Fundamentals of Machine Learning and Analyzing Data with Python

A Practical Approach

## ความรรรู้พพพื้นฐานทางดรานการเรรียนรรรู้เครรื่พองจจักร และการววิเคราะหห์ขรอมรู้ลดรวยโปรแกรมภาษาไพธอน ดด้วยววิธฝฝึ กปฏวิบบัตวิ

Olarik Surinta

**Fundamentals of Machine Learning and Analyzing Data with Python**

### A Practical Approach

**ความรรรู้พพพื้นฐานทางดรานการเรรียนรรรู้เครรื่พองจจักร และการววิเคราะหห์ขรอมรู้ลดรวยโปรแกรมภาษาไพธอน** ดด้วยววิธฝฝึ กปฏวิบบัตวิ

**Fundamentals of Machine Learning and Analyzing Data with Python**

### A Practical Approach

**ความรรรู้พพพื้นฐานทางดรานการเรรียนรรรู้เครรื่พองจจักร และการววิเคราะหห์ขรอมรู้ลดรวยโปรแกรมภาษาไพธอน** ดด้วยววิธฝฝึ กปฏวิบบัตวิ

## Olarik Surinta

Department of Information Technology Faculty of Informatics, Mahasarakham University

Maha Sarakham, Thailand

**คคานคา**

เทคโนโลยธีสารสนเทศสมบัยใหมม่ททาใหด้โลกเกวิดการปฏวิวบัตวิททาใหด้มนนุษยย์พบัฒนานวบัตกรรม ไมม่วม่าจะ เปป็ นสวิสิ่งประดวิษฐย์ทธีสิ่สามารถจบับตด้องไดด้ ไปจนซอฟตย์แวรย์ทธีสิ่สามารถใชด้กบันอยม่างแพรม่หลายผม่านระบบเครรือ ขม่ายอวินเทอรย์เนป็ต ททาใหด้เกวิดระบบทธีสิ่ททางานผม่านโลกออนไลนย์มธีอบัตราการเจรวิญเตวิบโตสสูงขขึ้ฝึนอยม่างกด้าว กระโดดในทนุกปธี อธีกทบัขึ้งยบังเกวิดขด้อมสูลมหาศาลทธีสิ่อยม่สูบนโลกออนไลนย์ ดบังนบัขึ้น จฝึงททาใหด้สามารถนทาขด้อมสูล เหลม่านบัขึ้นมาววิเคราะหย์ และประมวลผล (Data Analytics) เพสิ่รือนทาแบบจทาลองไปใชด้เปป็ นตบัวชม่วยในการ ววิเคราะหย์และพบัฒนาธนุรกวิจ ททาใหด้สามารถมธีสารสนเทศทธีสิ่จะใชด้สทาหรบับแขม่งขบันกบับคม่สูแขม่งทางธนุรกวิจ การ ววิเคราะหย์ขด้อมสูลนบัขึ้นตด้องอาศบัยความรด้สูทางดด้านตม่าง ๆ มากมาย เชม่น เหมรืองขด้อมสูล (Data Mining) การ เรธียนรด้สูเครสิ่รืองจบักร (Machine Learning) และววิทยาการขด้อมสูล (Data Science) ซสิ่ฝึงเปป็ นแนวโนด้ ม (Trend) ใหมม่ของโลกในยนุคปบั จจนุบบัน ซสิ่ฝึงความรด้สูเหลม่านธีขึ้มธีสม่วนชม่วยใหด้สามารถเลรือกใชด้อบัลกอรวิธสิ่ฝึมไดด้ ถสูกตด้อง ตรงกบับปบั ญหา และตอบโจทยย์ทางธนุรกวิจมากทธีสิ่สนุด ในหนบังสรือเลม่มนธีขึ้ไดด้นทาเสนอเกธีสิ่ยวกบับการใชด้ โปรแกรมภาษาไพธอน (Python) ซสิ่ฝึงเปป็ นภาษาทธีสิ่นวิยมใชด้งานกบันในกลม่นุมนบักววิเคราะหย์ขด้อมสูล (Data Analyst) ซสิ่ฝึงมธีไลบรารธีสิ่ (Library) และเครสิ่รืองมรือ (Tool) ใหด้เลรือกใชด้มากมาย

**สารบจัญ**

[บทททท **1 Jupyter Notebook** **1**](#_bookmark0)

[ววธธการตวดตต งต โปรแกรม Jupyter Notebook 1](#_bookmark1)

[ววธธการเรธยกใชตโปรแกรม Jupyter Notebook 2](#_bookmark2)

[ไฟลลนามสกกล (File Extension) ของโปรแกรม Jupyter 3](#_bookmark5)

[บทททท **2** การโหลดขขอมมล **(Loading Data)** **5**](#_bookmark7)

[Iris Dataset 5](#_bookmark8)

[ลต กษณะของชกดขตอมมล Iris 7](#_bookmark10)

[MNIST Dataset 9](#_bookmark12)

[การ Visualization เพพพอดมรมปภาพตต วเลข 13](#_bookmark14)

[ขตอมมลทพธนนาไปใชตใน scikit-learn 14](#_bookmark15)

[โหลดชกดขตอมมล MNIST ดตวย Scipy 15](#_bookmark16)

[การ Visualization ชกดขตอมมล MNIST 18](#_bookmark17)

[บทททท **3** ขขอมมลชชดเรทยนรขมและขขอมมลชชดทดสอบ **(Training and Test Data)** **20**](#_bookmark19)

[การกนาหนดอต ตราทพธใชตสาน หรต บแบพงขตอมมล 21](#_bookmark20)

[การแสดงขตอมมลในรมปแบบของกราฟ 21](#_bookmark21)

[บทททท **4** ไลบรารทท **Matplotlib (Matplotlib Library)** **25**](#_bookmark22)

[การบต นททกกราฟ (Figure) 27](#_bookmark23)

[การแสดงรมปภาพ (Image Show) 28](#_bookmark24)

[ววธธการแสดงรมปภาพโดยใชตอวนเทอรลเฟซ (Interface) ทพธตพางกต น 28](#_bookmark25)

[การเพพวมเสตนตาราง (Grid) ในการพลอต 30](#_bookmark26)

[การกนาหนดววธธการพลอต: Line Color และ Style 34](#_bookmark27)

[การกนาหนด Axes Limit 37](#_bookmark28)

[การพลอต Label 39](#_bookmark29)

[การพลอต Legend 39](#_bookmark30)

[การกนาหนดการพลอต Legend 40](#_bookmark31)

[การพลอตแบบ Scatter 45](#_bookmark32)

[การตกแตพง Scatter 49](#_bookmark33)

[การพลอต Histograms, Binnings และ Density 52](#_bookmark34)

[การคนานวณคพา Histogram 54](#_bookmark35)

[การพลอตชกดขตอมมล MNSIT ทพธอยมพใน scikit-learn 55](#_bookmark36)

[การพลอตชกดขตอมมล MNIST ในรมปแบบ 2 มวตว โดยใชตววธธ IsoMap 55](#_bookmark37)

[บทททท **5** ไลบรารทท **Seaborn (Seaborn Library)** **57**](#_bookmark38)

[การแสดงชกดขตอมมล Iris แบบ Visualization 58](#_bookmark39)

[การทนางานรพวมกต นระหวพาง seaborn และ scikit-learn 59](#_bookmark40)

[บทททท **6** การววเคราะหหการถดถอยเชวงเสขน **(Linear Regression)** **62**](#_bookmark41)

[การจนาลองชกดขตอมมลเพพพอใชตในการคนานวณ Linear Regression 63](#_bookmark42)

[การเรธยกใชตโมดมล LinearRegression 63](#_bookmark43)

[จต ดการขตอมมลเพพพอใชตในการคนานวณ 64](#_bookmark44)

[การเรธยนรมตเพพพอสรตางโมเดล (Train the model) 64](#_bookmark45)

[การพยากรณลผลลต พธลจากขตอมมลใหมพ (Predict Labels for Unknown Data) 65](#_bookmark46)

[การพยากรณลขตอมมล Diabetes ดตวย Linear Regression 66](#_bookmark47)

[การพยากรณลขตอมมล Housing ดตวย Linear Regression 71](#_bookmark48)

[บทททท **7** ตตวจาจ แนกแบบไบนารทท **(**ฺ**Binary Classifier)** **79**](#_bookmark49)

[Stochastic Descent 81](#_bookmark50)

[สรตางโมเดล Stochastic Descent 82](#_bookmark51)

[การพยากรณลดตวยโมเดล Stochastic Descent 82](#_bookmark52)

[การวตดประสวทธวภาพ (Performance Measurement) 84](#_bookmark53)

[การประเมวนประสวทธวภาพของอต ลกอรวธทมดตวย Confusion Matrix 84](#_bookmark54)

[การนนาโมเดลไปทดสอบกต บขตอมมลชกดทดสอบ 87](#_bookmark55)

[บทททท **8** การคาจ นวณเพทพอนบขานใกลขทททสชด **k** ตาจ แหนทง **(**ฺ**K-Nearest Neighbors)** **89**](#_bookmark56)

[การสรตางโมเดลของ KNN 90](#_bookmark57)

[การพยากรณลโดยใชตโมเดลของ KNN 91](#_bookmark58)

[การทดสอบประสวทธวภาพของโมเดล KNN 92](#_bookmark59)

[การแสดงผลการทดลองดตวย Confusion Matrix 92](#_bookmark60)

[การใชตงานอต ลกอรวธทม KNN กต บขตอมมลโรคเบาหวาน (Diabetes Dataset) 95](#_bookmark61)

[สรตางโมเดล KNN ดตวยคพา n\_neighbor ทพธไดตจากการทดลอง 97](#_bookmark62)

[คนาสตพง Pandas Crosstab 98](#_bookmark63)

[แสดงผลลต พธลจากการทดลองดตวย Classification Report 98](#_bookmark64)

[การทดสอบคพาพารามวเตอรล (Hyperparameter Tuning) ดตวยววธธ Grid Search 99](#_bookmark65)

[การจต ดหมวดหมมพชกดขตอมมล MNIST ดตวยอต ลกอรวธทม KNN 100](#_bookmark66)

[KNN Classifier 101](#_bookmark67)

[บทททท **9** การจตดหมวดหมมทขขอมมลดขวย **Naive Bayes (**ฺ**Naive Bayes Classification)** **105**](#_bookmark68)

[การเตรธยมขตอมมลเพพพอใชตในการเรธยนรมต 107](#_bookmark69)

[สรตางโมเดล Naive Bayes 108](#_bookmark70)

[พยากรณลขตอมมลดตวยโมเดล Naive Bayes และแสดงประสวทธวภาพของโมเดล 108](#_bookmark71)

[บทททท **10** การววเคราะหหองคหประกอบหลตก **(**ฺ**Principal Component Analysis)** **111**](#_bookmark72)

[การสรตางโมเดล PCA 111](#_bookmark73)

[การเพพวมขตอมมลจากตต วแปรเขตาไปเกกบเพพวมใน DataFrame 112](#_bookmark76)

[สรตางโมเดล Naive Bayes ดตวยคกณลต กษณะพวเศษทพธไดตจาก PCA 115](#_bookmark77)

[การพยากรณลคกณลต กษณะพวเศษทพธไดตจาก PCA ดตวยอต ลกอรวธทม Naive Bayes และประสวทธวภาพจากการพยากรณล](#_bookmark78)

[. 117](#_bookmark78)

[บทททท **11** การจตดกลชทมดขวยอตลกอรวธธม **K-Means (**ฺ**K-Means Clustering)** **119**](#_bookmark79)

[จนาลองขตอมมลเพพพอใชตในอต ลกอรวธทม K-Means 120](#_bookmark81)

[สรตางโมเดลของอต ลกอรวธทม K-Means 121](#_bookmark82)

[การพยากรณลดตวยอต ลกอรวธทม K-Means 121](#_bookmark83)

[สรตางขตอมมลใหมพเพพพอทดสอบการแบพงกลกพมดตวยอต ลกอรวธทม K-Means 123](#_bookmark84)

[พยากรณลขตอมมลทพธสรตางขตทนมาใหมพดตวยอต ลกอรวธทม K-Means 124](#_bookmark85)

[บทททท **12** การรขมจาจ ใบหนขา **(**ฺ**Face Recognition)** **127**](#_bookmark86)

[สรตางโมเดลของอต ลกอรวธทม SVM 128](#_bookmark87)

[การวตดประสวทธวภาพของการรมตจนาใบหนตา (Classification Report) 130](#_bookmark89)

[การแสดงความถมกตตองของการพยากรณลดตวย Confusion Matrix 131](#_bookmark90)

[อต ตราความถมกตตอง (Accuracy Result) ของการพยากรณลรมปภาพใบหนตา 131](#_bookmark91)

[บทททท **13** การรขมจาจ ตตวอตกษร **(**ฺ**Character Recognition)** **133**](#_bookmark92)

[สรตางโมเดลของอต ลกอรวธทม MLP 136](#_bookmark94)

[การวตดประสวทธวภาพของการเรธยนรมต 136](#_bookmark95)

[การพยากรณลและวตดประสวทธวภาพของการมตจนา 137](#_bookmark96)

[การ Visualization รมปภาพตต วเลข และแสดงผลการพยากรณล 138](#_bookmark97)

**สามารถดาวนห์โหลดตจัวอยย่างโปรแกรม และบทเรรียนนรีพื้ในรรู้ปแบบของ Jupyter Notebook ไดรจากเวว็บไซตห์ github **

[https://github.com/mrolarik/basic-machine-learning-using-scikit-](https://github.com/mrolarik/basic-machine-learning-using-scikit-learn) [learn](https://github.com/mrolarik/basic-machine-learning-using-scikit-learn)

**สารบจัญภาพประกอบ**

ภาพประกอบทพธ 1: ลต กษณะการทนางานของโปรแกรม Jupyter Notebook 1

ภาพประกอบทพธ 2: ตต วอยพางการเรธยกใชต jupyter notebook ผพานหนตาจอ Terminal 2

ภาพประกอบทพธ 3: ตต วอยพางโปรแกรม Jupyter 3

ภาพประกอบทพธ 4: ตต วอยพางโครงสรตางการเกกบขตอมมลของ Jupyter โดยจต ดเกกบในรมปแบบ JSON 3

ภาพประกอบทพธ 5: ชกดขตอมมลดอกไมต Iris ประกอบดตวยดอกไมต 3 สายพต นธกล ไดตแกพ Versicolor, Setosa และ

