



Modeling & Evaluation Workshop

Break Through Tech New York @ Cornell Tech

Today's Agenda



I. Modeling

1. Preparing data
2. Different approaches
3. Choosing your model
4. Training basic models
5. Training advanced model
6. Evaluating performance

II. Modeling in Kaggle

1. Kaggle Introduction
2. Main functions/usages in Kaggle
3. Model selection in Kaggle
4. How to use Kaggle models in your code
5. Model performance illustrations (optional)

III. Practice



Preparing Data

Holdout method: the dataset is randomly divided into two or three subsets.

Train Data:

Subset of the dataset used to build predictive models.

(Validation Data):

Subset of the dataset used to assess the performance of the model built in the training phase.

Test Data:

Set unseen by the model to avoid overfitting.

```
# Now we split the dataset in train and test part
# here the train set is 75% and test set is 25%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=2)
```

Another method is Cross-validation.

Different approaches



Supervised learning

- Labeled training examples
- Model trained to make accurate predictions

Ex: House prices in NYC

Unsupervised learning

- Dataset without labels
- Goal: learn something about the data (Hidden clusters, outliers)

Ex: Netflix movie suggestion

Reinforcement learning

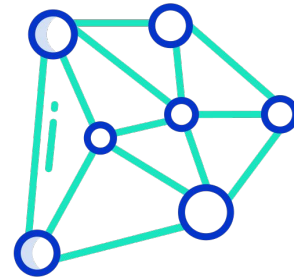
- Agent is interacting with the world over time
- Good behavior taught with rewards

Ex: Chess playing agent

Deep learning

- Can be supervised or unsupervised
- Loosely inspired by the brain

Ex: Virtual assistants like Siri





Choosing your model

1

Categorize your model:

- By input:
 - Labelled data (Supervised)
 - Unlabelled data (Unsupervised)
- By output:
 - Number (Regression)
 - Class (Classification)
 - Set of input groups (Clustering)

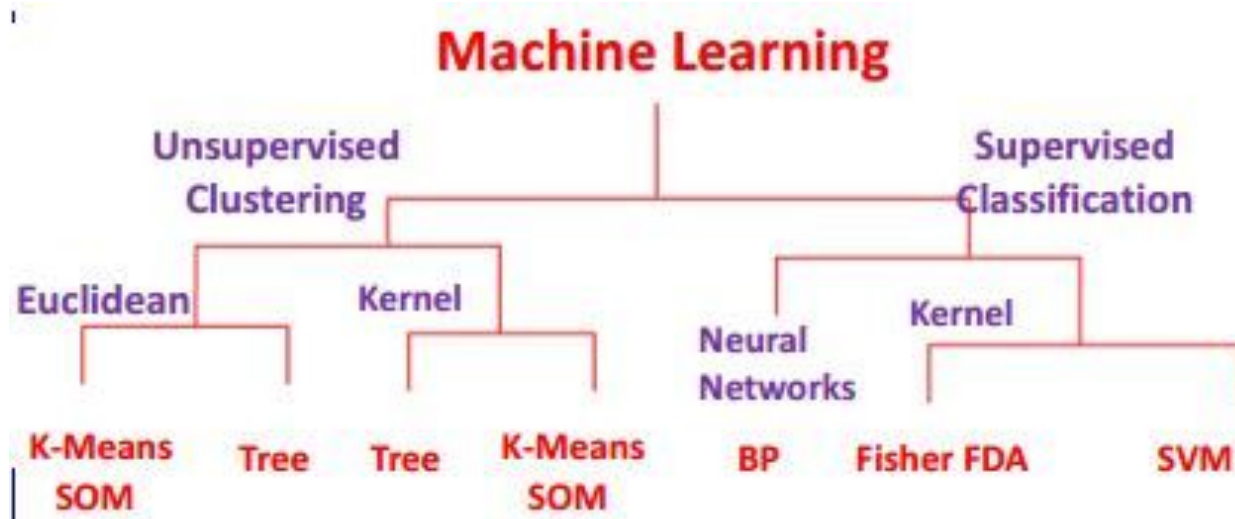
2

Select implementable models:

- Different feature engineering are required.
- Performance in regard to the size of the dataset.
- Complexity can grow with the size of the dataset.
- Deep learning training needs huge computational complexity.



