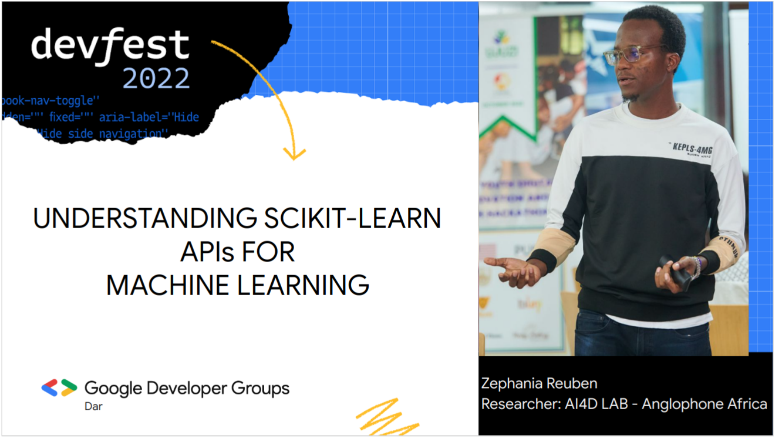


devfest
2022
book-nav-toggle"
dden="" fixed="" aria-label="Hide
Hide side navigation"

UNDERSTANDING SCIKIT-LEARN APIs FOR MACHINE LEARNING

Google Developer Groups
Dar

Zephania Reuben
Researcher: AI4D LAB - Anglophone Africa

Meet Zephania Reuben [Nsoma]

- Software Developer - Beem Africa [2021 - 2022]
  - Python & Data Science
- Speaker/Facilitator - Python & Artificial Intelligence
  - PyCon 2019, 2020, 2021 by Python Community Tanzania
  - IndabaX Tanzania 2021 by Deep Learning Indaba
  - Data Science Training 2021, 2022 by Vema Academy
  - DevFestDar 2021 by GDG Dar es Salaam
  - Teens in AI [Tanzania] 2021, 2022 by Ujuzi Forum
  - Advanced AI Training 2022 by AI4D Lab - Anglophone Africa
  - EnhanceMind AI Conference 2022 by CameLabs
- Organizing Committee
  - AI and Life 2019 by TeleSoftAI
  - UmojaHack 2020 by Zindi Africa
  - PyCon Tanzania 2021, 2022 by Python Community Tanzania
  - IndabaX Tanzania 2022 by Deep Learning Indaba
  - EnhanceMind AI Conference 2022 by CameLabs

Who is Zephania Now?

Aspiring AI Researcher & Consultant

Outline

1. Introduction
2. Types of Machine Learning
3. Machine Learning WorkFlow
4. Python Libraries for DS and ML
5. Introduction to Scikit-Learn
6. Scikit-Learn APIs Design
7. Linear Models with Scikit-Learn

Problem 1

"You need to predict how much user “A” will like a movie that she hasn't seen based on her ratings of movies that she has seen.”

Problem 2

"You need to predict how much transaction “T” is likely to be fraudulent based on previous transactions.”