Virginica 5

ภาพประกอบทพธ 6: ตต วอยพางการเกกบขตอมมลของชกดขตอมมล iris 8

ภาพประกอบทพธ 7: ตต วอยพางของขตอมมลชกด MNSIT 10

ภาพประกอบทพธ 8: การ Visualization ชกดขตอมมล MNIST พรตอมทต งต แสดง label ของแตพละตต วเลข 19

ภาพประกอบทพธ 9: แสดงสมการถดถอยเชวงเสตน Linear Regression 62

ภาพประกอบทพธ 10: เสตน Hyperplane ทพธไดตจากการคนานวณดตวยววธธ Linear Regression 66

ภาพประกอบทพธ 11: แสดงการพลอตคพา target จรวง (Actual Target) และคพา target ทพธไดตจากการพยากรณล

(Predicted Target) 70

ภาพประกอบทพธ 12: ตต วอยพางขตอมมลทพธจต ดเกกบในรมปแบบของ pandas DataFrame 73

ภาพประกอบทพธ 13: แสดงเสตน Hyperplane ทพธใชตแบพงขตอมมลออกเปกนสองสพวน 79

ภาพประกอบทพธ 14: แสดงลต กษณะการทนางานของอต ลกอรวธทม KNN 89

ภาพประกอบทพธ 15: กราฟแสดงประสวทธวภาพของอต ลกอรวธทม KNN เมพพอเปลพธยนคพาพารามวเตอรล n\_neighbors 97

ภาพประกอบทพธ 16: ตต วอยพางการจต ดกลกพมดตวยววธธ K-Means โดยแสดงใหตดมทธละขต นต ตอน 119

ภาพประกอบทพธ 17: ตต วอยพางใบหนตาจากชกดขตอมมล lfw ทพธนนามาใชตในการรมตจนาใบหนตา 128

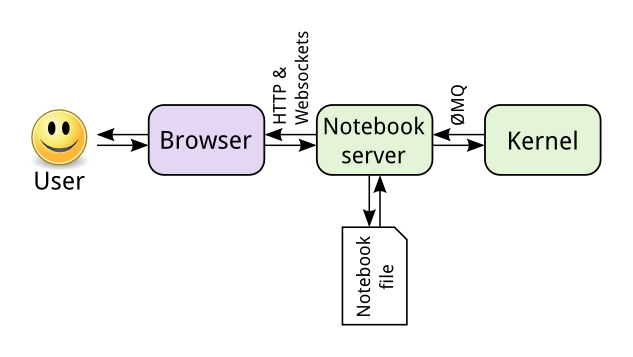
ภาพประกอบทพธ 18: ผลลต พธลทพธไดตจากการพยากรณลใบหนตาดตวยววธธ PCA และ SVM 130

ภาพประกอบทพธ 19: ตต วอยพางโครงสรตางของ Multi-Layer Perceptron (MLP) 133

ภาพประกอบทพธ 20: เปรธยบเทธยบระหวพางคพาทพธแทตจรวง และคพาทพธไดตจากการพยากรณล 139

บทททท **1 Jupyter Notebook**

Jupyter หรพอ Jupyter Notebook เปกนเคอรลนอล (Kernel) ของโปรแกรมภาษา Python ทพธ ทนาใหตสามารถคนานวณและตวดตพอสพพอสารกต บสพวนตวดตพอกต บผมตใชตงาน (Frontend) ไดตสะดวก และ เปกนการทนางานผพานเวกบบราวเซอรล (Browser)



ภาพประกอบททท **1:** ลต กษณะการทนางานของโปรแกรม Jupyter Notebook

ววธทการตวดตตขงโปรแกรม **Jupyter Notebook**

คนาสตพง

การตวดตต งต โปรแกรม Jupyter Notebook สามารถตวดตต งต ผพานเทอรลมวนต ล (Terminal) โดยใชต

**$ pip install jupyter**

2 ความรมตพตพนฐานทางดตานการเรธยนรมตเครพพองจตกร และการววเคราะหลขตอมมลดตวยโปรแกรมภาษาไพธอน

ตต วอยพางขตางตตนเปกนการตวดตต งต สนาหรต บ Python เวอรลชต น 2 แตพสนาหรต บ Python เวอรลชต น 3 จะ ตตองใชตคนาสตพง ดต งนตธ

**$ pip3 install jupyter**

ววธทการเรทยกใชขโปรแกรม **Jupyter Notebook**

เมพพอตวดตต งต โปรแกรม Jupyter เสรกจเรธยบรตอย สามารถเรธยกใชตโปรแกรม Jupyter โดยพวมพล คนาสตพงผพาน Terminal ดต งนตธ

**$ jupyter notebook**

เมพพอเรธยกใชตคนาสตพง $ jupyter notebook จากนต ตนโปรแกรมจะทนาการจนาลองเปกนเซวรลฟเวอรล

(Server) เพพพอใหตสามารถทนางานผพานเวกบบราวเซอรล โดยจะเปว ดเวกบบราวเซอรล ณ ตนาแหนพงทพธอยมพ ปต จจกบต น ดต งนต ตน หากตตองการทพธจะใหตโปรแกรม Jupyter ทนางานในโฟลเดอรลทพธตตองการ ใหตเปว ด

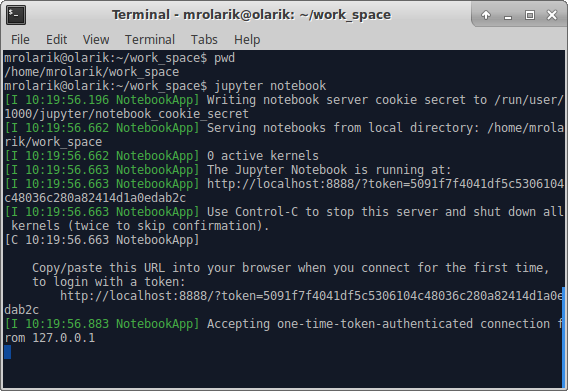
Terminal และไปยต งตนาแหนพงทพธตตองการ จากนต ตนจทงจะเรธยกใชตคนาสตพง $ jupyter notebook (ดต งภาพ

ประกอบทพธ [2](#_bookmark3)) เชพน

**$ cd /home/user/work\_space**

**$ jupyter notebook**

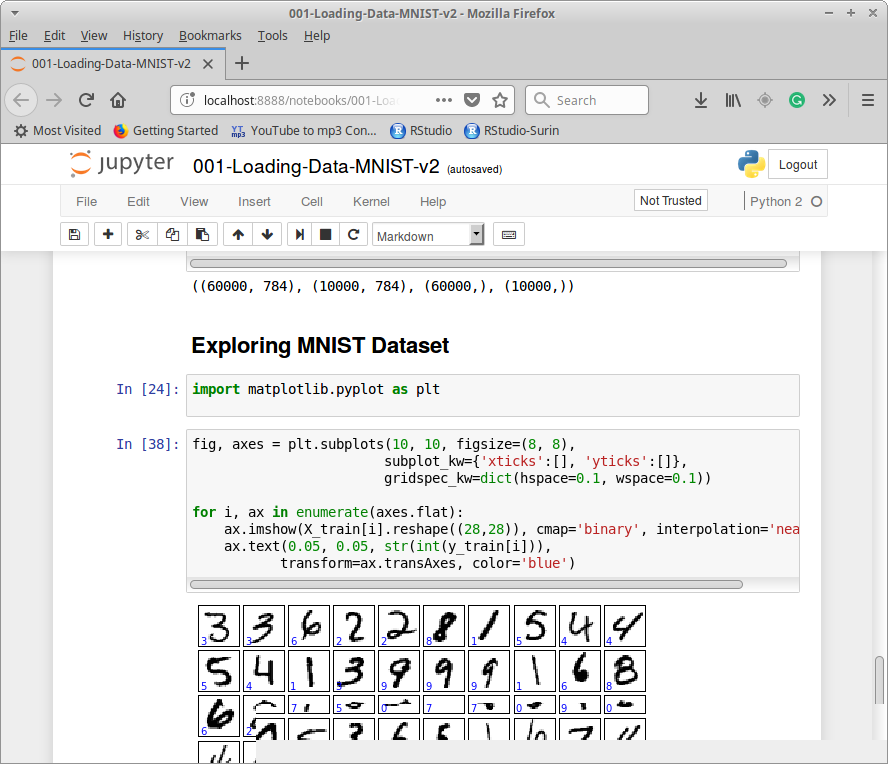
จากตต วอยพาง เมพพอเรธยกใชตโปรแกรม jupyter โปรแกรมจะเปว ดเวกบบราวเซอรลและทนางาน ณ ตนาแหนพง /home/user/work\_space โดย user เปกนโฟลเดอรลทพธเปลพธยนไปตามชพพอผมตใชตงาน โดย jupyter notebook จะจนาลองตต วเปกนเซวรลฟเวอรลเพพพอใหตสามารถเปว ดและทนางานผพานเวกบบราวเซอรล (ดต งภาพประกอบทพธ [3](#_bookmark4))



ภาพประกอบททท **2:** ตต วอยพางการเรธยกใชต jupyter notebook

ผพานหนตาจอ Terminal

Fundamentals of Machine Learning and Analyzing Data with Python 3



ภาพประกอบททท **3:** ตต วอยพางโปรแกรม Jupyter

ไฟลหนามสกชล **(File Extension)** ของโปรแกรม **Jupyter**

โปรแกรม Jupyter จะมธนามสกกล (Extension) เปกน .ipynb ซพทงจะแตกตพางจากโปรแกรม ภาษา Python ทตพวไปทพธจะใชตนามสกกล .py โดยโปรแกรม Jupyter จะเกกบขตอมมลทต งต หมดในรมปแบบ ของ JSON แสดงดต งภาพประกอบทพธ [4](#_bookmark6)

{

"cells": [

{

"attachments": {}, "cell\_type": "markdown", "metadata": {}, "source": [

"# Jupyter\n", "\n",

"Jupyter Notebook เปกน Kernel ของโปรแกรม Python ทพธทนาใหตสามารถคนานวณ และตวดตพอสพพอสาร

กต บ frontend interface โดยเปกนการทนางานผพานเวกบบราวเซอรล\n", "\n",

"![alt text] [(http://jupyter.readthedocs.io/en/latest/\_images/notebook\_components.png](http://jupyter.readthedocs.io/en/latest/_images/notebook_components.png) \ "Kernel\")"

]

},

ภาพประกอบททท **4:** ตต วอยพางโครงสรตางการเกกบขตอมมลของ Jupyter โดยจต ดเกกบในรมปแบบ JSON

4 ความรมตพตพนฐานทางดตานการเรธยนรมตเครพพองจตกร และการววเคราะหลขตอมมลดตวยโปรแกรมภาษาไพธอน

บทททท **2** การโหลดขขอมมล **(Loading Data)**

ในบทนตธ นนาเสนอชกดขตอมมล (Dataset) ทพธใชตกต นอยพางแพรพหลายในงานดตานการเรธยนรมต เครพพองจต กร (Machine Learning) โดยขตอมมลดต งกลพาว ประกอบดตวย Iris และ MNSIT dataset ซพทง ไดตนนาเสนอลต กษณะของขตอมมล และววธธการเรธยกใชตขตอมมล

## Iris Dataset

ชกดขตอมมลดอกไมต Iris (Iris Dataset) เปกนชกดขตอมมลพตพนฐานทพธใชตอยพางแพรพหลายทต งต ทางดตาน การเรธยนรมตเครพพองจต กร และทางสถวตว (Statistics) หากตวดตต งต โปรแกรม scikit-learn จะสามารถเรธยก ใชตชกดขตอมมล Iris ไดตทต นทธ ตต วอยพางรมปภาพดอกไมต Iris แสดงดต งภาพประกอบทพธ [5](#_bookmark9)



ภาพประกอบททท **5:** ชกดขตอมมลดอกไมต Iris ประกอบดตวยดอกไมต 3 สายพต นธกล ไดตแกพ Versicolor, Setosa และ

Virginica

หากตวดตต งต โปรแกรม scikit-learn สามารถเรธยกใชตชกดขตอมมล Iris ซพทงเปกนชกดขตอมมลดอกไมต ทนาไดตดต งตพอไปนตธ

ววธธทพธ 1

**from sklearn import datasets iris\_dataset = datasets.load\_iris()**

ววธธทพธ 2

**from sklearn.datasets import load\_iris iris\_dataset = load\_iris()**

จากตต วอยพางขตางตตนฟต งกลชต น (Function) load\_iris() เปกนฟต งกลชต นทพธอยมพในโมดมล (Module) สามารถเรธยกใชตไดตเมพพอตวดตต งต และอวมพอรลต (Import) scikit-learn มาใชตงาน เมพพอเรธยกใชตฟต งกลชต น load\_iris() จะคพอคพา (Return) ออกมาเปกน Bunch object ซพทงเปกนโครงสรตางแบบ dictionary ใน ภาษา Python ประกอบไปดตวย keys และ values การใชตงานสามารถทนาไดตดต งตพอไปนตธ

**print("Keys of iris\_dataset: \**

**\n{}".format(iris\_dataset.keys()))**

Keys of iris\_dataset:

['target\_names', 'data', 'target', 'DESCR', 'feature\_names']

จากตต วอยพาง DESCR คพอ key ทพธเกกบคพา (Value) โดยคพานต ตนคพอรายละเอธยดของชกดขตอมมล

iris แสดงดต งตพออยพางตพอไปนตธ

**print(iris\_dataset['DESCR'])**

Iris Plants Database

====================

Notes

Data Set Characteristics:

:Number of Instances: 150 (50 in each of three classes)

:Number of Attributes: 4 numeric, predictive attributes and the class

:Attribute Information:

* sepal length in cm
* sepal width in cm
* petal length in cm
* petal width in cm
* class:
  + Iris-Setosa
  + Iris-Versicolour
  + Iris-Virginica

:Summary Statistics:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ==============  ============== | ====  Min  ==== | ====  Max  ==== | =======  Mean  ======= | ===== SD  ===== | ====================  Class Correlation  ==================== |
| sepal length: | 4.3 | 7.9 | 5.84 | 0.83 | 0.7826 |
| sepal width: | 2.0 | 4.4 | 3.05 | 0.43 | -0.4194 |
| petal length: | 1.0 | 6.9 | 3.76 | 1.76 | 0.9490 (high!) |
| petal width: | 0.1 | 2.5 | 1.20 | 0.76 | 0.9565 (high!) |
| ============== | ==== | ==== | ======= | ===== | ==================== |

:Missing Attribute Values: None

:Class Distribution: 33.3% for each of 3 classes.