Can we recognize different models?





More than an AI detector Preserve what's human.

We bring transparency to humans navigating a world filled with AI content. GPTZero is the gold standard in AI detection, trained to detect ChatGPT, GPT4, Bard, LLaMa, and other AI models.

Check out our products →

Was this text written by a **human** or **AI**?

Try detecting one of our sample texts:

ChatGPT

GPT4

Bard

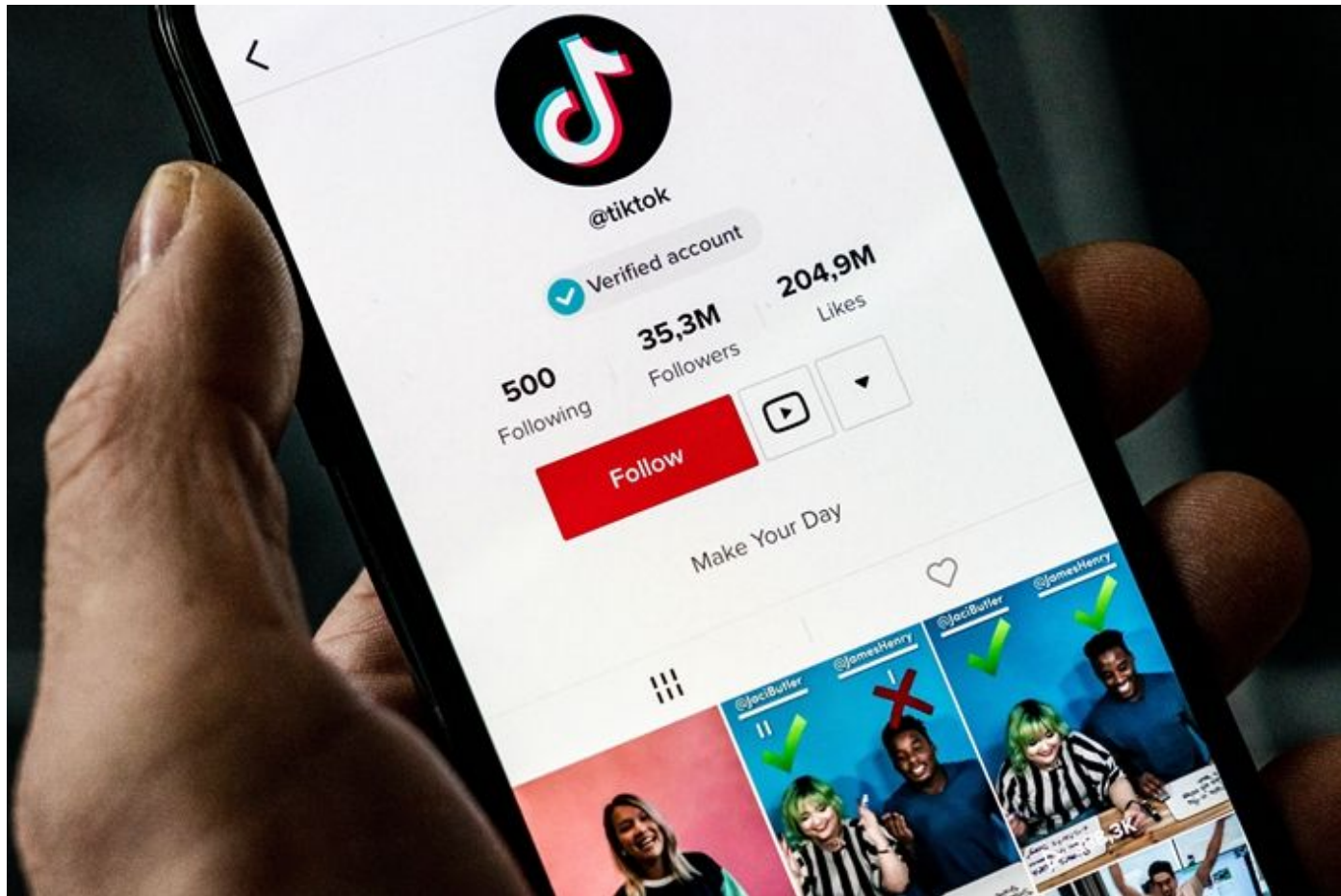
Human

AI + Human

Paste your text here ...