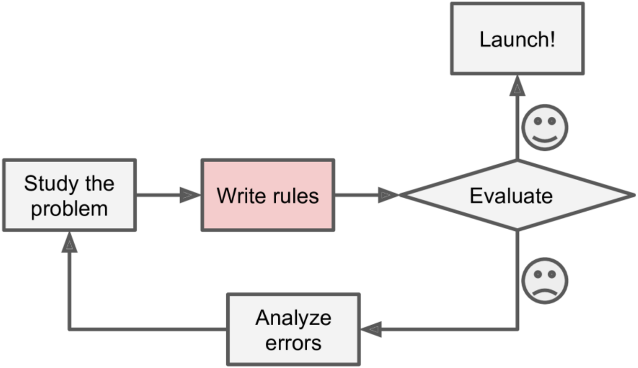
Ways to solve

- Traditional Methods
- Machine Learning

Traditional Methods

- Complex rules
- Hard to maintain

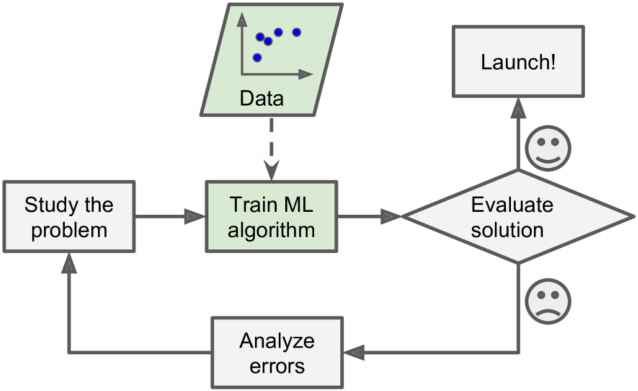
Traditional Methods



Machine Learning

- Automatic pattern learning
- Ease to maintain
- Adopt to changes
- More accurate

Machine Learning



Machine Learning

What does it mean to learn?

- In Machine Learning an important concept is "generalization", the ability to generalize.

Machine Learning

A computer program is said to learn from experience E with respect to some task T and some performance P, if its performance on T, as measured by P, improves with experience E.

- Tom Mitchell, 1997.

Checker Learning Problem

- Task T : Playing Checker.

- Experience E: Playing practice game against itself.

- Performance Measure P: % of games won against opponents.

Types of Machine Learning

- Supervised Machine Learning
- Unsupervised Machine Learning
- Semi-Supervised Machine Learning
- Reinforcement Learning

Machine Learning Algorithms

Supervised Machine Learning Algorithms

- Training data includes the desired solutions called labels.

Machine Learning Algorithms

Some Supervised Machine Learning Algorithms

- Linear & Logistic Regression
- Decision Trees
- Support Vector Machines
- Random Forest & Ensemble Models
- K-Nearest Neighbors
- Neural Networks

Machine Learning Algorithms

Unsupervised Machine Learning Algorithms

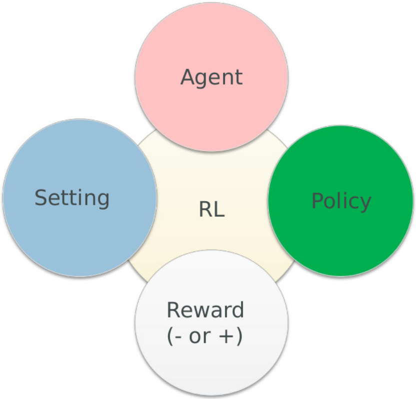
- They only extracts pattern from the provided data during learning.

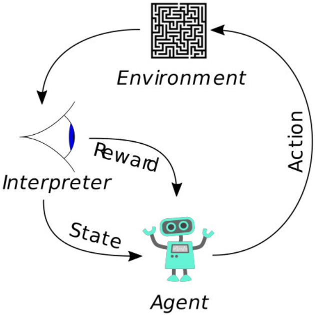
Machine Learning Algorithms

Some Unsupervised Machine Learning Algorithms

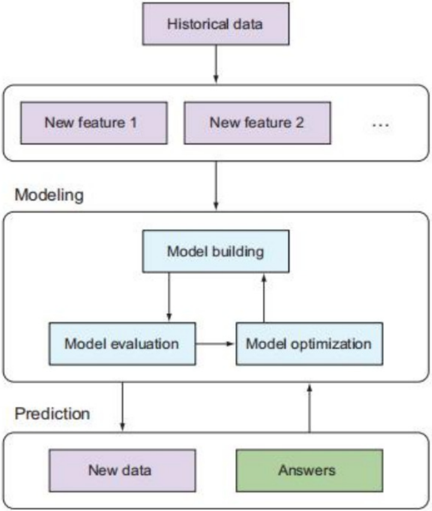
- Clustering
- Anomaly Detection
- Dimensionality Reduction