:Creator: R.A. Fisher

:Donor: Michael Marshall [(MARSHALL%PLU@io.arc.nasa.gov)](mailto:(MARSHALL%25PLU@io.arc.nasa.gov)

:Date: July, 1988

### ลตกษณะของชชดขขอมมล **Iris**

ชกดขตอมมล iris มธเอาทลพกท หรพอ target อยมพ 3 กลกพม ประกอบดตวย versicolor, setosa และ

virginica สามารถเรธยกดมโดยใชตคนาสตพง

**print("Target names: \**

**\n{}".format(iris\_dataset['target\_names']))**

Target names:

['setosa' 'versicolor' 'virginica']

หากตตองการแสดงรายชพพอของ feature หรพอ attribute ของชกดขตอมมล iris สามารถทนาไดตโดย

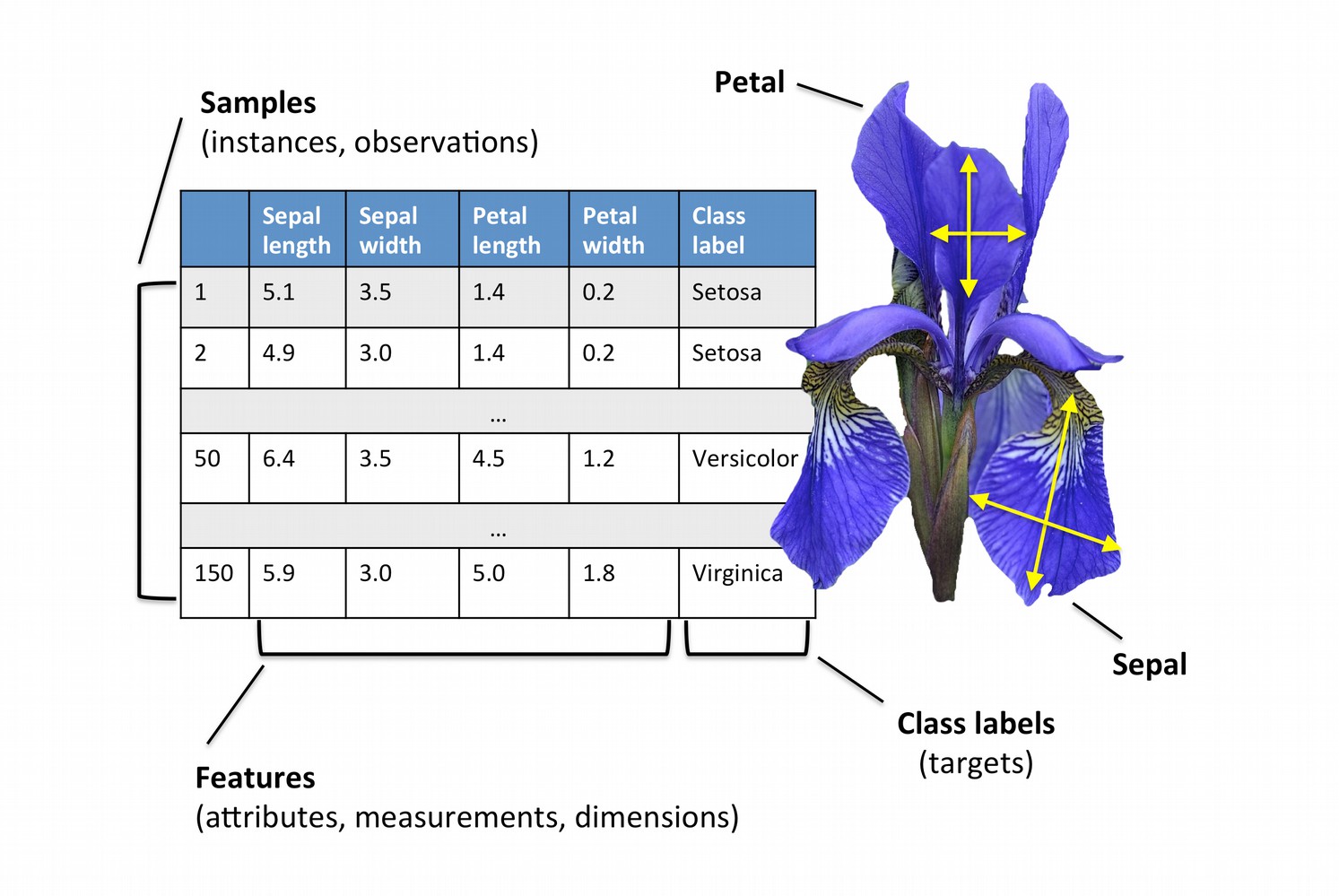
**print("Feature names: \**

**\n{}".format(iris\_dataset['feature\_names']))**

Feature names:

['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

ในแตพละ feature ทพธปรากฎในชกดขตอมมล iris คพอการวตดความกวตางและความยาวของกลธบใบทพธ เรธยกวพา Sepal และ Petal โดยมธหนพวยเปกนเซกนตวเมตร (cm.) ตต วอยพางของการเกกบขตอมมลแสดงดต ง ภาพประกอบ [6](#_bookmark11)



ภาพประกอบททท **6:** ตต วอยพางการเกกบขตอมมลของชกดขตอมมล iris

การเรธยกดมขตอมมลสามารถทนาไดตโดยเรธยกใชต Key ทพธชพพอ data โดยตต วอยพางดต งตพอไปนตธ แสดงใหตเหกนถทงการเรธยก ใชตขตอมมลจนานวน 10 ชกด

**print("First ten rows of data:\n \**

**{}".format(iris\_dataset['data'][0:10]))**

First ten rows of data:

[[5.1 3.5 1.4 0.2]

|  |  |  |
| --- | --- | --- |
| [4.9 | 3. | 1.4 0.2] |
| [4.7 | 3.2 | 1.3 0.2] |
| [4.6 | 3.1 | 1.5 0.2] |
| [5. | 3.6 | 1.4 0.2] |
| [5.4 | 3.9 | 1.7 0.4] |
| [4.6 | 3.4 | 1.4 0.3] |
| [5. | 3.4 | 1.5 0.2] |
| [4.4 | 2.9 | 1.4 0.2] |
| [4.9 | 3.1 | 1.5 0.1]] |

หากตตองการเรธยกดมขตอมมล class สามารถทนาไดตโดยเรธยกใชต key ทพธชพพอ target ดต งตต วอยพางตพอไปนตธ

# ตต วอยพางผลลต พธล หรพอ class ของขตอมมล

**print("Target:\n{}".format(iris\_dataset['target']))**

Target:

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2]

จากตต วอยพางขตางตตน

1. หมายถทง iris setosa
2. หมายถทง iris versicolor
3. หมายถทง virginica

หากตตองการเรธยกดมขนาดของ target หรพอจนานวนของขตอมมลสามารถทนาไดตโดย

# แสดงจนานวนของผลลต พธล

**print("shape of target: \**

**{}".format(iris\_dataset['target'].shape))**

shape of target: (150,)

หากตตองการเรธยกดมประเภทของขตอมมลทพธใชตจต ดเกกบสามารถทนาไดตโดย

# ประเภทขตอมมลของ target

**print("Type of target: \**

**{}".format(type(iris\_dataset['target'])))**

Type of target: <type 'numpy.ndarray'>

ประเภทของขตอมมล (Data Type) ทพธใชตจต ดเกกบคพอ numpy.ndarray ซพทง numpy เปก นไลบรารพธ ทพธใชตจต ดการขตอมมลประเภทตต วเลขมธการจต ดเกกบแบบอารลเรยล (Array)

## MNIST Dataset

ขตอมมลชกด MNIST เปกนชกดขตอมมลลายมพอตต วเลขอารบวก (ตต วเลข 0-9) โดยรมปภาพมธขนาด

28x28 พวกเซล และมธขตอมมลในชกดเรธยนรมตจนานวน 60,000 รมปภาพ และขตอมมลชกดทดสอบจนานวน

10,000 รมปภาพ สนาหรต บ scikit-learn นต ตน MNIST เปกนชกดขตอมมลตต วอยพางทพธกนาหนดใหตรมปภาพตต ว อต กษรลายมพอตต วเลขอารบวกในแตพละรมปมธขนาด 8x8 พวกเซล และมธจนานวน 5,620 ชกดเทพานต ตน ตต วอยพาง ขตอมมลชกด MNIST แสดงดต งภาพประกอบทพธ [7](#_bookmark13)



ภาพประกอบททท **7:** ตต วอยพางของขตอมมลชกด MNSIT

การเรธยกใชตชกดขตอมมล MNIST ทพธมาพรตอมกต บ scikit-learn สามารถทนาไดตดต งตพอไปนตธ

**from sklearn import datasets digits = datasets.load\_digits()**

ดต งนต ตน หากตตองการดมรายชพพอของ key ของชกดขตอมมล MNSIT สามารถทนาไดตโดย

**print("Keys of digits: \**

**\n{}".format(digits.keys()))**

Keys of digits:

['images', 'data', 'target\_names', 'DESCR', 'target']

หากตตองการดมรายละเอธยดของชกดขตอมมล MNIST ทนาไดตโดย

**print(digits['DESCR'])**

Optical Recognition of Handwritten Digits Data Set

===================================================

Notes

Data Set Characteristics:

:Number of Instances: 5620

:Number of Attributes: 64

:Attribute Information: 8x8 image of integer pixels in the range 0..16.

:Missing Attribute Values: None

:Creator: E. Alpaydin (alpaydin '@' boun.edu.tr)

:Date: July; 1998

This is a copy of the test set of the UCI ML hand-written digits datasets [http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits](http://archive.ics.uci.edu/ml/datasets/Optical%2BRecognition%2Bof%2BHandwritten%2BDigits)

The data set contains images of hand-written digits: 10 classes where each class refers to a digit.

หากตตองการดมขนาดของชกดขตอมมล MNIST สามารถใชตคนาสตพง shape ซพทงเปกนเมธอด

(Method) หนพทงทพธใชตรพวมกต บไลบรารพธ numpy

**print("size of the MNIST dataset", digits.images.shape)**

('size of the MNIST dataset', (1797, 8, 8))

สามารถเรธยกดมขตอมมลชพพอของ target ไดตดต งนตธ

#### print("target name", digits.target\_names)

('target name', array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))

รมปภาพตต วเลขอารบวกทพธจต ดเกกบมธขนาด 8x8 พวกเซล หากตตองการตรวจสอบขตอมมลของ แตพละตต วเลขสามารถเรธยกดมไดตดต งนตธ

**print("Shape of each image", \ digits.images[0].shape)**

**digits.images[0]**

('Shape of each image', (8, 8))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| array([[ 0., 0., | 5., | 13., | 9., | 1., 0., 0.], |
| [ 0., 0., | 13., | 15., | 10., | 15., 5., 0.], |
| [ 0., 3., | 15., | 2., | 0., | 11., 8., 0.], |
| [ 0., 4., | 12., | 0., | 0., | 8., 8., 0.], |
| [ 0., 5., | 8., | 0., | 0., | 9., 8., 0.], |
| [ 0., 4., | 11., | 0., | 1., | 12., 7., 0.], |
| [ 0., 2., | 14., | 5., | 10., | 12., 0., 0.], |
| [ 0., 0., | 6., | 13., | 10., | 0., 0., 0.]]) |

คนาสตพง digits.images[0].shape ใชตเพพพอดมขนาดของตต วเลขในตนาแหนพงทพธ 0 และคนาสตพง digits.images[0] เปกนการแสดงขตอมมลทต งต หมดของตต วเลขในตนาแหนพงทพธ 0 ดต งนต ตน หากตตองการดม รายละเอธยดของขตอมมลมากกวพา 1 ชกด สามารถใชตคนาสตพงดต งตพอไปนตธ

**#**แสดงตตวอยทางขขอมมลจาจ นวน **2** ชชด ตตขงแตทชชดททท **print("Sample image 0 to 2") digits.images[0:2]**

**0** ถธง **2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sample image 0 | to 2 |  | | | | | |
| array([[[ 0., | 0., | 5., | 13., | 9., | 1., | 0., | 0.], |
| [ 0., | 0., | 13., | 15., | 10., | 15., | 5., | 0.], |
| [ 0., | 3., | 15., | 2., | 0., | 11., | 8., | 0.], |
| [ 0., | 4., | 12., | 0., | 0., | 8., | 8., | 0.], |
| [ 0., | 5., | 8., | 0., | 0., | 9., | 8., | 0.], |
| [ 0., | 4., | 11., | 0., | 1., | 12., | 7., | 0.], |
| [ 0., | 2., | 14., | 5., | 10., | 12., | 0., | 0.], |
| [ 0., | 0., | 6., | 13., | 10., | 0., | 0., | 0.]], |
| [[ 0., | 0., | 0., | 4., | 15., | 12., | 0., | 0.], |
| [ 0., | 0., | 3., | 16., | 15., | 14., | 0., | 0.], |
| [ 0., | 0., | 8., | 13., | 8., | 16., | 0., | 0.], |
| [ 0., | 0., | 1., | 6., | 15., | 11., | 0., | 0.], |
| [ 0., | 1., | 8., | 13., | 15., | 1., | 0., | 0.], |
| [ 0., | 9., | 16., | 16., | 5., | 0., | 0., | 0.], |
| [ 0., | 3., | 13., | 16., | 16., | 11., | 5., | 0.], |
| [ 0., | 0., | 0., | 3., | 11., | 16., | 9., | 0.]]]) |

การดมขตอมมล target ทต งต หมดสามารถทนาไดตโดยใชตคนาสตพง

**print("target (y / label / class) from 0 to 20") digits.target[:20]**

target (y / label / class) from 0 to 20

array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

จากตต วอยพางขตางตตน คนาสตพง digits.target[:20] หมายถทงเรธยกดมขตอมมลตต งต แตพตนาแหนพงทพธ 0 ถทง ตนาแหนพงทพธ 20

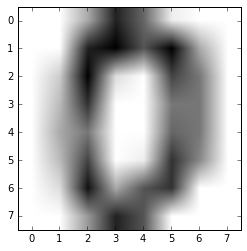
การ **Visualization** เพทพอดมรมปภาพตตวเลข