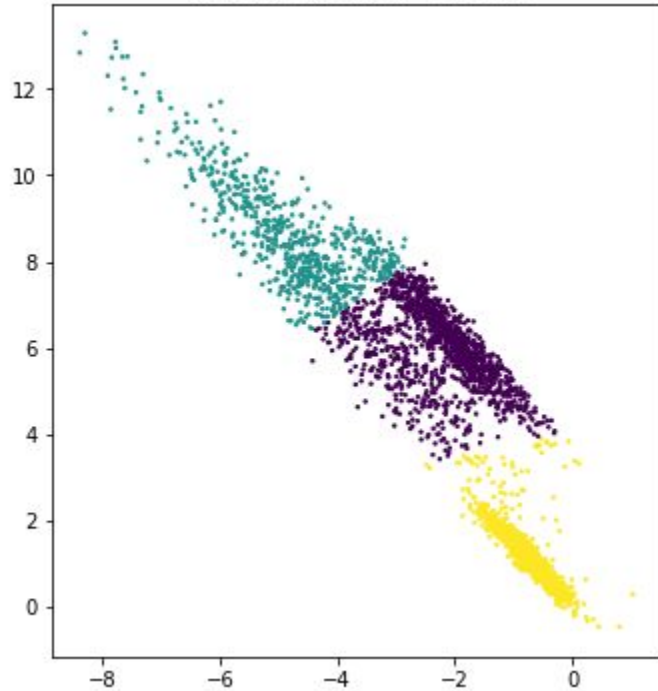
Amazon Go Store, Vox



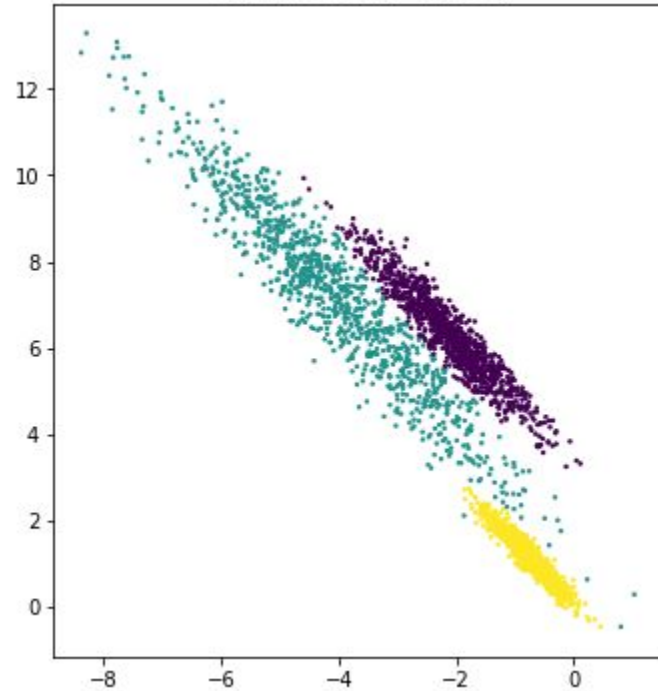
Different types of models: Clustering



K-Means (accuracy=0.8117)