Reinforcement Learning Algorithm

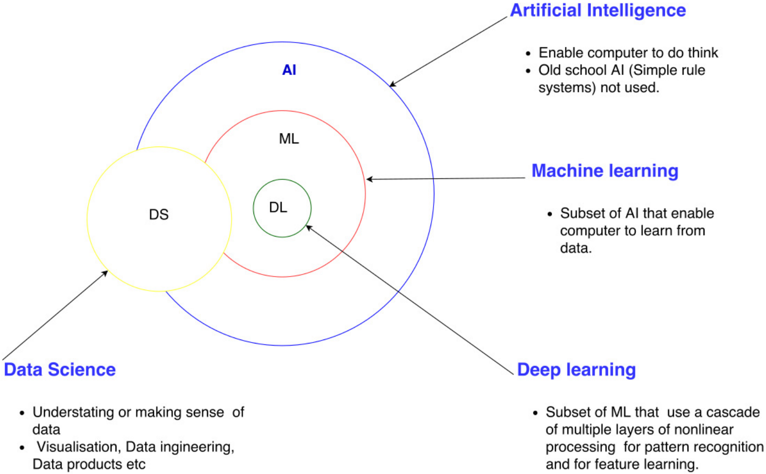




Simplified Machine Learning WorkFlow



What is the difference between DS, ML, AI, and DL?



Artificial Intelligence

- Enable computer to do think
- Old school AI (Simple rule systems) not used.

Machine learning

- Subset of AI that enable computer to learn from data.

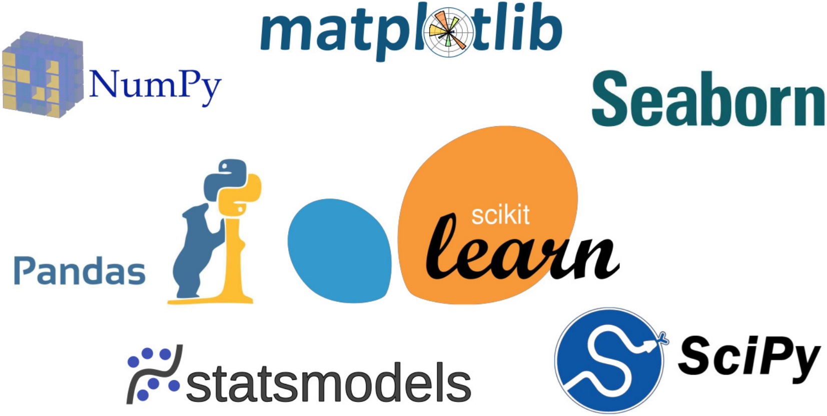
Data Science

- Understating or making sense of data
- Visualisation, Data ingineering, Data products etc

Deep learning

- Subset of ML that use a cascade of multiple layers of nonlinear processing for pattern recognition and for feature learning.

Python Libraries for DS and ML



Machine Learning with Scikit-Learn

- Scikit-Learn is a free software machine learning library for the Python programming language.
- It features various classification, regression, and clustering algorithms.
- It provides many unsupervised learning algorithms.It's built upon some of the technologies such as :
  - NumPy
  - Pandas and
  - Matplotlib
- It is also used for data wrangling(manipulation) and data analysis.

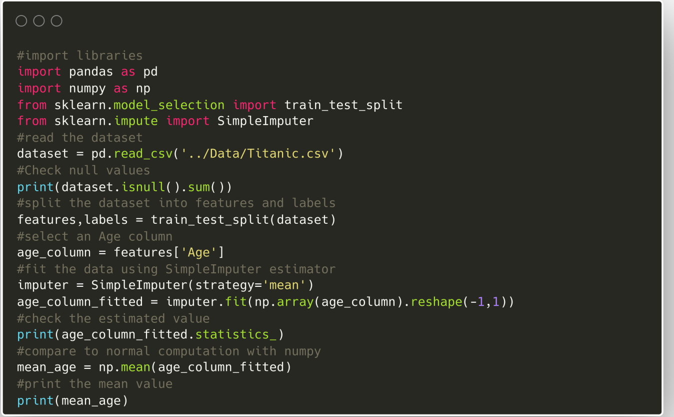
Scikit-Learn: APIs Design

- Scikit-Learn library is organized in three fundamental APIs(Interfaces):
  - Estimator
  - Predictor
  - Transformer

