age = df['Age'].mean
df['Age']=df['Age'].fillna(mean_age)

#מלוי ערכים חסרים בעמודת מחרוזת
df['Cabin'] = df['Cabin'].fillna('Missing')
```

טיפול בערכים חסרים ברמת הטבלה

```
# מחיקת שורות עם ערכים חסרים בטבלה
df.dropna(axis=0,inplace=True)

# מחיקת עמודות עם ערכים חסרים בטבלה
df.dropna(axis=1,inplace=True)
```

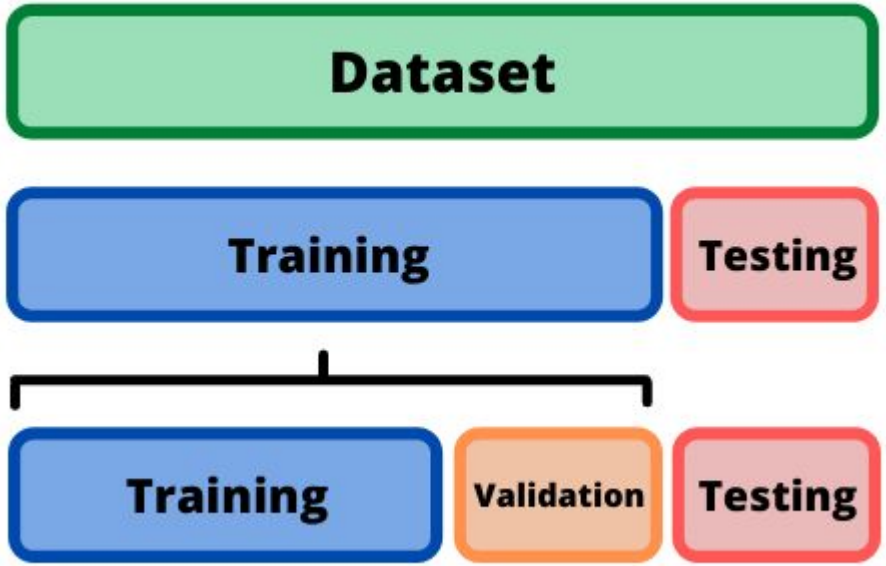
טיפול בערכים חסרים ברמת הטבלה

```
#מלוי ערכים חסרים בעמודות נומריות בטבלה
numerical_cols = df.select_dtypes(include=np.number).columns
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())

#מלוי ערכים חסרים בעמודות מחרוזת בטבלה
object_cols = df.select_dtypes(include='object').columns
df[object_cols] = df[object_cols].fillna('Missing')
```

חלוקה אימון ומבחן

חלוקה אימון ומבחן



בחירת מאפיינים ומשתנה מטרה

```
#Chose attributes
X=df.drop(['species','island','sex'],axis=1).to_numpy()
X.shape
```