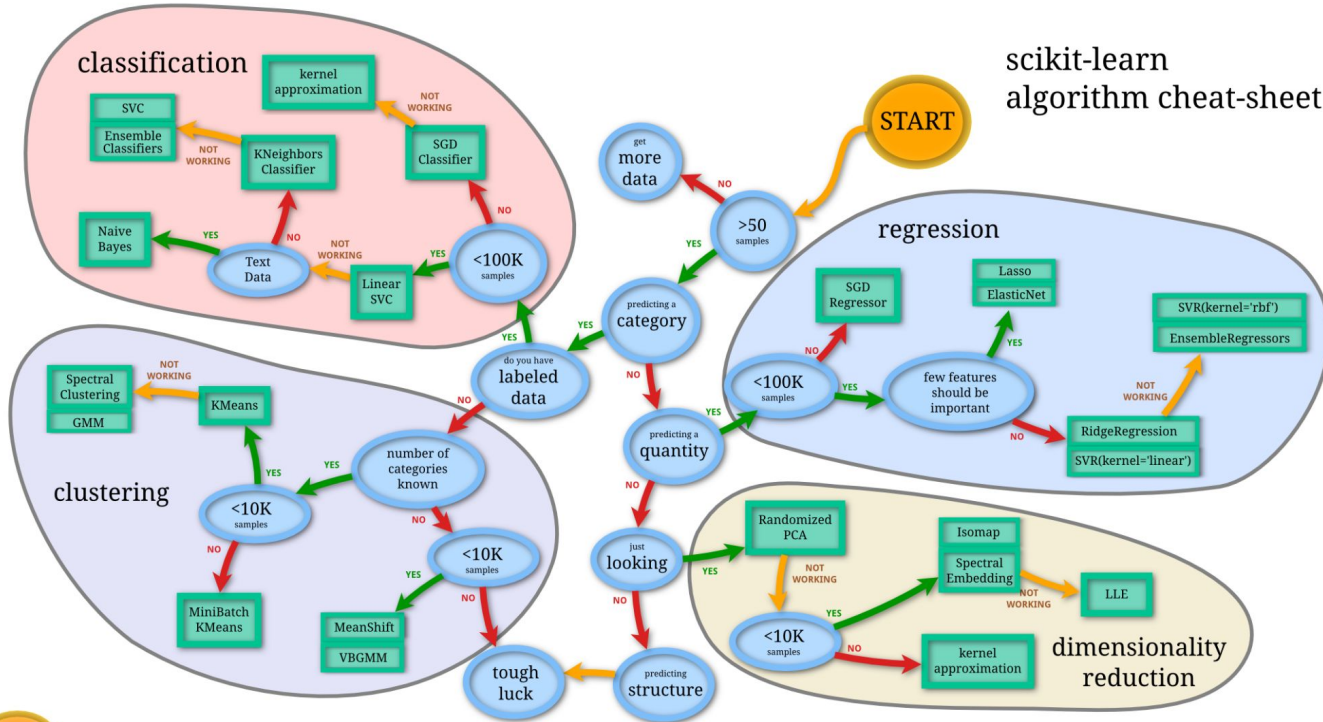
GMM (accuracy=0.9803)



Choosing your model



scikit-learn
algorithm cheat-sheet





Training basic models

Classification methods:

- Logistic Regression
- Naive Bayes
- Decision Tree

Regression methods:

- Linear Regression
- Polynomial Regression
- Support Vector Regression

Unsupervised learning:

- K-means
- PCA

Sklearn: Python library with built-in models

```
#Importing the Decision Tree from scikit-learn library
from sklearn.tree import DecisionTreeClassifier
# Training the model is as simple as this
# Use the function imported above and apply fit() on it
DT= DecisionTreeClassifier()
DT.fit(X_train,y_train)
```

Training advanced models



Complex compositions:

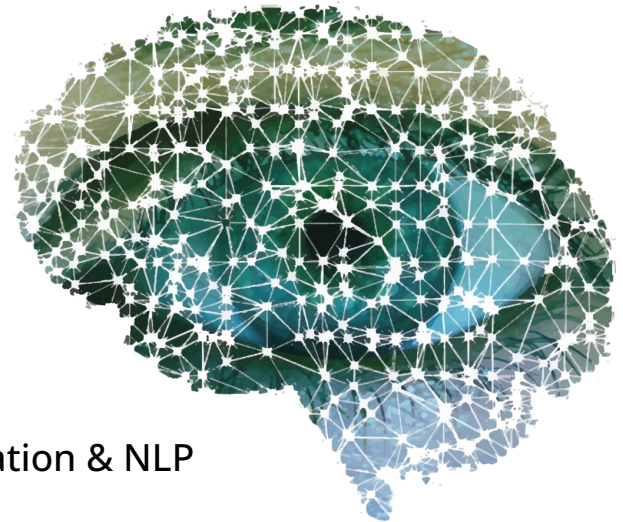
- Decision trees are easy to interpret.
- But, most often used in compositions such as Random Forest or Gradient boosting.

Deep learning algorithms:

- Have artificial neural networks structures
- Find patterns and represent data on their own
- Don't require much human intervention
- Adapt faster to the data at hand

Examples:

- Perceptron for binary classification
 - Multi-layer perceptron
- Convolutional Neural Networks: used in image classification & NLP



Evaluating performance



1 Implement all selected models.

2 Compare performances with evaluation metrics.
The choice of the metric depends on the task.

Classification Metrics:

- Accuracy
- Confusion matrix
- Logarithm loss

Regression Metrics:

- Root Mean Squared Error
- Mean Absolute Error




3 Select the best model

4 Optimize the chosen model's hyperparameters

The Confusion Matrix

		ACTUAL	
		POSITIVE	NEGATIVE
PREDICTED	Positive	TRUE POSITIVE	FALSE POSITIVE Type I Error
	Negative	FALSE NEGATIVE Type II Error	TRUE NEGATIVE



Evaluating performance



1 **F1** Score



2 **R²** Score

Evaluating performance



Computation of evaluation metrics with *Python*

```
# We use the predict() on the model to predict the output
pred=DT.predict(X_test)

# for classification we use accuracy and F1 score
print(accuracy_score(y_test,pred))
print(f1_score(y_test,pred))

# for regression we use R2 score and MAE(mean absolute error)
# all other steps will be same as classification as shown above
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
print(mean_absolute_error(y_test,pred))
```



Part II: Modeling in Kaggle





Kaggle Intro

What is Kaggle?

Kaggle is a community and data science platform that provides:

- **Tools** to build, train and deploy ML models on **open source (OS) code** and technologies.
- A place where a broad community get support and contribute to open source projects.

Main functions/usages in Kaggle



● Code

☰ kaggle

+ Create

🏠 Home

🏆 Competitions

📁 Datasets

🤖 Models

◀> Code

🗨️ Discussions

📖 Learn

∨ More

Code

Explore and run machine learning code with Kaggle Notebooks.
Find help in the [Documentation](#).

+ New Notebook



🔍 Search public notebooks

☰ Filters

All notebooks

Recently Viewed

Python

R

Beginner

NLP

Random Forest

GPU

TPU

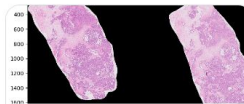
Competition notebook

Scheduled notebook

Competition Metric

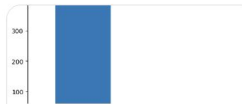
📈 Trending

See all (425)



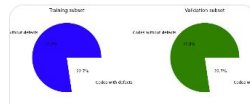
EDA + Baseline

Updated an hour ago
UBC Ovarian Cancer Subtype
Classification and Outlier Detection (UBC-OCEAN)



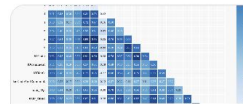
Know your data!! - EDA

Updated 11 hours ago
UBC Ovarian Cancer Subtype
Classification and Outlier Detection (UBC-OCEAN)



Software defect classifier

Updated 5 hours ago
Binary Classification with a Software
Defects Dataset



[PG S3 E23] EDA + Modeling (Ensemble+NN)

Updated 5 hours ago
Software Defect Prediction+1

Main functions/usages in Kaggle



- **Competitions** – A place for self contained ML demo apps.