Scikit-Learn: Estimator

- Estimator: this is the core interface of Scikit-Learn, estimator objects are used to perform estimation of some parameters based on dataset.
- All learning algorithms, whether supervised or unsupervised, classification,regression, or clustering, implement the estimator interface and expose a fit() method.
- The fit() method takes the dataset also sometimes labels for supervised learning and in this way estimator "learns" how to make predictions on unseen data for supervised learning.
- For instance an imputer object in the code snippets below is an estimator:-

Scikit-Learn: Estimator

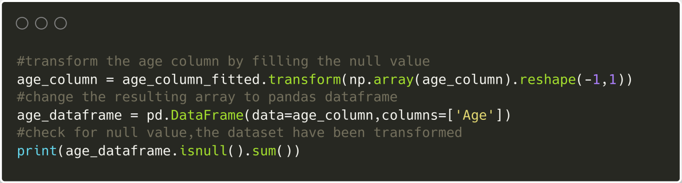


```
#import libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
#read the dataset
dataset = pd.read_csv('../Data/Titanic.csv')
#Check null values
print(dataset.isnull().sum())
#split the dataset into features and labels
features,labels = train_test_split(dataset)
#select an Age column
age_column = features['Age']
#fit the data using SimpleImputer estimator
imputer = SimpleImputer(strategy='mean')
age_column_fitted = imputer.fit(np.array(age_column).reshape(-1,1))
#check the estimated value
print(age_column_fitted.statistics_)
#compare to normal computation with numpy
mean_age = np.mean(age_column_fitted)
#print the mean value
print(mean_age)
```

Scikit-Learn: Transformer

- Transformer: Transformer extends estimator class and transformer objects they can also transform a dataset.
- Transformation is performed by the method transform() which passes the dataset as it's parameter.
- It returns the transformed dataset.
- Transformation is done based on the learned parameters, also a method fit_transform() can be used to perform both fit and transformation.
- Refer to the code snippets below:-

Scikit-Learn: Transformer

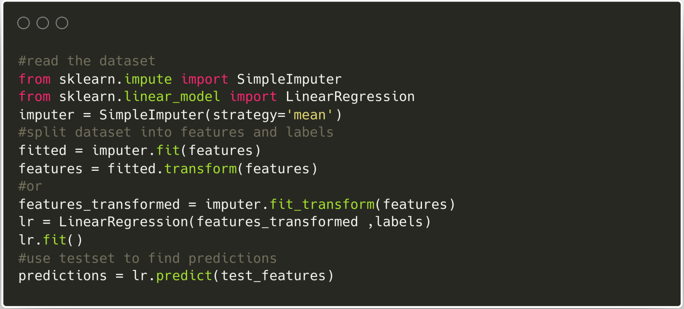


```
#transform the age column by filling the null value
age_column = age_column_fitted.transform(np.array(age_column).reshape(-1,1))
#change the resulting array to pandas dataframe
age_dataframe = pd.DataFrame(data=age_column,columns=['Age'])
#check for null value,the dataset have been transformed
print(age_dataframe.isnull().sum())
```

Scikit-Learn: Predictor

- Also predictor class extends the estimator interface, and for a given model to "work" it must implement (and expose) a predict() method.
- Estimators are capable of making predictions given dataset.
- For example the LinearRegression model is a predictor.
- Predict method takes a dataset of new instance and returns a dataset corresponding to predictions.
- Also has score method to measure the quality of predictions given a test set.
- Refer to the code snippets below:-

