Machine Learning

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions.

Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data.

Machine Learning

Unsupervised Learning

Dimensionality Reduction

- Meaningful Compression
- Structure Discovery
- Big data Visualistaion
- Feature Elicitation

Clustering

- Recommender Systems
- Targetted Marketing
- Customer Segmentation

Supervised Learning

Classification

- Image Classification
- Customer Retention
- Idenity Fraud Detection
- Diagnostics

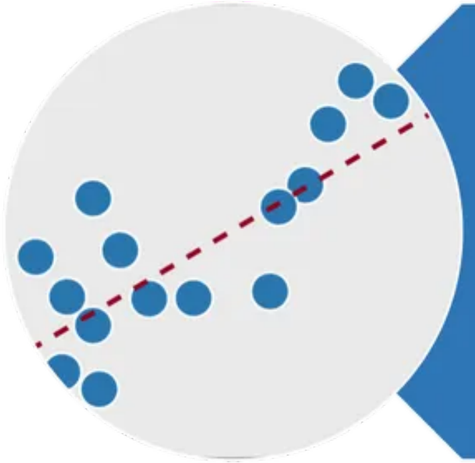
Regression

- Advertising Popularity Prediction
- Weather Forecasting
- Population Growth Prediction
- Market Forecasting
- Estimating life expectancy

Reinforcement Learning

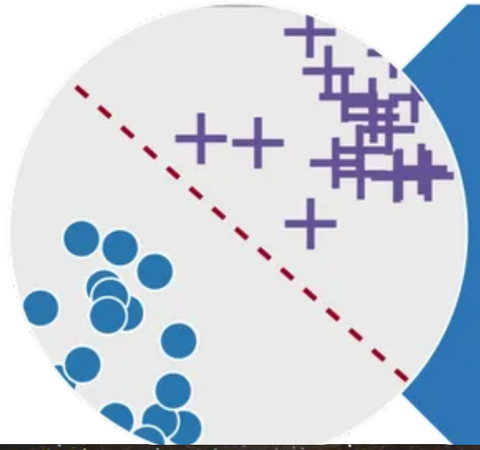
- Real-time decisions
- Game AI
- Robot Navigation
- Skill Acquisition
- Learning Tasks

Supervised Learning Algorithms



Regression

- Linear Regression
- Random Forest
- Multi-layer Perceptron
- AdaBoost
- Gradient Boosting
- Convolutional Neural Networks



Classification

- Logistic Regression
- Decision Tree
- KNN
- Support vector machines
- Naive Bayes
- Convolutional Neural Networks

Tools used in the project

Scikit - learn

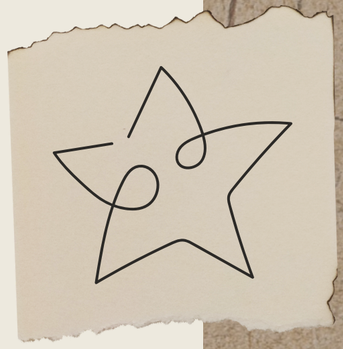
Scikit-learn offers simple tools for data-mining and analysis of data. It is open-source and runs on top of Scipy, numpy and matplotlib.

Numpy

Being an interpreter based language, Python suffered from the problem of low-speed in its operations. Therefore, in order to mitigate this issue, Numpy was introduced in 2006.

XGBoost

XGBoost is a package that provides an efficient implementation of the gradient boosting algorithm



Detecting Parkinson's Disease

USING EXTREME GRADIENT BOOSTING ALGORITHM

STEPS


STEP 01
Data collection
STEP 02
Data preparation
STEP 03
Choosing a model
STEP 04
Training
STEP 05
Evaluation
STEP 06
Parameter tuning
STEP 07
Prediction

Dataset used

parkinsons.data

This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds to one of 195 voice recordings from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to the "status" column which is set to 0 for healthy and 1 for PD.

Packages used

Numpy

Pandas

Min-Max scaler from
Sklearn.preprocessing

Train-test-split from
sklearn.model_selection

XGBClassifier from xgboost

Accuracy score from sklearn.metrics

Data Cleaning & Feature Engineering

In [5]:
```
features = df.loc[:, df.columns != 'status'].values[:, 1:] # values use for array format
labels = df.loc[:, 'status'].values
```

In [6]:
```
df['status'].value_counts()
```

Out[6]:
```
1    147
0     48
Name: status, dtype: int64
```

In [8]:
```
scaler = MinMaxScaler((-1, 1))

X = scaler.fit_transform(features)
y = labels
```

XGBoost Algortihm

XGBoost is an implementation of gradient boosted decision trees. Although, it was designed for speed and performance.

Basically, it is a type of software library. That you can download and install on your machine. Then have to access it from a variety of interfaces.

```
model = XGBClassifier(use_label_encoder = False)
model.fit(x_train, y_train)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', use_label_encoder=False,
              validate_parameters=1, verbosity=None)
```

Model Training

```
y_prediction = model.predict(x_test)

print("Accuracy Score is", accuracy_score(y_test, y_prediction) * 100)
```

Accuracy :
96.66%

Model Prediction

Summary

In this Python machine learning project, we learned to detect the presence of Parkinson's Disease in individuals using various factors. We used an XGBClassifier for this and made use of the sklearn library to prepare the dataset. This gives us an accuracy of **96.66%**, which is great considering the number of lines of code in this python project.

Thank you!