The screenshot displays the Kaggle website interface. On the left is a navigation sidebar with options: Create, Home, Competitions (highlighted), Datasets, Models, Code, Discussions, Learn, and More. At the bottom of the sidebar is a 'View Active Events' link. The main content area features a search bar, 'Sign In' and 'Register' buttons, and a 'Competitions' header. Below the header is a 'Host a Competition' button and a paragraph of introductory text. A secondary search bar and 'Filters' button are also present. A row of category buttons includes 'All Competitions', 'Featured', 'Getting Started', 'Research', 'Community', and 'Playground'. The 'Get Started' section is expanded, showing three featured competitions: 'Titanic - Machine Learning from Disaster' (14833 Teams), 'House Prices - Advanced Regression Techniques' (3964 Teams), and 'Spaceship Titanic' (2062 Teams). The Kaggle logo is visible in the bottom right corner.

Model selection in Kaggle



Search

Sign In

Register

Overview Data **Code** Models Discussion Leaderboard Rules

New Notebook

Unpinned notebooks



Comprehensive data exploration with Python

Updated 1y ago

1931 comments · House Prices - Advanced Regression Techniques

13756

Gold



Stacked Regressions : Top 4% on LeaderBoard

Updated 6y ago

1090 comments · House Prices - Advanced Regression Techniques

7014

Gold



Regularized Linear Models

Updated 9mo ago

Score: 0.12096 · 341 comments · House Prices - Advanced Regression Techniques

1793

Gold



Submitting From A Kernel

Updated 6y ago

497 comments · House Prices - Advanced Regression Techniques

1665

Gold



House prices: Lasso, XGBoost, and a detailed EDA

Updated 5y ago

260 comments · House Prices - Advanced Regression Techniques

1572

Gold



Handling Missing Values

Updated 5y ago

443 comments · House Prices - Advanced Regression Techniques +2

1378

Gold

Hotness

Most Votes

Most Comments

Recently Created

Recently Run

Public Score

Relevance



How to use Kaggle models in your code

Bringing it all together!

- 1. Authentication (to save/submit your notebooks!)**
- 2. Data Preparation**
- 3. Fine-tuning the model for your requirements**
- 4. Training the customised model**
- 5. Model outputs**



Model performance illustrations

Building and sharing your model demos using Kaggle

[Try this out in notebook!](#)

A screenshot of a Kaggle notebook interface. The top bar shows the notebook title "BTTAI Workshop Modelling.ipynb" and a star icon. Below the title is a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", followed by the text "All changes saved". The main content area is dark grey and contains a table of contents for the notebook "Real or Not? NLP with Disaster Tweets". The table of contents lists several topics under the heading "NLP:".

BTTAI Workshop Modelling.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Real or Not? NLP with Disaster Tweets

NLP:

- EDA (with WordCloud)
- Bag of Words
- TF IDF
- GloVe
- BERT with TFHub and with Submission
- PCA visualization for the main models
- Showing Confusion Matrices for BERT, Simpletransformers with DistilBERT and GloVe



Tips

- 1. Data and model security**
- 2. Downsides using open source code/tools**
- 3. Complexity and adaptability of Kaggle models**