การแสดงขตอมมล (Visualization) สามารถทนาไดตหลากหลายววธธ เชพน ใชตฟต งกลชต นจาก scikit- image, matplotlib และ pylab การแสดงขตอมมลสามารถทนาไดตดต งตพอไปนตธ

การแสดงขขอมมลดขวย **pylab**

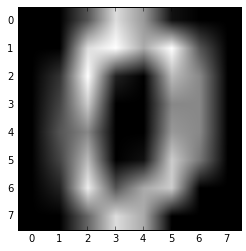
**import pylab**

**pylab.imshow(digits.images[0], cmap=pylab.cm.gray\_r) pylab.show()**

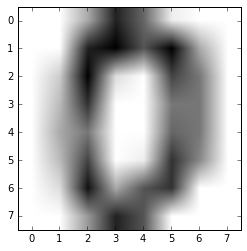
****

การแสดงขขอมมลดขวย **matplotlib**

**import matplotlib.pyplot as plt plt.imshow(digits.images[0], cmap=plt.get\_cmap('gray')) plt.show()**

****

**import matplotlib.pyplot as plt plt.imshow(digits.images[0], cmap=plt.cm.gray\_r) plt.show()**

****

ขขอมมลทททนจาไปใชขใน **scikit-learn**

ขตอมมลทพธจะนนาไปใชตใน scikit-learn จะตตองแปลงใหตเปกนเวกกเตอรล (Vector) ใน 1 ตต วเลขชอง ชกดขตอมมล MNIST จต ดเกกบแบบ array 2 มวตว จทงตตองแปลงใหตอยมพในรมปแบบของ vector ดต งนต ตน รมปภาพขนาด 8x8 พวกเซล เมพพอแปลงใหตเปกน vector จทงเปกนเวกกเตอรลขนาด 64 การแปลงใหตเปกนเวกก เตอรลในกรณธ นตธ ใชตคนาสตพง reshape สามารถทนาไดตดต งตพอไปนตธ

**data = digits.images.reshape((digits.images.shape[0], -1)) print("size of the data", data.shape)**

('size of the data', (1797, 64))

จากตต วอยพางขตางตตน การปรต บเปลพธยนรมปรพางขตอมมลทนาไดตโดยใชตคนาสตพง reshape จากนต ตนสามารถ ใชตคนาสตพง data.shape เพพพอดมขนาดของขตอมมลหลต งจากการปรต บเปลพธยน โดยขตอมมลใหมพจะมธขนาดเปกน (1797, 64) สามารถเรธยกดมขตอมมลในแตพละแถวดต งตพอไปนตธ

**print("size of each row", data[0].shape) data[0]**

('size of each row', (64,))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| array([ 0., | 0., 5., | 13., 9., 1., | 0., | 0., | 0., 0., 13., | 15., 10., |
| 15., | 5., 0., | 0., 3., 15., | 2., | 0., | 11., 8., 0., | 0., 4., |
| 12., | 0., 0., | 8., 8., 0., | 0., | 5., | 8., 0., 0., | 9., 8., |
| 0., | 0., 4., | 11., 0., 1., | 12., | 7., | 0., 0., 2., | 14., 5., |
| 10., | 12., 0., | 0., 0., 0., | 6., | 13., | 10., 0., 0., | 0.]) |

โหลดชชดขขอมมล **MNIST** ดขวย **Scipy**

เนพพ องจากการโหลดชกดขตอมมล MNIST จาก scikit-learn สามารถใชตไดตเพธยงชกดขตอมมล ตต วอยพางทพธมธขนาดของรมปภาพเพธยง 8x8 พวกเซล และมธจนานวน 5,620 ชกดขตอมมลเทพานต ตน ดต งนต ตน หาก ตตองการใชตชกดขตอมมล MNIST ชกดสมบมรณลจะตตองใชตขตอมมลจากแหลพงอพพน สามารถดาวนลโหลด MNIST ไดตจากลวงกลตพอไปนตธ

<https://github.com/amplab/datascience-sp14/blob/master/lab7/mldata/mnist-> [original.mat](https://github.com/amplab/datascience-sp14/blob/master/lab7/mldata/mnist-original.mat)

ไฟลลทพธโหลดจะมธนามสกกล .mat ซพทงจต ดเกกบอยมพในฟอรลแมต (Format) ของโปรแกรม MATLAB ไฟลลมธขนาด 55.4 MB ซพทงมธขนาดใหญพพอสมควร เมพพอดาวนลโหลดไฟลลเสรกจเรธยบรตอย สามารถเรธยกใชต ดต งคนาสตพงดต งตพอไปนตธ

**from scipy.io import loadmat**

**mnist\_raw = loadmat("mldata/mnist-original.mat") mnist = {**

**"data": mnist\_raw["data"].T,**

**"target": mnist\_raw["label"][0],**

**"COL\_NAMES": ["label", "data"],**

**"DESCR": "mldata.org dataset: mnist-original",**

**}**

จากคนาสตพง **loadmat("mldata/mnist-original.mat")** จะตตองระบกตนาแหนพงของไฟลลใหต ถมกตตอง ในตต วอยพางไฟลล .mat จต ดเกกบอยมพทพธโฟลเดอรล mldata

สามารถดมรายละเอธยดของขตอมมล MNSIT โดยพวมพลคนาสตพง mnist ดต งตต วอยพางตพอไปนตธ

**mnist**

{'COL\_NAMES': ['label', 'data'],

'DESCR': 'mldata.org dataset: mnist-original', 'data': array([[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

...,

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0]], dtype=uint8),

'target': array([0., 0., 0., ..., 9., 9., 9.])}

ตต วแปร mnist จต ดเกกบในรมปแบบของ dictionary โดยจะใหตเกกบขตอมมลทพธมธจนานวน 70,000 รมปภาพ โดยทพธรมปภาพทพธ 1-60,000 ใชตสนาหรต บการเรธยนรมต (Training Set) และรมปภาพทพธ 60,001- 70,000 ใชตสนาหรต บการทดสอบ (Test Set) แตพละรมปภาพถมกกนาหนดใหตมธขนาด 28x28 พวกเซล และ แตพละพวกเซลจะเกกบคพา 0 (white) และ 255 (black)

รายละเอธยดของ key ทพธอยมพในตต วแปร mnsit แสดงดต งตพอไปนตธ

**data** ใชตเกกบขตอมมลพวกเซลของรมปภาพทต งต หมด

**target** ใชตเกกบ target / class / label ของรมปภาพตต วเลข

จากนต ตนสรตางตต วแปร X,y เพพพอจต ดเกกบขตอมมลจากตต วแปร mnsit เพพพอนนาไปใชตงาน

**X,y = mnist['data'], mnist['target']**

**X.shape, y.shape**

((70000, 784), (70000,))

โดยตต วแปร X ใชตสนาหรต บจต ดเกกบขตอมมล (data) ตต วเลขทต งต หมด และตต วแปร y ใชตสนาหรต บเกกบ ขตอมมล label

จากตต วอยพางขตางตตน ตต วแปร x จะมธขนาด (70000, 784) นตพ นหมายถทงมธจนานวน 70,000 ตต วเลข แตพละตต วเลขมธขนาด 784 attribute (28x28) และตต วแปร y มธขนาด 70000 label ซพทงตตอง สตมพต นธลกต บจนานวนของตต วแปร x

ตต วเลขแตพละตต วถมกจต ดเกกบเปกนแบบ vector ดต งนต ตน การแสดงขตอมมลจะตตองแปลงกลต บใหตเปกน

array 2 มวตวโดยใชตคนาสตพง reshape จากนต ตนจทงจะสามารถแสดงขตอมมลแบบ Visualization ไดต

**import matplotlib.pyplot as plt some\_digit = X[36000]**

**some\_digit\_image = some\_digit.reshape(28, 28)**

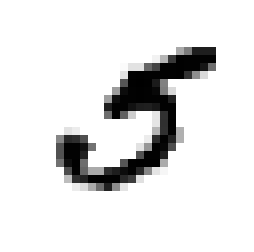
**plt.imshow(**

**some\_digit\_image, cmap = plt.cm.binary,**

**interpolation="nearest")**

**plt.axis("off") plt.show()**

**y[36000]**



5.0

เพพพอสะดวกตพอการแสดงผล สามารถสรตางเปนฟต งกลชต นในการ Visualization ขตอมมล ดต งตต วอยพางตพอไปนตธ

**import matplotlib.pyplot as plt def plot\_digit(X, y):**

**digit\_image = X.reshape(28, 28)**

**plt.imshow(**

**digit\_image,**

**cmap = plt.cm.binary, interpolation="nearest")**

**plt.axis("off") plt.show()**

**print("Label", y)**

โดย

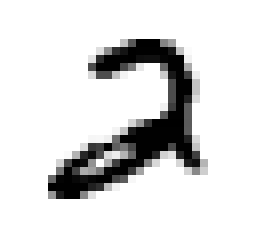
**def** คพอการประกาศฟต งกลชต นในภาษา Python

**plot\_digit** คพอชพพอของฟต งกลชต น

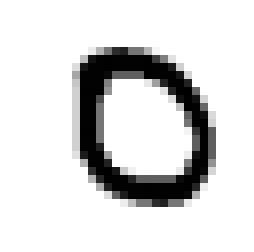
**plot\_digit(X, y)** คพอการประกาศวพาฟต งกลชต น plot\_digit กนาหนดใหตรต บคพา พารามวเตอรลจนานวน 2 คพา คพอ X และ y

จากนต ตนสามารถเรธยกใชตฟต งกลชต น plot\_digit ดต งตต วอยพางตพอไปนตธ

**plot\_digit(X[15000], y[15000]) plot\_digit(X[2000], y[2000])**

****

('Label', 2.0)



('Label', 0.0)

การ **Visualization** ชชดขขอมมล **MNIST**

ชกดขตอมมล MNIST เปกนชกดขตอมมลตต วเลขอารบวก ทพธมธจนานวน 70,000 ตต วเลข หากตตองการทพธจะ Visualization

เพพพอใหตเหกนขตอมมลไดตอยพางชต ดเจน และงพายตพอการเขตาใจสามารถทนาไดตโดย คนาสตพงตพอไปนตธ

**import matplotlib.pyplot as plt**

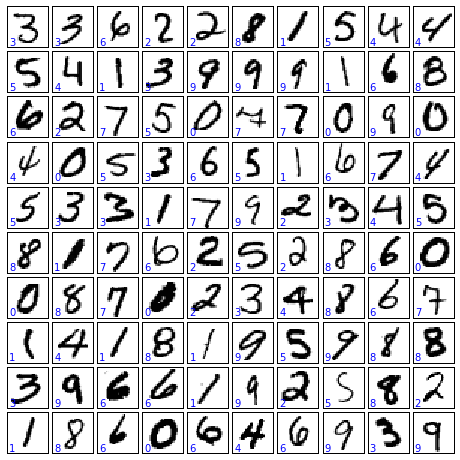
**fig, axes = plt.subplots(10, 10, figsize=(8, 8),**

**subplot\_kw={'xticks':[], 'yticks':[]}, gridspec\_kw=dict(hspace=0.1, wspace=0.1))**

**for i, ax in enumerate(axes.flat): ax.imshow(X\_train[i].reshape((28,28)), cmap='binary',**

**interpolation='nearest')**

**ax.text(0.05, 0.05, str(int(y\_train[i])), transform=ax.transAxes, color='blue')**

****

ภาพประกอบททท **8:** การ Visualization ชกดขตอมมล MNIST พรตอมทต งต แสดง label ของแตพละตต วเลข

จากภาพประกอบทพธ [8](#_bookmark18) แสดงใหตเหกนถทงการ Visualization ขตอมมลตต วเลขทต งต สวตน 100 รมปภาพ พรตอมทต งต แสดง target/class ทพธถมกตตองของขตอมมลทต งต หมด

บทททท **3** ขขอมมลชชดเรทยนรขมและขขอมมลชชดทดสอบ **(Training and Test Data)**

กระบวนการเรธยนรมตของเครพพองจต กรจะตตองแบพงขตอมมลออกเปกนสองสพวน ประกอบดตวย ขตอมมลชกด เรธยนรมต (Training Set) และขตอมมลชกดทดสอบ (Test Set) ซพทงขตอมมลชกดเรธยนรมตจะถมกนนาไปเรธยนดตวย ดตวยววธธการเรธยนรมตเครพพองจต กรเพพพอสรตางออกมาเปกนโมเดล (Model) จากนต ตนจทงทดสอบโมเดลทพธสรตาง ดตวยขตอมมลชกดทดสอบ หากโมเดลทพธสรตางมธประสวทธวภาพดธจทงนนาโมเดลนต ตนไปใชตงานจรวง บางชกดขตอมมล อาจไมพไดตแบพงขตอมมลออกเปกนสองชกดทพธชต ดเจน ดต งนต ตนจทงมธความจนาเปกนทพธจะตตองเขธยนโปรแกรมเพพพอ แบพงขตอมมลออกเปกนสองสพวน

โปรแกรม scikit-learn มธเครพพองมพอทพธชพวยในการแบพง (Split) และสลต บ (Shuffle) ขตอมมล โดยใชตฟต งกลชต นดต งตพอไปนตธ

**from sklearn.model\_selection import train\_test\_split # Loading iris dataset**

**from sklearn.datasets import load\_iris iris\_dataset = load\_iris()**

**# split and shuffle data**

**X\_train, X\_test, y\_train, y\_test = \ train\_test\_split(iris\_dataset['data'], \ iris\_dataset['target'], random\_state=0)**

จากตต วอยพางขตางตตน เมพพอใชตคนาสตพง train\_test\_split โปรแกรมจะแบพงขตอมมล และนนามาจต ดเกกบ ลงไปในตต วแปลจนานวน 4 ตต วประกอบดตวย X\_train, X\_test, y\_train และ y\_test โดยเปกนขตอมมล ประเภท numpy.ndarray โปรแกรมจะแบพงขตอมมลออกเปกน 75% สนาหรต บเรธยนรมต และเกกบอยมพใน ตต วแปร X\_train และ y\_train สพวนทพธเหลพออธก 25% สนาหรต บทดสอบ จะถมกจต ดเกกบในตต วแปร X\_test และ y\_test