```
(333, 4)
```

```
#chose target
y = df['island'].to_numpy()
y.shape
```

```
(333,)
```

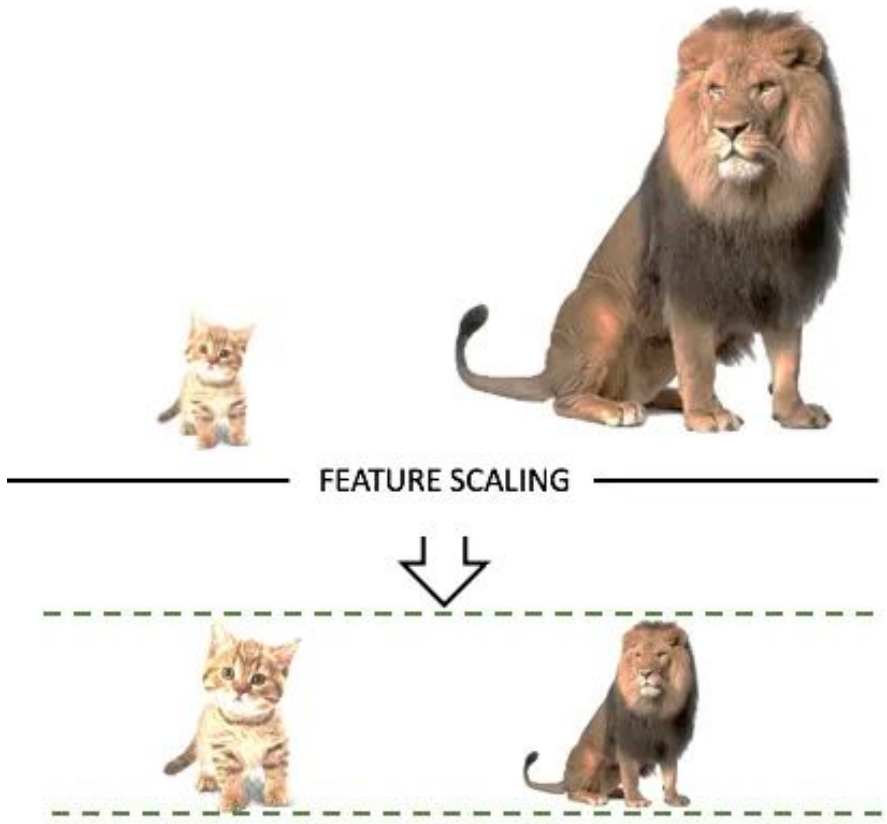
Train test split

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,\
                                               shuffle=True,random_state=42)
print("train: ", X_train.shape, y_train.shape)
print("test:  " , X_test.shape, y_test.shape)
```

```
train:  (266, 4) (266,)
test:   (67, 4) (67,)
```