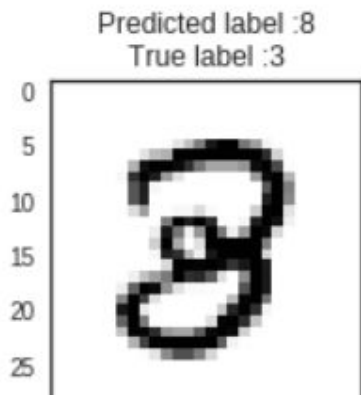
Part III: Practice

kaggle

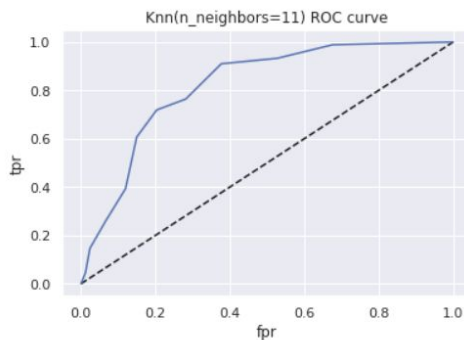
Your Time to Try



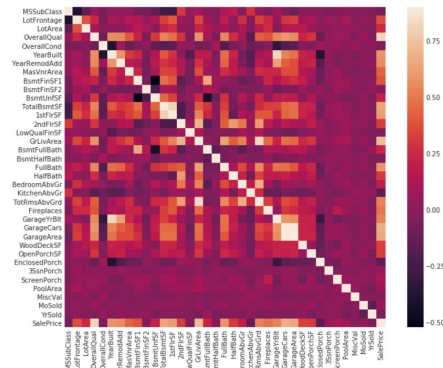
CNN



KNN



Regression





Your Time to Try

Decision Tree

Deep Learning

RNN

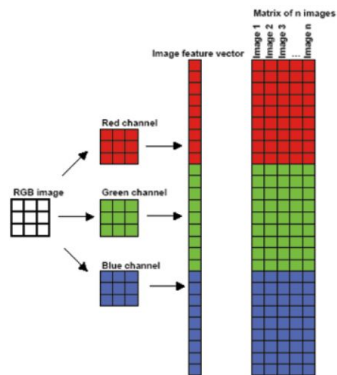
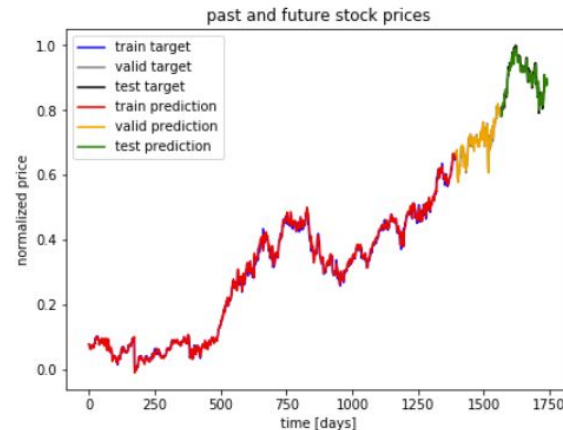
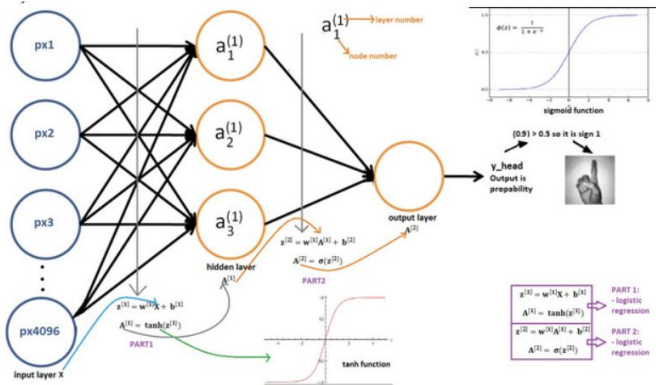


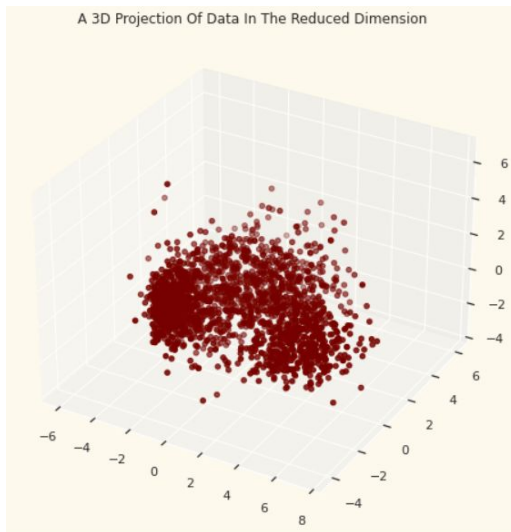
Figure 1
Image unfolding and the preparation of an image matrix



Your Time to Try



Clustering



Prediction