สามารถเรธยกดมขนาดของขตอมมลดตวยคนาสตพง shape ดต งตต วอยพางตพอไปนตธ

**print("X\_train shape: {}".format(X\_train.shape)) print("y\_train shape: {}".format(y\_train.shape)) print("X\_test shape: {}".format(X\_test.shape)) print("y\_test shape: {}".format(y\_test.shape))**

X\_train shape: (112, 4)

y\_train shape: (112,)

X\_test shape: (38, 4)

y\_test shape: (38,)

### การกาจ หนดอตตราทททใชขสาจ หรตบแบทงขขอมมล

โปรแกรม scikit-learn อนก ญาตใหตผมตใชตแบพงขนาดของขตอมมลชกดเรธยนรมตและชกดทดสอบไดต โดย ใชต keyword คพอ test\_size สามารถทนาไดตดต งตพอไปนตธ

**#** กจาหนดใหข **test set** มทจาจ นวนขขอมมล **50%** จากขขอมมลทตขงหมด

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris\_dataset['data'], \ iris\_dataset['target'], test\_size=0.5, \**

**random\_state=0)**

จากตต วอยพางกนาหนดใหต test\_size = 0.5 คพอการระบกขนาดของ Train set และ Test set ใหตมธ ขนาดเปกน 50% สามารถตรวจสอบขนาดของขตอมมลไดตดต งนตธ

**print("X\_train shape: {}".format(X\_train.shape)) print("y\_train shape: {}\n".format(y\_train.shape)) print("X\_test shape: {}".format(X\_test.shape)) print("y\_test shape: {}".format(y\_test.shape))**

X\_train shape: (75, 4)

y\_train shape: (75,)

X\_test shape: (75, 4)

y\_test shape: (75,)

### การแสดงขขอมมลในรมปแบบของกราฟ

จากตต วอยพางขตางตตนไดตกนาหนด test\_size = 50 ทนาใหตแบพงขตอมมลออกเปกน Training set และ Test set ชกดละเทพา ๆ กต น ขตอมมลทต งต สองสพวนสามารถ Visualization ใหตเหกนขตอมมลเขตาใจขตทน โดย แสดงในลต กษณะของกราฟ โดยกราฟทพธจะนนาเสนอเปกนกราฟ 2 มวตว ประกอบดตวยแกน X และแกน Y ดต งนต ตน จทงตตองเลพอกขตอมมลจากชกดขตอมมล iris ทพธมธทต งต หมด 4 attribute มาแสดงเพธยง 2 attribute

Fundamentals of Machine Learning and Analyzing Data with Python 23

การแสดงขตอมมล Training set สามารถทนาไดตดต งนตธ

**import numpy as np**

**import matplotlib.pyplot as plt**

**from matplotlib.colors import ListedColormap**

**markers = ('s', 'x', 'o')**

**colors = ('red', 'blue', 'lightgreen')**

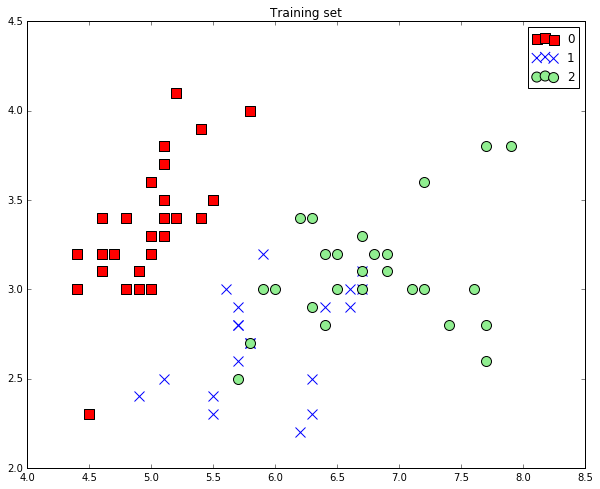
**cmap = ListedColormap(colors[:len(np.unique(y\_test))]) plt.figure(figsize=(10, 8))**

**for idx, cl in enumerate(np.unique(y\_train)): plt.scatter(x=X\_train[y\_train == cl, 0],**

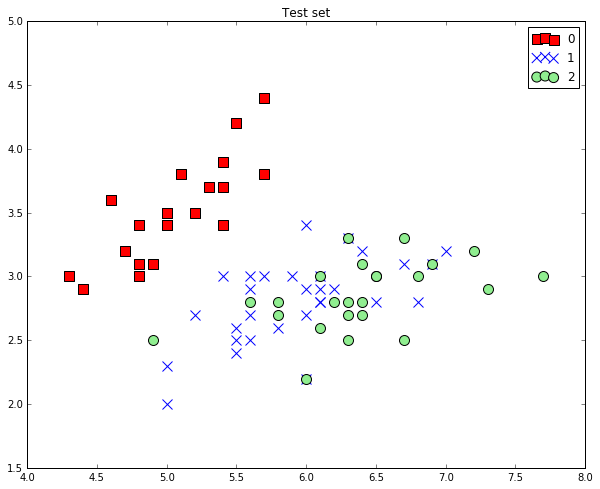
**y=X\_train[y\_train == cl, 1], c=cmap(idx), marker=markers[idx], label=cl, s=100)**

**plt.title('Training set') plt.legend()**

**plt.show()**

****

เชพนเดธยวกต บขตอมมล Test set สามารถแสดงในรมปแบบของกราฟ โดยตตองเปลพธยนตต วแปรจาก X\_train และ y\_train ใหตเปกน X\_test และ y\_test ตามลนาดต บ การแสดงขตอมมล Test set แสดงดต ง ตต วอยพางตพอไปนตธ



บทททท **4** ไลบรารทท **Matplotlib (Matplotlib Library)**

Matplotlib เปก นไลบรารพธทพธใชตในโปรแกรมภาษา Python สนาหรต บการ Visualize ขตอมมล (Data Visualization) หรพอพลอต (Plot) ขตอมมลทต งต ในรมปแบบ 2 มวตว และ 3 มวตว การตวดตต งต ไลบรารพธ Matplotlib ในลวนก กซล (Linux) สามารถทนาไดตโดยพวมพลคนาสตพงลงไปในเทอรลมวนต ล ดต งนตธ

**$ pip install matplotlib**

ทต งต นตธ ยต งสามารถตวดตต งต ผพาน Jupyter โดยพวมพลคนาสตพงดต งตพอไปนตธ

**$ ! pip install matplotlib**

หลต งจากตวดตต งต Matplotlib เสรกจเรธยบรตอยสามารถเรธยกใชตภายในโปรแกรมภาษา Python

โดยใชตคนาสตพงดต งตพอไปนตธ เพพพออวมพอรลตไลบรารพธ Matplotlib เขตามาใชตงาน

**import matplotlib.pyplot as plt**

ตต วอยพางของการสรตางกราฟโดยใชต Matplotlib แสดงดต งตต วอยพางตพอไปนตธ

# ----------file: my\_plot.py ------------ **import matplotlib.pyplot as plt import numpy as np**

**x = np.linspace(0, 10, 100) plt.plot(x, np.sin(x))**

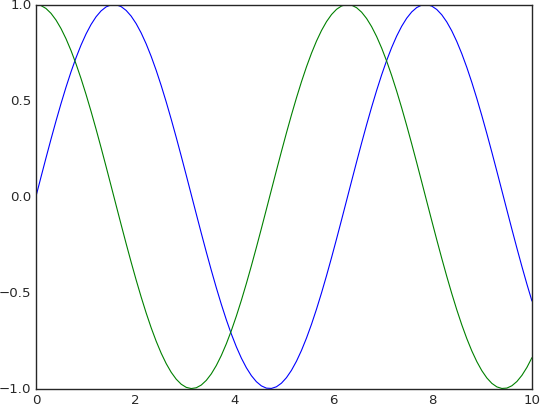
**plt.plot(x, np.cos(x))**

**plt.show()**

จากตต วอยพางขตางตตนเปกนสครวปตล (Script) ภาษา Python ทพธบต นททกไวตในไฟลลชพพอ my\_plot.py ดต งนต ตน หากตตองการจะเรธยกใชตไฟลลภาษา Python เพพพอทนางาน (Run) จะตตองพวมพลคนาสตพงทพธ Terminal ดต งตต วอยพางตพอไปนตธ

**$ python my\_plot.py**

โปรแกรมจะแสดงกราฟออกทางจอภาพ ดต งตต วอยพางตพอไปนตธ



โปรแกรม Matplotlib อนก ญาตใหตทนาการปรต บเปลพธยนรมปแบบของกราฟไดตตามตตองการ ดต ง ตต วอยพางตพอไปนตธ

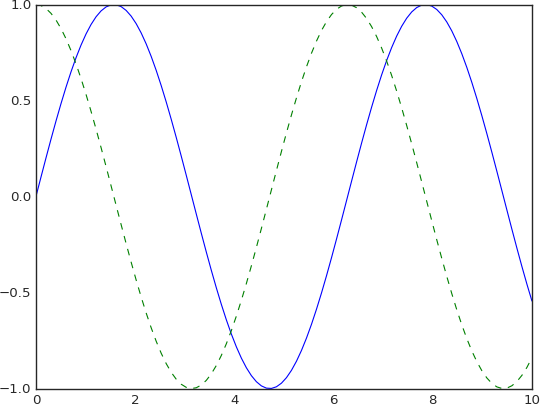
**import matplotlib.pyplot as plt import numpy as np**

**x = np.linspace(0, 10, 100)**

**fig = plt.figure() plt.plot(x, np.sin(x), '-')**

**plt.plot(x, np.cos(x), '--') plt.show()**

ผลลต พธลทพธไดตแสดงดต งตต วอยพางตพอไปนตธ



การบตนทธกกราฟ **(Figure)**

ในโปรแกรม Matplotlib สามารถบต นททกกราฟใหตเปกนรมปภาพ โดยใชตคนาสตพง savefig() โดย การบต นททก จะบต นททกในรมปแบบของไฟลล PNG ตต วอยพางการบต นททกแสดงดต งตต วอยพางตพอไปนตธ

**fig.savefig('my\_figure.png')**

จากตต วอยพางขตางตตน เปกนการบต นททกรมปภาพเกกบลงไปในโฟลเดอรล หากตตองการทพธจะตรวจสอบ รมปภาพ (ในกรณธ นตธ ชพพอทพธถมกกนาหนดคพอ my\_figure.png) จากนต ตนสามารถตรวจสอบรมปภาพ my\_figure.png ทพธบต นททกลงไป สามารรถทนาไดตโดยใชตคนาสตพง ls เพพพอดมขตอมมล แตพในกรณธ ทพธตตองการดม ผพาน Jupyter Notebook จะตตองใชตเครพพองหมาย "!” เขตามาชพวย ดต งนต ตน เครพพองหมาย "!” จทงเปรธยบ เสมพอนการทนางานผพานหนตาจอเทอรลมวนต ล

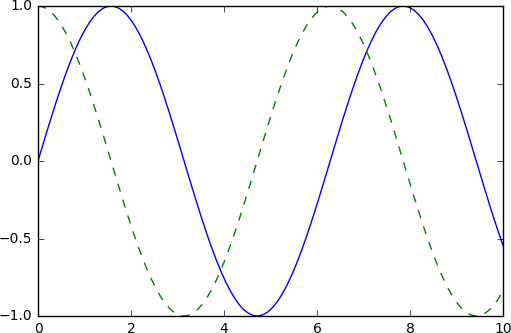
#### ! ls -lh my\_figure.png

-rw-rw-r-- 1 mrolarik mrolarik 26K Jul 19 00:12 my\_figure.png

การแสดงรมปภาพ **(Image Show)**

ในโปรแกรม Python สามารถเลพอกใชตไลบรารพธตพาง ๆ เพพพอใชตสนาหรต บการเปว ดรมปภาพจากไฟลล แสดงดต งตพอไปนตธ

**from IPython.display import Image Image('my\_figure.png')**

****

ววธทการแสดงรมปภาพโดยใชขอวนเทอรหเฟซ **(Interface)** ทททตทางกตน

#### Matlab-style Interface

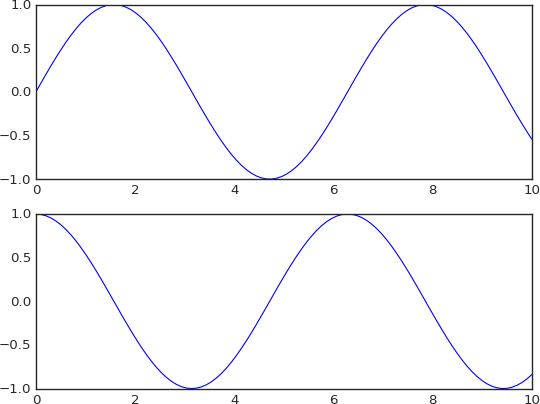
**plt.figure()** # สรตาง plot figure

# สรตาง panel แรกจากทต งต หมด 2 panel และเซกต axis ใหตกต บกราฟแรก **plt.subplot(2, 1, 1)** # (rows, columns, panel number) **plt.plot(x, np.sin(x))**

# สรตาง panel ทพธ 2 และเซกต axis

**plt.subplot(2, 1, 2) plt.plot(x, np.cos(x))**

**plt.show()**



#### Object-oriented Interface

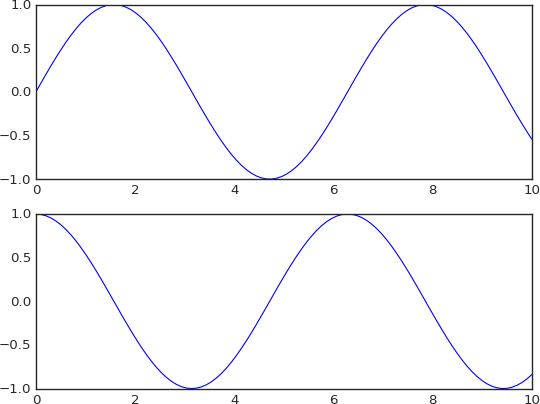
# สรตาง grid

# ax จะเปกน array ของ Axes object

**fig, ax = plt.subplots(2)** # สรตาง Axes จนานวน 2 object

# เรธยกใชตเมธอด (Method) plot()

**ax[0].plot(x, np.sin(x))**

**ax[1].plot(x, np.cos(x)) plt.show()**

การเพทวมเสขนตาราง **(Grid)** ในการพลอต

การเพพวมเสตนตาราง หรพอ Grid ในการพลอตทนาไดตโดยใชตคนาสตพง plt.style.use('seaborn- whitegrid') แสดงดต งตพอไปนตธ