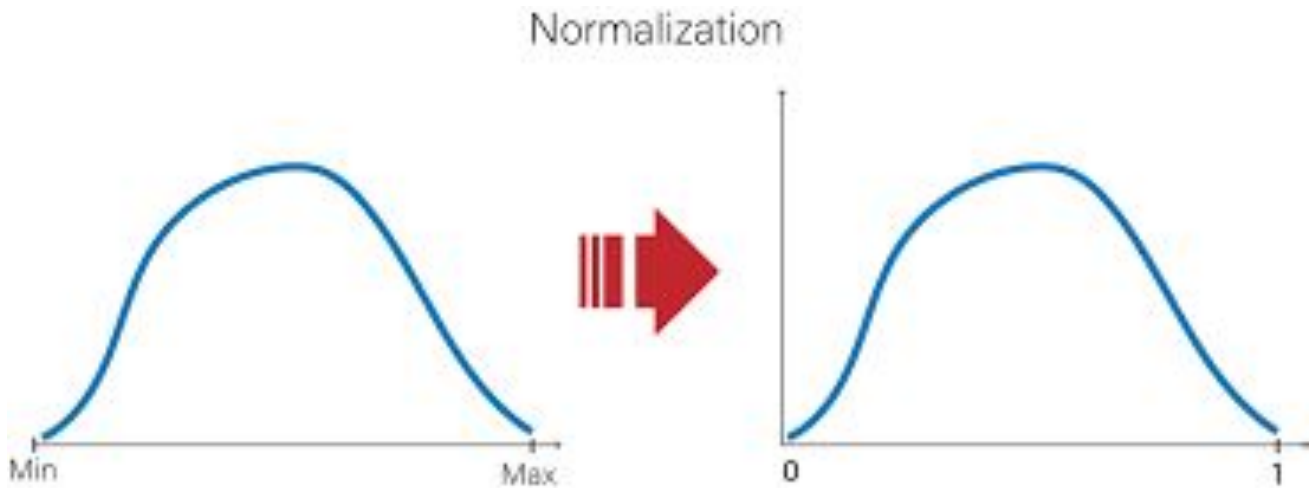
נרמול

Scaling the data



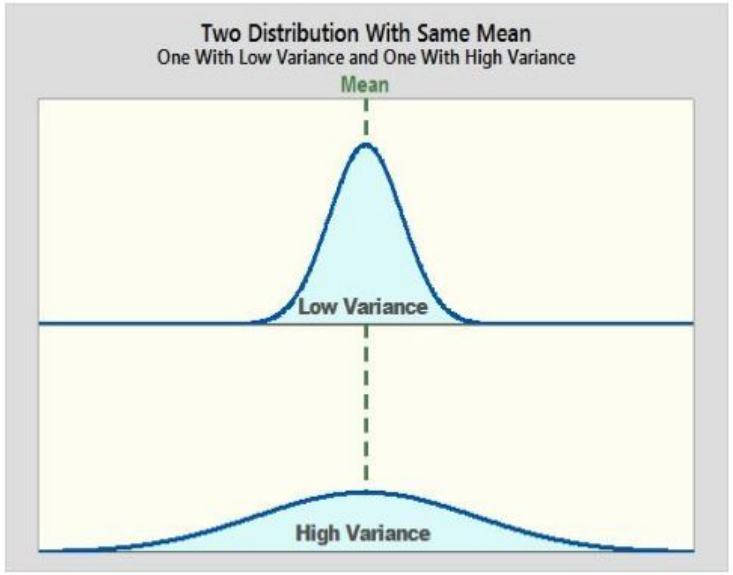
Max/Min Normalization

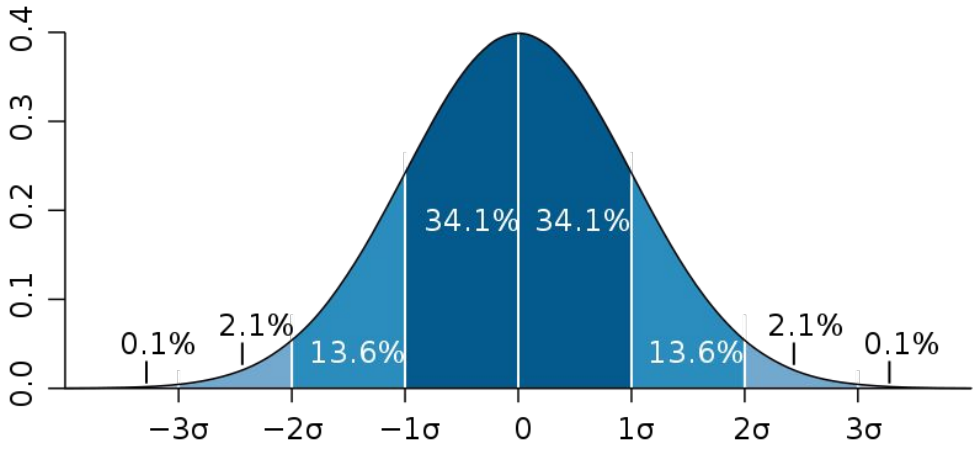
$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$



התפלגות נורמלית ומדדי פיזור

- שונות (variance)
- סטיית תקן (standard deviation)

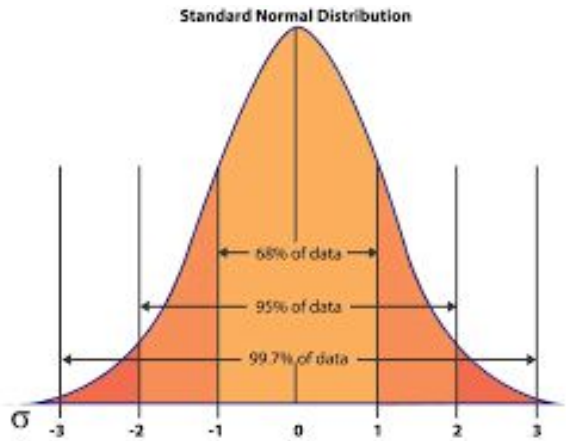




https://www.strictlybythenumbers.com/bias-vs-variance
https://en.wikipedia.org/wiki/Standard_deviation