Scikit-Learn: Predictor



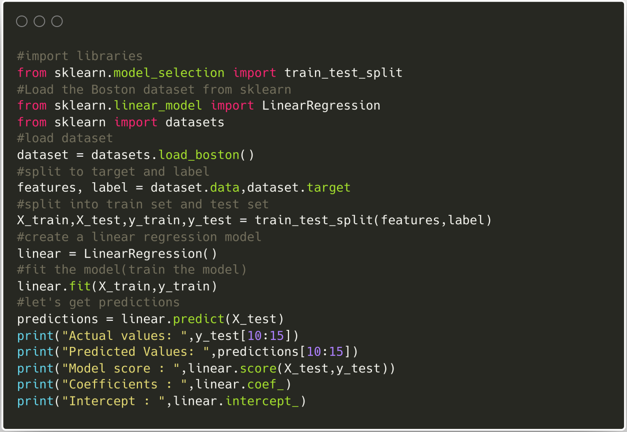
```
#read the dataset
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
imputer = SimpleImputer(strategy='mean')
#split dataset into features and labels
fitted = imputer.fit(features)
features = fitted.transform(features)
#or
features_transformed = imputer.fit_transform(features)
lr = LinearRegression(features_transformed ,labels)
lr.fit()
#use testset to find predictions
predictions = lr.predict(test_features)
```

Linear Models: Linear Regression

- This is model which is made up of simple linear function, and it is very easy to visualize.
- Traditional linear regression is the first, and therefore, probably the most fundamental model a straight line through data.
- A mathematical expression for linear regression is as follows:

$$y = a + bx$$

Linear Models: Linear Regression



```python
#import libraries
from sklearn.model_selection import train_test_split
#Load the Boston dataset from sklearn
from sklearn.linear_model import LinearRegression
from sklearn import datasets
#load dataset
dataset = datasets.load_boston()
#split to target and label
features, label = dataset.data,dataset.target
#split into train set and test set
X_train,X_test,y_train,y_test = train_test_split(features,label)
#create a linear regression model
linear = LinearRegression()
#fit the model(train the model)
linear.fit(X_train,y_train)
#let's get predictions
predictions = linear.predict(X_test)
print("Actual values: ",y_test[10:15])
print("Predicted Values: ",predictions[10:15])
print("Model score : ",linear.score(X_test,y_test))
print("Coefficients : ",linear.coef_)
print("Intercept : ",linear.intercept_)
```

Linear Models: Logistic Regression

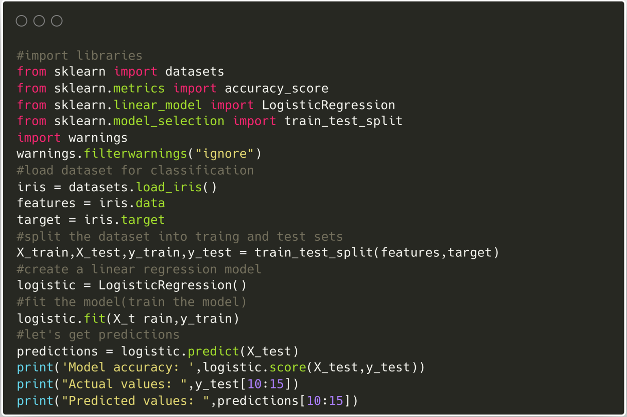
- It allows us to predict probability that an observation is of a certain class. It is been called regression but it is classification algorithm.
- It combines the linear equation and logit/sigmoid or logistic function.
- A mathematical expression for linear regression is as follows:

$$z = a + bx$$

Sigmoid function expressed as: $f(z) = \frac{1}{e^{-z}}$, then

$$y = \frac{1}{e^{-(a+bx)}}$$

Linear Models: Logistic Regression



```
#import libraries
from sklearn import datasets
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
#load dataset for classification
iris = datasets.load_iris()
features = iris.data
target = iris.target
#split the dataset into traing and test sets
X_train,X_test,y_train,y_test = train_test_split(features,target)
#create a linear regression model
logistic = LogisticRegression()
#fit the model(train the model)
logistic.fit(X_t rain,y_train)
#let's get predictions
predictions = logistic.predict(X_test)
print('Model accuracy: ',logistic.score(X_test,y_test))
print("Actual values: ",y_test[10:15])
print("Predicted values: ",predictions[10:15])
```

Hands on

Refer to this notebook here

Thank You!, Twitter: @nsomazr