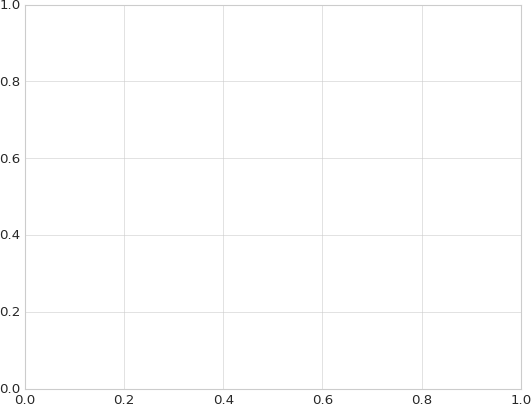
**import numpy as np**

**import matplotlib.pyplot as plt**

**plt.style.use('seaborn-whitegrid') fig = plt.figure()**

**ax = plt.axes()**

ผลลต พธลทพธไดตคพอกราฟทพธแสดงจะมธเสตน Grid ทต งต ในแนวตต งต และแนวนอน

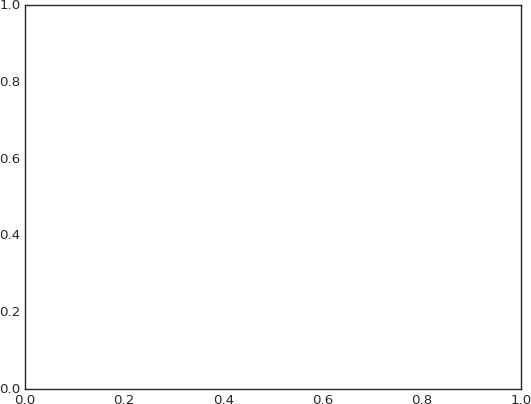


หากไมพตตองการแสดงเสตน Grid สามารถใชตคนาสตพง plt.style.use('seaborn-white')

ดต งตต วอยพางตพอไปนตธ

**plt.style.use('seaborn-white') fig = plt.figure()**

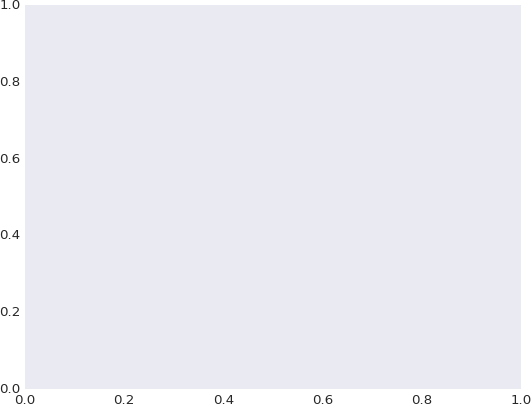
**ax = plt.axes() plt.show()**



หากตตองการใหตพตพนหลต งมพด (Dark) สามารถใชตคนาสตตพง plt.style.use('seaborn-dark') พตพน หลต งของกราฟจะเปลพธยนจากสธขาวเปกนโทนสธฟต า

**plt.style.use('seaborn-dark') fig = plt.figure()**

**ax = plt.axes() plt.show()**

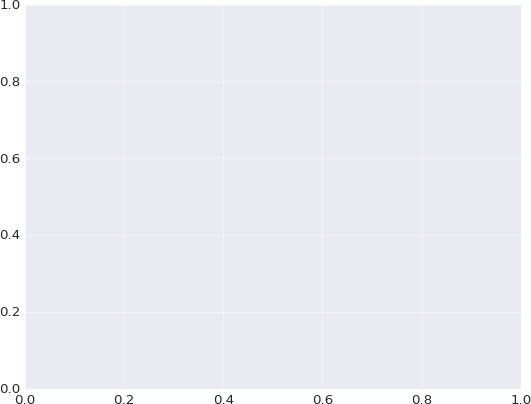
****

หากตตองการใหตพตพนหลต งมพด (Dark) และมธเสตน Grid สามารถใชตคนาสตตพง

plt.style.use('seaborn-darkgrid')

**plt.style.use('seaborn-darkgrid') fig = plt.figure()**

**ax = plt.axes() plt.show()**

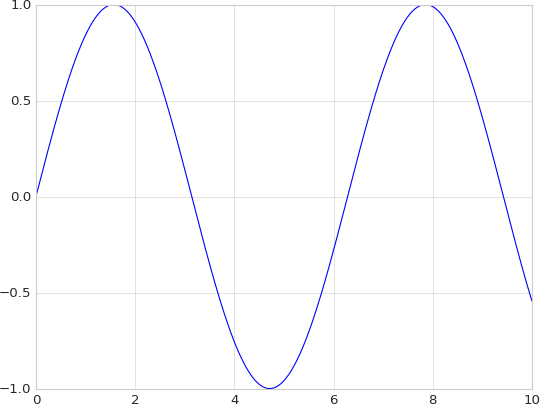
****

ตต วอยพางการพลอตกราฟเสตน

**plt.style.use('seaborn-whitegrid') fig = plt.figure()**

**ax = plt.axes()**

**x = np.linspace(0, 10, 1000) ax.plot(x, np.sin(x)) plt.show()**

****

หากตตองการใหตพลอตจนานวนสองเสตนใน Figure เดธยวกต น สามารถทนาไดตโดยใชตคนาสตพง

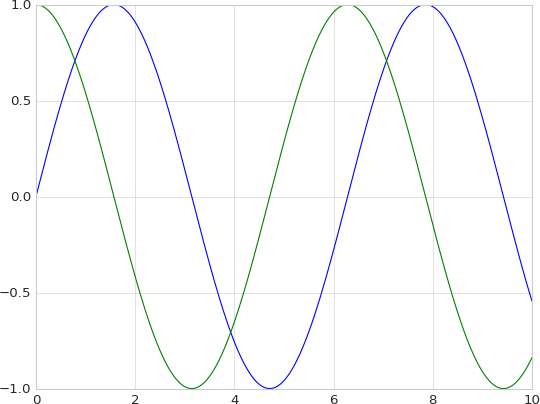
plt.plot() ไดตมากกวพา 1 ครต งต ดต งตพอไปนตธ

**plt.style.use('seaborn-whitegrid') plt.figure()**

**plt.axes()**

**x = np.linspace(0, 10, 1000) plt.plot(x, np.sin(x))**

**plt.plot(x, np.cos(x)) plt.show()**



การกาจ หนดววธทการพลอต**: Line Color** และ **Style**

กนาหนด Line Color

**plt.style.use('seaborn-whitegrid') plt.plot(x, np.sin(x-0), color='blue')**

# กนาหนดสธโดยใชตชพพอ

**plt.plot(x, np.sin(x-1), color='g')**

# กนาหนดสธโดยใชตชพพอยพอ จากโหมดสธ rgbcmyk

**plt.plot(x, np.sin(x-2), color='0.75')**

# กนาหนดสธโดยใชตคพาสธเทา ระหวพาง 0 ถทง 1

**plt.plot(x, np.sin(x-3), color='#FFDD44')**

# กนาหนดสธโดยใชต Hex code (RRGGBB จาก 00 ถทง FF)

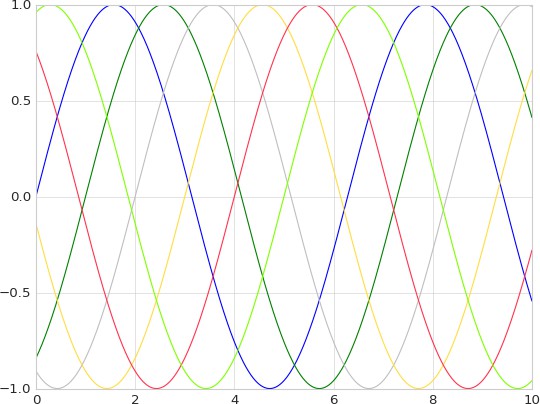
**plt.plot(x, np.sin(x-4), color=(1.0,0.2,0.3))**

# กนาหนดสธโดยใชตคพาสธ RGB tuple, คพาจาก 0 ถทง 1

**plt.plot(x, np.sin(x-5), color='chartreuse')**

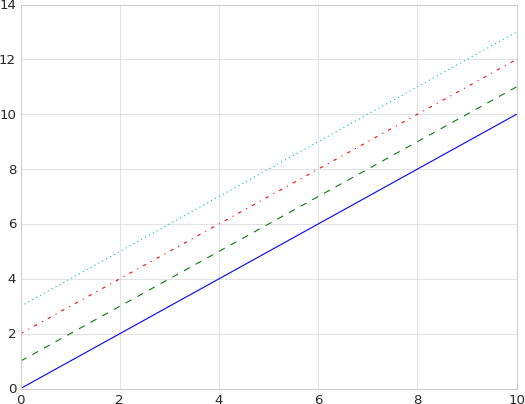
# กนาหนดสธโดยใชตคพาสธ HTML

**plt.show()**

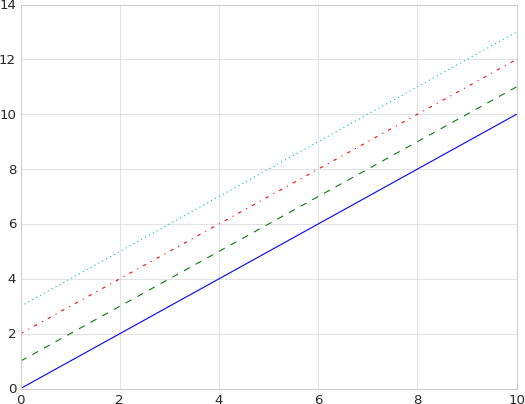


กนาหนด Line Style

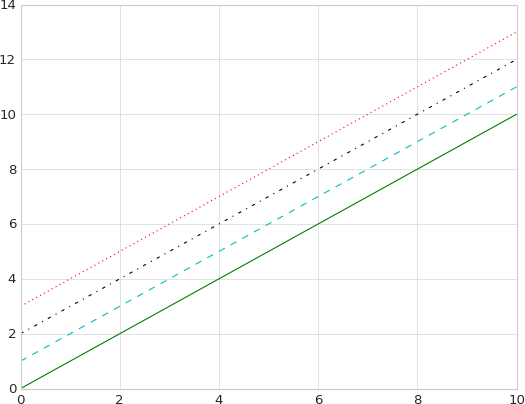
**plt.plot(x, x + 0, linestyle='solid') plt.plot(x, x + 1, linestyle='dashed') plt.plot(x, x + 2, linestyle='dashdot') plt.plot(x, x + 3, linestyle='dotted') plt.show()**

****

**plt.plot(x, x + 0, linestyle='-')** # solid **plt.plot(x, x + 1, linestyle='--')** # dashed **plt.plot(x, x + 2, linestyle='-.')** # dashdot **plt.plot(x, x + 3, linestyle=':')** # dotted **plt.show()**

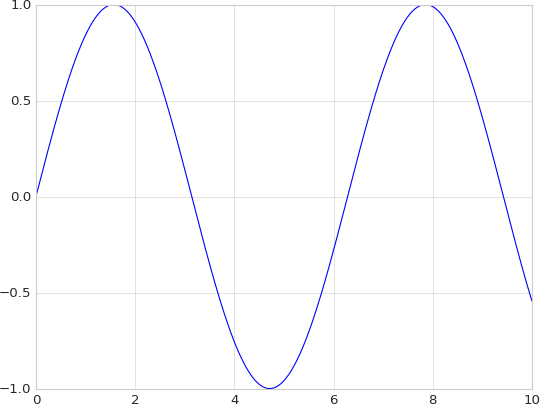
****

**plt.plot(x, x + 0, '-g')** # solid green **plt.plot(x, x + 1, '--c')** # dashed cyan **plt.plot(x, x + 2, '-.k')** # dashdot black **plt.plot(x, x + 3, ':r')** # dotted red **plt.show()**

****

การกาจ หนด **Axes Limit**

**plt.plot(x, np.sin(x)) plt.show()**

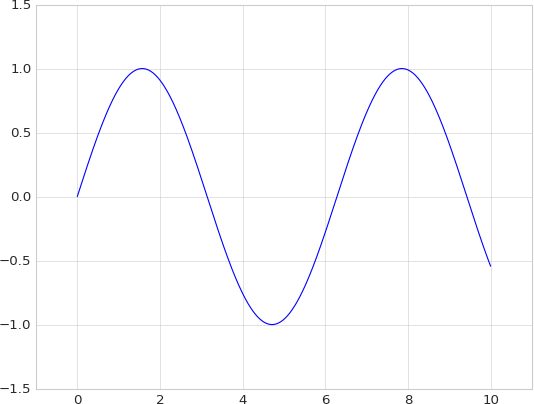
****

จากตต วอยพางขตางตตน กราฟทพธแสดงจะมธขนาดพอดธกต บเสตนกราฟ หากตตองการกนาหนด Axes Limit สามารถทนาไดตโดยใชตคนาสตพง plt.xlim(), plt.ylim() แสดงดต งตพอไปนตธ

**plt.plot(x, np.sin(x))**

**plt.xlim(-1,11)**

**plt.ylim(-1.5, 1.5) plt.show()**



จากตต วอยพางขตางตตน Scale ของแกน X และแกน Y จะเปลพธยนไปตามทพธกนาหนดไวตใน

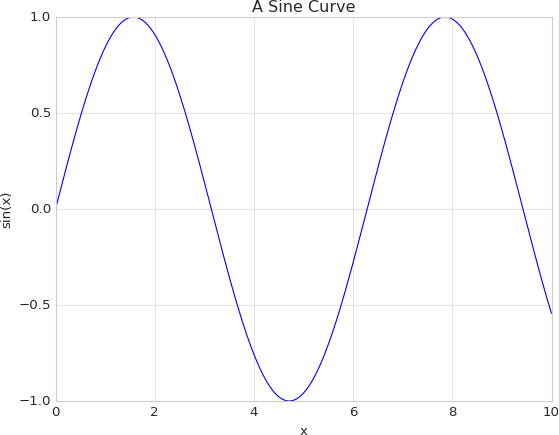
plt.xlim(-1, 11) และ plt.ylim(-1.5, 1.5)

