ML Models

Supervised Learning

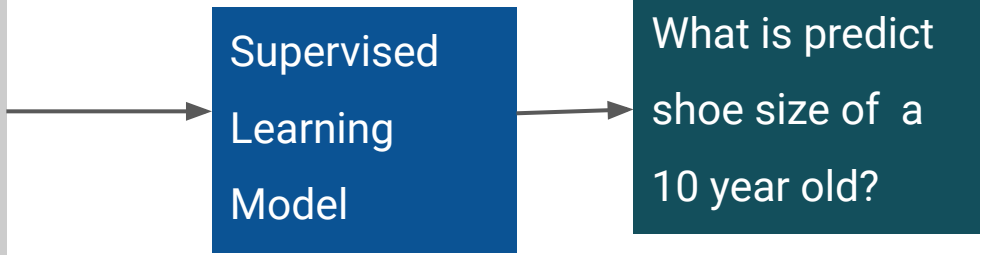
Unsupervised Learning

- **Supervised machine learning relies on labelled input and output training data.**
- **Unsupervised learning processes unlabelled or raw data**

Supervised Learning

- Regression Model
- Classification Model

Age(Input)	->	Shoe Size(Output)
5	->	6
9	->	7
11	->	9
17	->	9.75



Regression Models

Dependent variables(inputs) ⇔ Independent variables(outputs)

Continuous

More specifically it focuses on how the dependent variable changes in relation to changes in independent variables

eg.The relationship between height and weight may be described by a linear regression model

Most commonly used types of Regression models are:

- Linear Regression
- Decision Tree
- Random Forests
- Neural Network

Classification Models

Discrete

Output values can only be between 0 and 1.

Most commonly used types of classification models are:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- Random Forest
- Neural Network

Unsupervised Learning

Patterns from input data without references to labeled outcomes

Type of unsupervised learning model :

- **Clustering**

Preview For Diabetes dataset

	A	B	C	D	E	F	G	H	I	J
1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome	
2	6	148	72	35	0	33.6	0.627	50	1	
3	1	85	66	29	0	26.6	0.351	31	0	
4	8	183	64	0	0	23.3	0.672	32	1	
5	1	89	66	23	94	28.1	0.167	21	0	
6	0	137	40	35	168	43.1	2.288	33	1	
7	5	116	74	0	0	25.6	0.201	30	0	
8	3	78	50	32	88	31	0.248	26	1	
9	10	115	0	0	0	35.3	0.134	29	0	
10	2	197	70	45	543	30.5	0.158	53	1	
11	8	125	96	0	0	0	0.232	54	1	
12	4	110	92	0	0	37.6	0.191	30	0	
13	10	168	74	0	0	38	0.537	34	1	
14	10	139	80	0	0	27.1	1.441	57	0	
15	1	189	60	23	846	30.1	0.398	59	1	
16	5	166	72	19	175	25.8	0.587	51	1	
17	7	100	0	0	0	30	0.484	32	1	
18	0	118	84	47	230	45.8	0.551	31	1	
19	7	107	74	0	0	29.6	0.254	31	1	
20	1	103	30	38	83	43.3	0.183	33	0	
21	1	115	70	30	96	34.6	0.529	32	1	
22	3	126	88	41	235	39.3	0.704	27	0	
23	8	99	84	0	0	35.4	0.388	50	0	

diabetes

Choose the model for Diabetes dataset

Diabetes dataset know both input and output.

So, we can use Supervised Learning Models.

It’s output values are 0 and 1. So, we can use ‘Classification’ algorithm.

So, will train this dataset by using **Support Vector Machine** Classification algorithms.

Support Vector Machine is one of the ‘Classification’ algorithms.

Prediction For Diabetes dataset

```
# loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv("/diabetes.csv")

# print shape of dataset (number of rows and columns)
dataset_shaped = diabetes_dataset.shape
print("dataset shaped : ", dataset_shaped)
#dataset shaped :  (768, 9)

# print column list of dataset
print("columns : ", list(diabetes_dataset.columns))
#columns :  ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
'DiabetesPedigreeFunction', 'Age', 'Outcome']
```

```
output_value_count = diabetes_dataset["Outcome"].value_counts()
print(output_value_count)
0    500
1    268
Name: Outcome, dtype: int64

# 0 is "Non-Diabetic" and 1 is "Diabetic"

#defined output is 'Outcome'
# separating the data and labels
X = diabetes_dataset.drop(columns="Outcome", axis=1)
Y = diabetes_dataset["Outcome"]

#X is inputs and Y is output.
#X has 768 rows and 8 columns (removed 'Outcome' column)
#Y has 768 rows and 1 columns (only have 'Outcome' column)
```

```
# Data Standardization
scaler = StandardScaler()
scaler.fit(X)

standardized_data = scaler.transform(X)

X = standardized_data # changed standardized data from original data

#Need to split training data and testing data based on the original data to train the model.

# Train Test Split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
print(X.shape, X_train.shape, X_test.shape)
#(768, 8) (614, 8) (154, 8)

# (test_size = 0.2) means we will use 20% of data for test data. So,  will use 80% for training data
# According to the condition, training data is 614 rows and testing data is 154 rows.
```

```
# Train the model
classifier = svm.SVC(kernel="linear")

# training the support vector machine classifier
classifier.fit(X_train, Y_train)

# Model Evaluation
# Accuracy score (if it is over 75, it has good performance accuracy.)

X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print("Accuracy score of the training data : ", training_data_accuracy)
# Accuracy score of the training data : 0.7866449511400652

X_test_prediction = classifier.predict(X_test)
testing_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print("Accuracy score of the test data : ", testing_data_accuracy)
 #Accuracy score of the test data :  0.7727272727272727
```

So, we can define that it has good performance accuracy.

Prediction each row of diabetes data

```
# can change input data at here
input_data = (1, 85, 66, 29, 0, 26.6, 0.351, 31) #second row of diabetes dataset

# changing the input data to numpy array
input_data_as_np_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_np_array.reshape(1, -1)

# standardize the input data
std_data = scaler.transform(input_data_reshaped)

prediction = classifier.predict(std_data)
print("predicted data : ", prediction) #predicted data :  [0]

if prediction[0] == 0:
  return "The person is not diabetic."
else:
  return "The person is diabetic."
```