Standardization

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation}\ (x)}$$



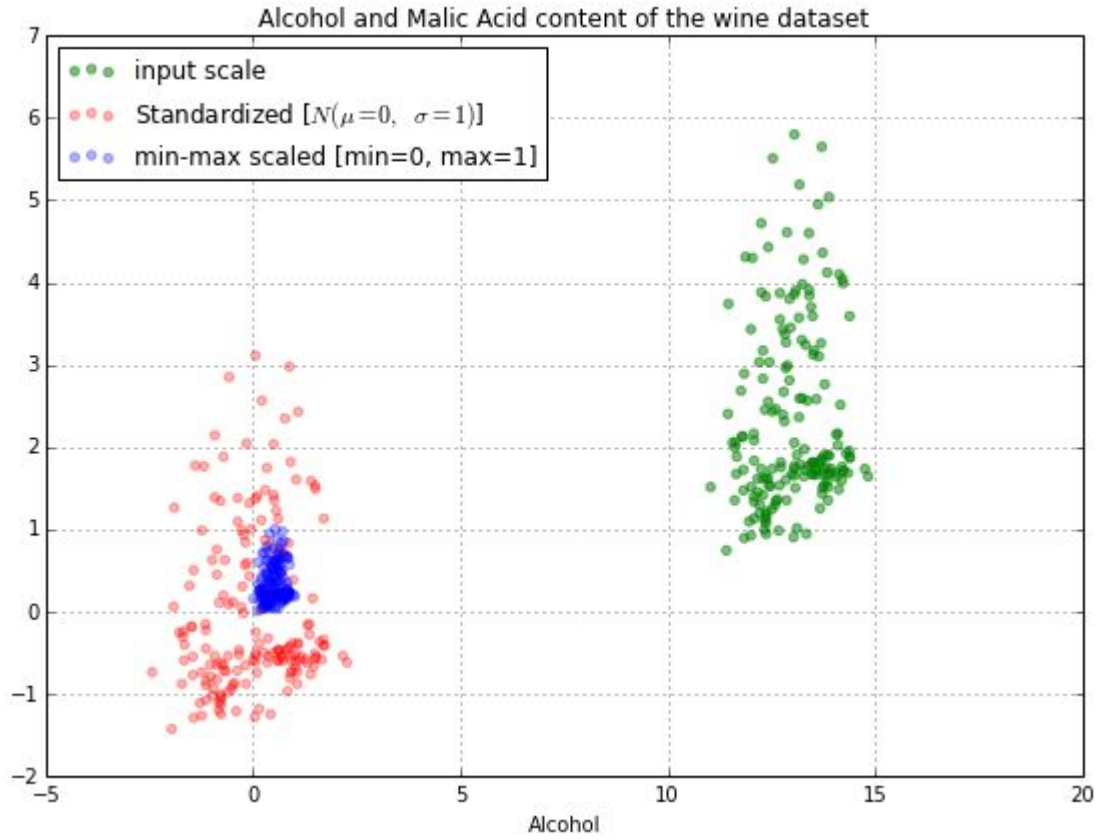
with mean:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Normalization (min-max scaling) vs. Standardization



Min Max Scaler

```
scaler = MinMaxScaler()
print('Min: {} Max: {}'.format(X_train.min(), X_train.max()))
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
print('Min: {} Max: {}'.format(X_train.min(), X_train.max()))
```

```
Min: 13.1 Max: 6050.0
Min: 0.0 Max: 1.0
```

Stanrdard Scaler

```
scaler = StandardScaler()
print('Min: {:.3f} Max: {:.3f}'.format(X_train.min(), X_train.max()))
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
print('Min: {:.3f} Max: {:.3f}'.format(X_train.min(), X_train.max()))
```

```
Min: 13.100 Max: 6050.000
Min: -2.214 Max: 2.869
```

שלבי האימון והמבחן

המסווג מותאם (fit) לנתוני האימון

```
# create KNN classifier
knn1 = KNeighborsClassifier(n_neighbors = 5)
# fit to train data
knn1.fit(X_train,y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

פרדיקציות ומדדי ביצוע על נתוני המבחן