การพลอต **Label**

หากตตองการเพพวม Label ใหตกต บกราฟ เชพน title, x-axis, y-axis สามารถทนาไดตโดย

**plt.plot(x, np.sin(x)) plt.title("A Sine Curve") plt.xlabel("x")**

**plt.ylabel("sin(x)") plt.show()**

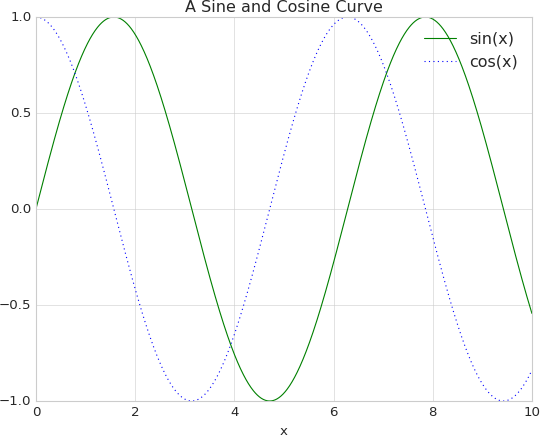
****

การพลอต **Legend**

**plt.plot(x, np.sin(x), '-g', label='sin(x)')**

**plt.plot(x, np.cos(x), ':b', label='cos(x)') plt.title("A Sine and Cosine Curve") plt.xlabel("x")**

**plt.legend() plt.show()**

****

การกาจ หนดการพลอต **Legend**

การเพพวมกรอบในกต บ Legeng

**import numpy as np**

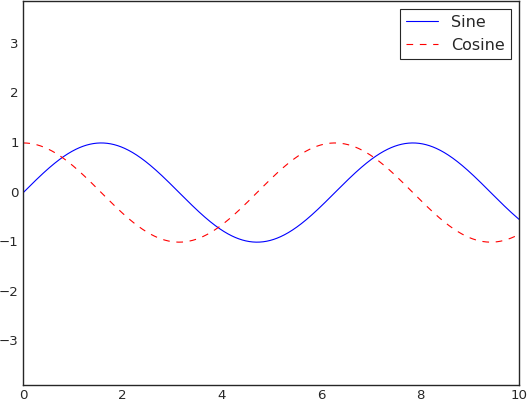
**import matplotlib.pyplot as plt plt.style.use('seaborn-white')**

**x = np.linspace(0, 10, 1000) fig, ax = plt.subplots()**

**ax.plot(x, np.sin(x), '-b', label='Sine')**

**ax.plot(x, np.cos(x), '--r', label='Cosine') ax.axis('equal')**

**leg = ax.legend(frameon=True)**

****

การเปลพธยนตนาแหนพง Legend ใหตอยมพมกมบนซตาย

**import numpy as np**

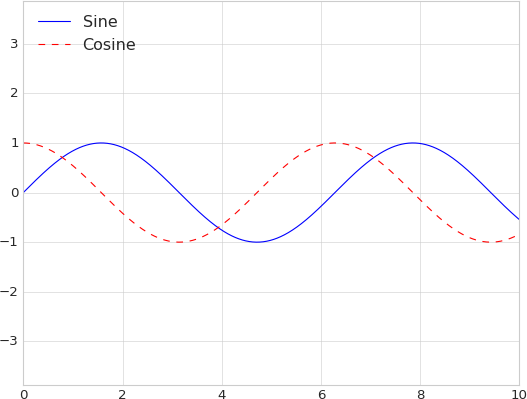
**import matplotlib.pyplot as plt plt.style.use('seaborn-whitegrid')**

**x = np.linspace(0, 10, 1000) fig, ax = plt.subplots()**

**ax.plot(x, np.sin(x), '-b', label='Sine')**

**ax.plot(x, np.cos(x), '--r', label='Cosine') ax.axis('equal')**

**ax.legend(loc='upper left', frameon=False) plt.show()**

****

การเปลพธยนตนาแหนพง Legend ใหตอยมพดตานลพางตรงกลาง

**import numpy as np**

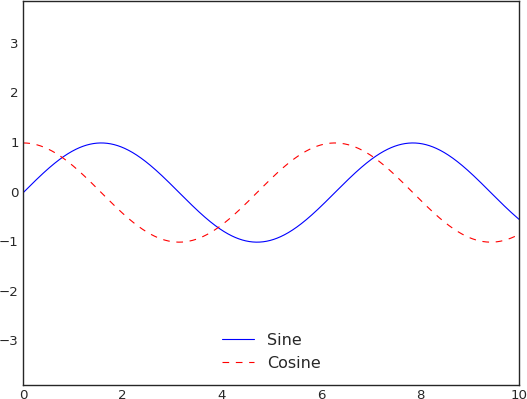
**import matplotlib.pyplot as plt plt.style.use('seaborn-white')**

**x = np.linspace(0, 10, 1000) fig, ax = plt.subplots()**

**ax.plot(x, np.sin(x), '-b', label='Sine')**

**ax.plot(x, np.cos(x), '--r', label='Cosine') ax.axis('equal')**

**ax.legend(loc='lower center', frameon=False) plt.show()**

****

**import numpy as np**

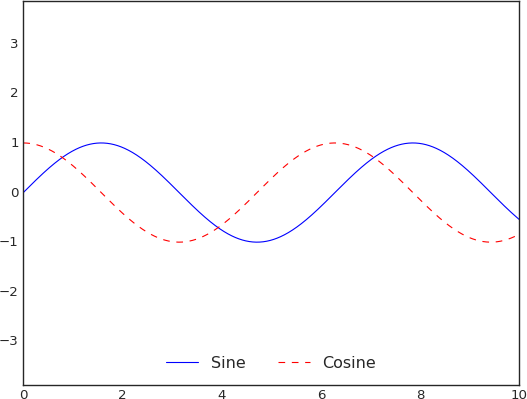
**import matplotlib.pyplot as plt plt.style.use('seaborn-white')**

**x = np.linspace(0, 10, 1000) fig, ax = plt.subplots()**

**ax.plot(x, np.sin(x), '-b', label='Sine')**

**ax.plot(x, np.cos(x), '--r', label='Cosine') ax.axis('equal')**

**ax.legend(loc='lower center', ncol=2, frameon=False) plt.show()**

****

การตกแตพง Legend

**import numpy as np**

**import matplotlib.pyplot as plt plt.style.use('seaborn-white')**

**x = np.linspace(0, 10, 1000) fig, ax = plt.subplots()**

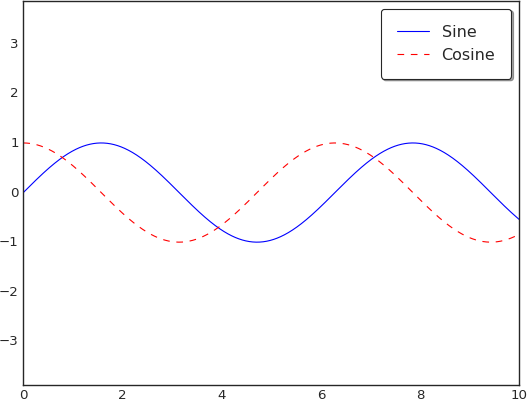
**ax.plot(x, np.sin(x), '-b', label='Sine')**

**ax.plot(x, np.cos(x), '--r', label='Cosine') ax.axis('equal')**

**leg = ax.legend(frameon=True, fancybox=True, framealpha=1,**

**shadow=True, borderpad=1)**

**plt.show()**

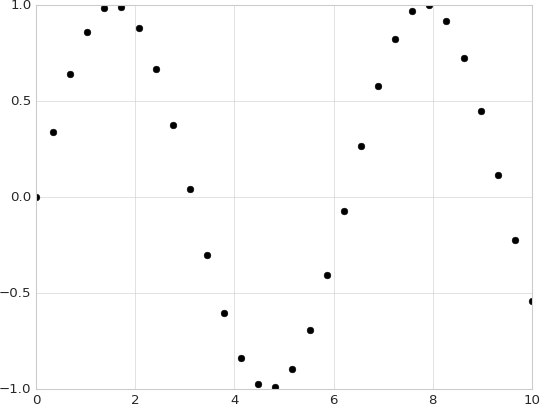
****

การพลอตแบบ **Scatter**

**import numpy as np**

**x = np.linspace(0, 10, 30) y = np.sin(x)**

**plt.plot(x, y, 'o', color='black') plt.show()**

****

**import numpy as np**

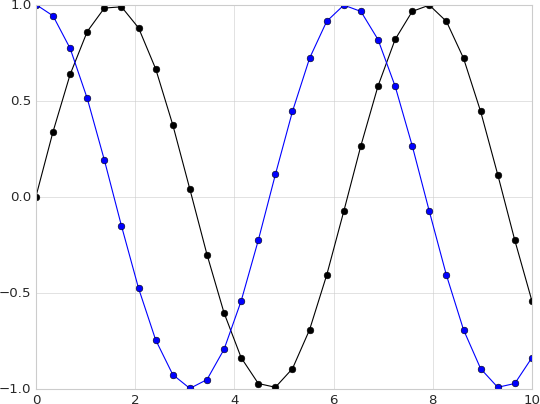
**x = np.linspace(0, 10, 30)**

**y = np.sin(x)**

**z = np.cos(x)**

**plt.plot(x, y, '-ok')** # line (-), circle marker (0), black (k)

**plt.plot(x, z, '-o', color='blue') plt.show()**

****

**import numpy as np**

**x = np.linspace(0, 10, 30)**

**y = np.sin(x)**

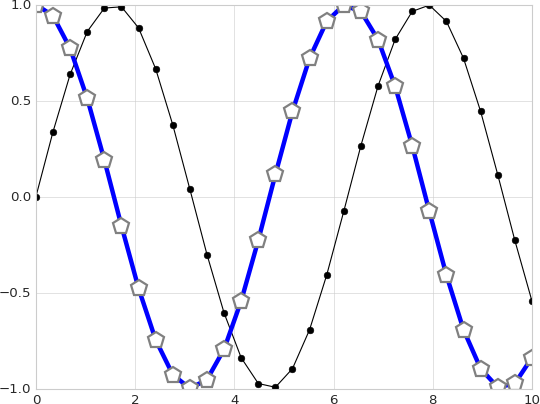
**z = np.cos(x)**

**plt.plot(x, y, '-ok')** # line (-), circle marker (0), black (k)

**plt.plot(x, z, '-pb',**

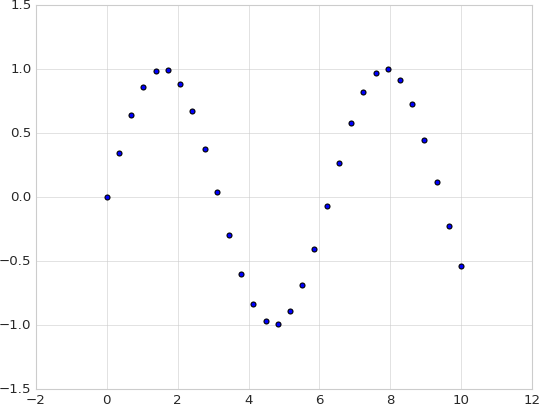
**markersize=15, linewidth=4, markerfacecolor='white', markeredgecolor='gray', markeredgewidth=2)**

**plt.show()**

****

**x = np.linspace(0, 10, 30) y = np.sin(x)**

**plt.scatter(x, y, marker='o') plt.show()**

****

การตกแตทง **Scatter**

**rng = np.random.RandomState(0) no\_plot = 50**

**x = rng.randn(no\_plot) y = rng.randn(no\_plot)**

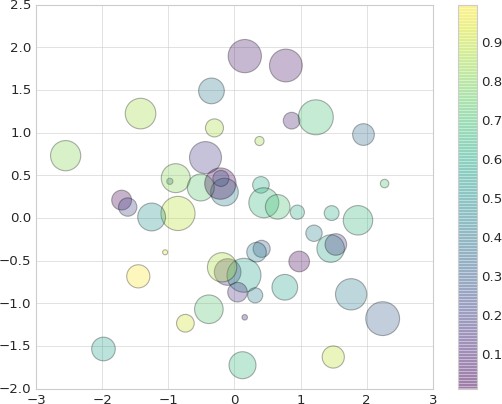
**colors = rng.rand(no\_plot)**

**sizes = 1000 \* rng.rand(no\_plot)**

**plt.scatter(x, y, c=colors, s=sizes, alpha=0.3,**

**cmap='viridis')**

**plt.colorbar() plt.show()**

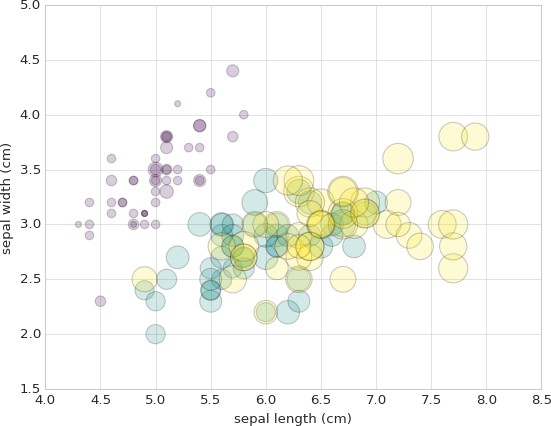
****

**from sklearn.datasets import load\_iris iris\_dataset = load\_iris()**

**features = iris\_dataset.data.T** # T = Matrix Transpose

**plt.scatter(features[0], features[1], alpha=0.2, s=300\*features[3], c=iris\_dataset.target, cmap='viridis')**

**plt.xlabel(iris\_dataset.feature\_names[0]) plt.ylabel(iris\_dataset.feature\_names[1]) plt.show()**

****

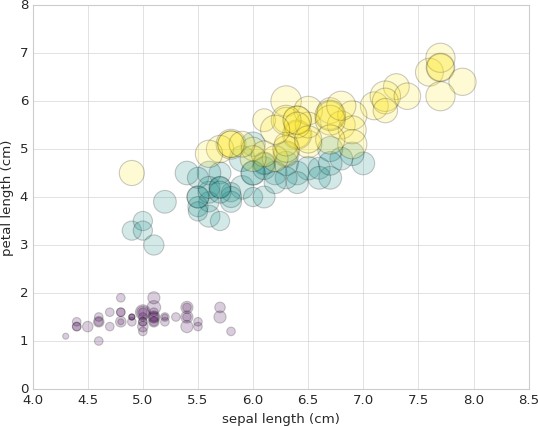
**from sklearn.datasets import load\_iris iris\_dataset = load\_iris()**