```
#predict on test samples
y_pred = knn1.predict(X_test)
y_pred.shape
```

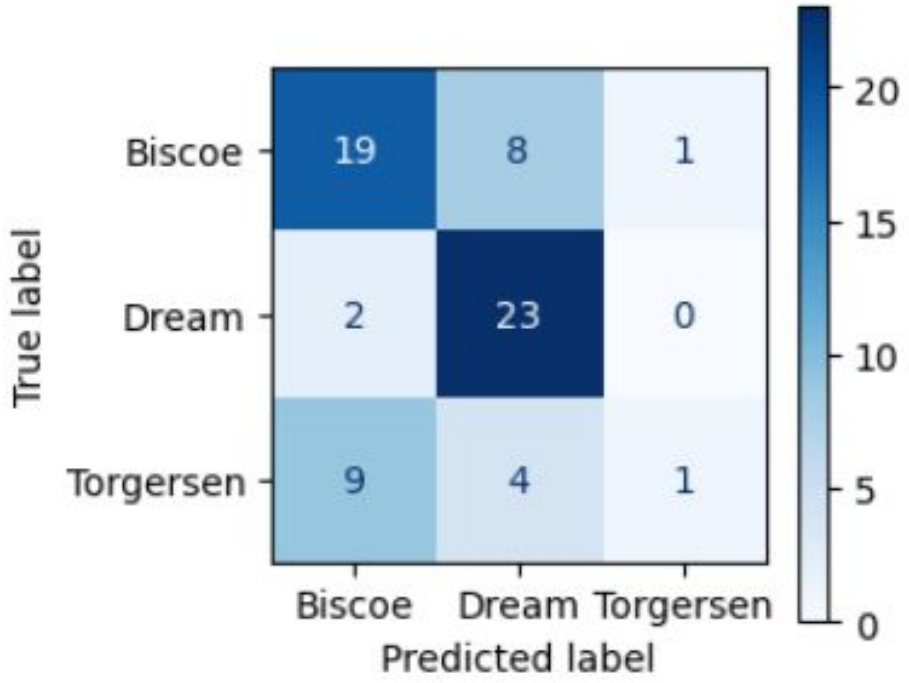
```
(67,)
```

```
#accuracy on test samples
accuracy_score(y_true=y_test, y_pred = y_pred)
```

```
0.6417910447761194
```

פרדיקציות ומדדי ביצוע על נתוני המבחן

```
#confusion matrix on test data
fig, ax = plt.subplots(figsize=(3, 3))
ConfusionMatrixDisplay.from_predictions(y_true=y_test,y_pred=y_pred,\
                                        values_format="d",cmap="Blues",ax=ax);
```



פרדיקציות ומדדי ביצוע על נתוני המבחן

```
#classification report on test data
print(classification_report(y_true=y_test, y_pred=y_pred))
```

	precision	recall	f1-score	support
Biscoe	0.63	0.68	0.66	28
Dream	0.66	0.92	0.77	25
Torgersen	0.50	0.07	0.12	14
accuracy			0.64	67
macro avg	0.60	0.56	0.52	67
weighted avg	0.61	0.64	0.59	67

Penguin Classification exercise

טענו מחברת Penguin Dataset

1. נקו נתונים חסרים
2. בחרו מאפיינים מספריים, ובחרו משתנה מטרה בדיד
3. חלקו לאימון ומבחן
4. נרמלו את הנתונים (בחרו שיטה)
5. בצעו fit על נתוני האימון
6. בצעו predict על נתוני המבחן
7. בדקו מדדי ביצוע על נתוני המבחן:
Accuracy, Confusion Matrix, Classification report

Penguin Regression exercise

טענו מחברת Penguin Dataset

1. נקו נתונים חסרים
2. בחרו מאפיינים מספריים, ובחרו משתנה מטרה מספרי
3. חלקו לאימון ומבחן
4. נרמלו את הנתונים (בחרו שיטה)
5. בצעו fit על נתוני האימון
6. בצעו predict על נתוני המבחן
7. בדקו מדדי ביצוע על נתוני המבחן (r2)

תודה על ההשתתפות