**features = iris\_dataset.data.T # T = Matrix Transpose**

**plt.scatter(features[0], features[2], alpha=0.2, s=300\*features[3], c=iris\_dataset.target, cmap='viridis')**

**plt.xlabel(iris\_dataset.feature\_names[0]) plt.ylabel(iris\_dataset.feature\_names[2])**

**plt.show()**

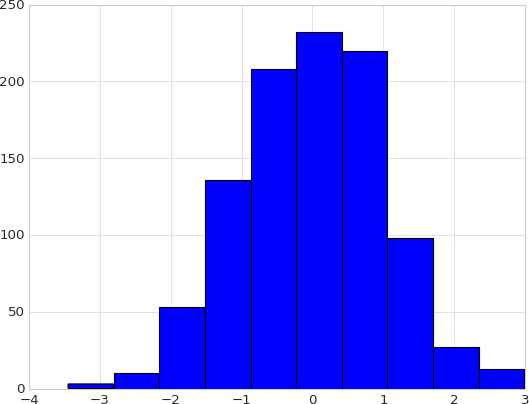
****

การพลอต **Histograms, Binnings** และ **Density**

**import numpy as np**

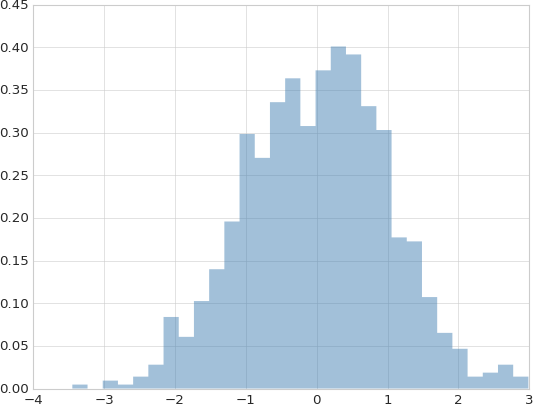
**import matplotlib.pyplot as plt data = np.random.randn(1000)**

**# simple histogram plt.hist(data) plt.show()**

****

**plt.hist(data, bins=30, normed=True, alpha=0.5, histtype='stepfilled', color='steelblue', edgecolor='none')**

**plt.show()**

****

# over-plotting multiple histograms

**x1 = np.random.normal(0, 0.8, 1000)**

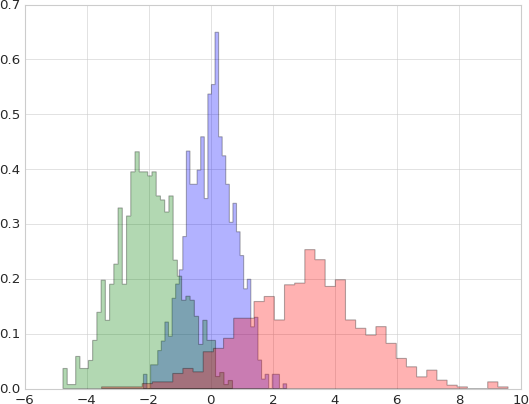
**x2 = np.random.normal(-2, 1, 1000)**

**x3 = np.random.normal(3, 2, 1000)**

**kwargs = dict(histtype='stepfilled', alpha=0.3,**

**normed=True, bins=40)**

**plt.hist(x1, \*\*kwargs) plt.hist(x2, \*\*kwargs) plt.hist(x3, \*\*kwargs) plt.show()**

****

การคาจ นวณคทา **Histogram**

**counts, bin\_edges = np.histogram(data, bins=5) print(counts)**

**counts, bin\_edges = np.histogram(data, bins=7) print(counts)**

[ 33 202 534 219 12]

[ 14 65 231 397 224 65 4]

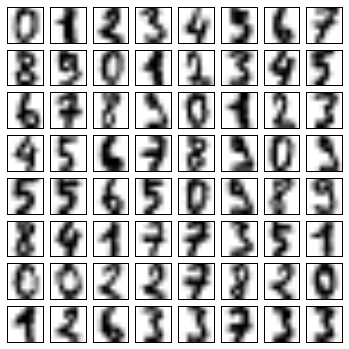
การพลอตชชดขขอมมล **MNSIT** ทททอยมทใน **scikit-learn**

การพลอตชกดขตอมมล MNSIT ทพธเรธยกใชตจากโปรแกรม scikit-learn สามารถทนาไดตดต งนตธ

**from sklearn import datasets digits = datasets.load\_digits()**

**fig, ax = plt.subplots(8, 8, figsize=(6,6)) for i, axi in enumerate(ax.flat):**

**axi.imshow(digits.images[i], cmap='binary') axi.set(xticks=[], yticks=[])**

****

การพลอตชชดขขอมมล **MNIST** ในรมปแบบ **2** มวตวโดยใชขววธท**IsoMap**

**from sklearn import datasets**

**digits = datasets.load\_digits(n\_class=5)**

ตต วอยพางขตางตตน แสดงววธธการดทงขตอมมล MNIST โดยกนาหนดใหต n\_class=5 นตพ นหมายถทงโหลดเฉพาะ class 0-4 เทพานต ตน

**from sklearn.manifold import Isomap iso = Isomap(n\_components=2)**

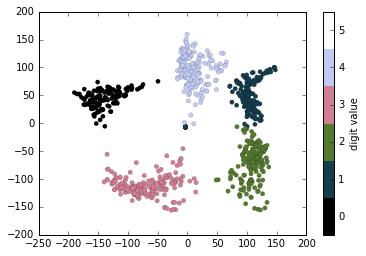
**projection = iso.fit\_transform(digits.data)**

จากนต ตนเรธยกใชตเมธอด Isomap เพพพอคนานวณหาคพาสหสตมพต นธลโดยกนาหนดใหต components = 2

**plt.scatter(projection[:, 0], projection[:, 1], lw=0.1, c=digits.target, cmap=plt.cm.get\_cmap('cubehelix', 6))**

**plt.colorbar(ticks=range(6), label='digit value') plt.clim(-0.5, 5.5)**

**plt.show()**

****

บทททท **5** ไลบรารทท**Seaborn (Seaborn Library)**

Seaborn Library เปก นไลบรารพธทพธใชตงานรพวมกต บ Python เพพพอใชตแสดงขตอมมลทางสถวตว (statistical data visualization) การตวดตต งต ไลบรารพธ Seaborn ทนาไดตโดยเปว ดโปรแกรม Terminal และพวมพลคนาสตพงดต งตพอไปนตธ

**$ pip install seaborn**

หากตตองการตวดตต งต ผพาน Jupyter สามารถทนาไดตโดย

**! pip install seaborn**

ไลบรารพธ seaborn จะทนางานควบคกพกต บไลบรารพธ Pandas ดต งนต ตน หากยต งไมพไดตตวดตต งต Pandas

สามารถทนาไดตโดยใชตคนาสตพงดต งตพอไปนตธ

**$ pip install pandas**

เมพพอตวดตต งต seaborn เปกนทพธเรธยบรตอย สามารถทดสอบโดยอวมพอรลตไลบรารพธ seaborn มาใชต งาน ดต งนตธ

**import seaborn as sns**

ทดสอบการใชตไลบรารพธ seaborn โดยเรธยกใชตชกดขตอมมล Iris ทนาไดตดต งตพอไปนตธ

**import seaborn as sns**

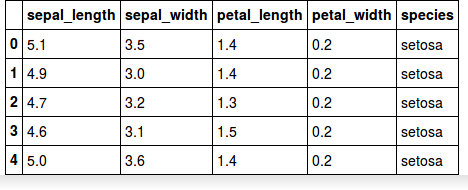
**iris\_dataset = sns.load\_dataset('iris') iris\_dataset.head()**

เมพพอโหลดชกดขตอมมล iris เสรกจสวตน ตต วแปร iris\_dataset จะถมกจต ดเกกบใหตอยมพในรมปแบบของ

pandas DataFrame หากตตองการดมขตอมมลทพธอยมพใน DataFrame สามารถใชตคนาสตพง head() เพพพอดมราย

ละเอธยด โดยคนาสตพง head() จะแสดงตต วอยพางจนานวน 5 ตต วอยพางเทพานต ตน ผลลต พธลทพธไดตจากการใชตคนาสตพง

iris\_dataset.head() แสดงดต งตพอไปนตธ



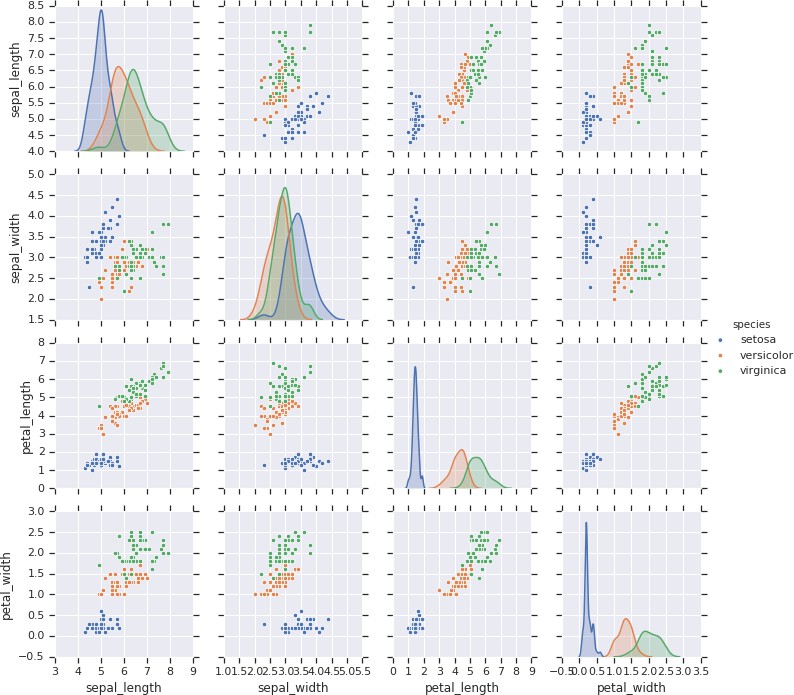
การแสดงชชดขขอมมล **Iris** แบบ **Visualization**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**sns.set()**

**sns.pairplot(iris\_dataset, hue='species', size=2.5) plt.show()**

****

Fundamentals of Machine Learning and Analyzing Data with Python 59

การทจางานรทวมกตนระหวทาง **seaborn** และ **scikit-learn**

ในกรณธ ทพธใชต seaborn โหลดขตอมมลมาใชตงาน ขตอมมลจะถมกจต ดเกกบในรมปแบบของ DataFrame ดต งนต ตนหากตตองการดทงขตอมมล (Extract) อารลเรยล feature และ target ออกมาจาก DataFrame สามารถทนาไดตดต งนตธ

**X\_iris = iris\_dataset.drop('species', axis=1)**

**X\_iris.shape**

(150, 4)

จากตต วอยพางขตางตตน คนาสตพง iris\_dataset.drop('species', axis=1) เปกนการลบขตอมมลของ

'species' ในแนวคอลต มนล ซพทงมธทต งต หมด 1 attribute ออกจาก iris\_dataset

**y\_iris = iris\_dataset['species'] y\_iris.shape**

(150,)

จากตต วอยพางขตางตตน เลพอกคอลต มนล 'species' ซพทงมธทต งต หมด 1 attribute เพพพอใชตเปกนขตอมมล y

หรพอ label

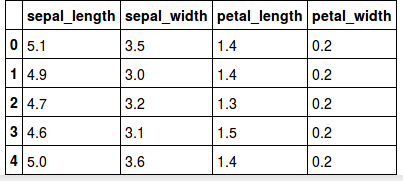
หากตตองการดมประเภทของขตอมมลสามารถใชตคนาสตพง type() แสดงดต งตต วอยพางตพอไปนตธ

**type(X\_iris)**

pandas.core.frame.DataFrame

จากตต วอยพางขตางตตนแสดงใหตเหกนวพาตต วแปร X\_iris เปกนขตอมมลประเภท DataFrame ดต งนต ตน สามารถใชตคนาสตพง head() เพพพอดมขตอมมลของ X\_iris

**X\_iris.head()**

****

หากตตองการเรธยกดมขตอมมลทพธละสพวน (Slice) สามารถทนาไดตดต งตพอไปนตธ

**X\_iris.iloc[1]** # row 1

sepal\_length 4.9

sepal\_width 3.0

petal\_length 1.4

petal\_width 0.2

Name: 1, dtype: float64

สามารถใชต index ในการเรธยกดมขตอมมล ซพทงคลตายกต บอารลเรยล จากตต วอยพางตพอไปนตธ คพอการเรธยกดม ขตอมมลทพธ row = 0 และ col = 0

**X\_iris.iloc[0,0]** # row 0, col 0

5.1

หากตตองการเรธยกดมขตอมมลในทกก ๆ row ทพธ col = 1 สามารถทนาไดตโดย

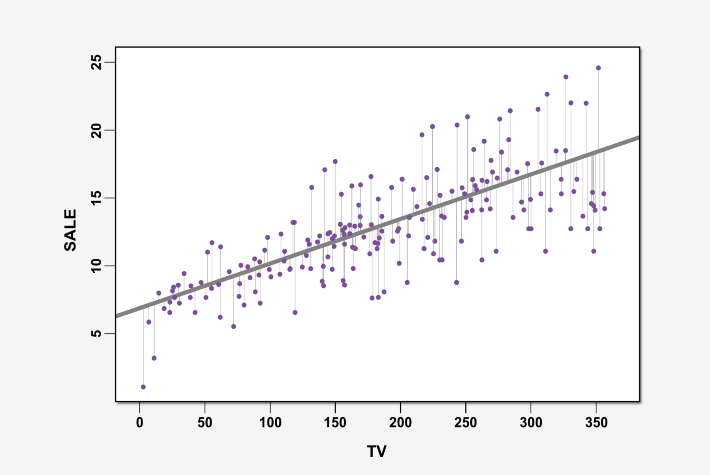
**X\_iris.iloc[:,1]**

|  |  |
| --- | --- |
| 0 | 3.5 |
| 1 | 3.0 |
| 2 | 3.2 |
| 3 | 3.1 |
| 4 | 3.6 |
| 5 | 3.9 |
|  | ... |
| 144 | 3.3 |
| 145 | 3.0 |
| 146 | 2.5 |
| 147 | 3.0 |
| 148 | 3.4 |
| 149 | 3.0 |
| Name: | sepal\_width, dtype: float64 |

บทททท **6** การววเคราะหหการถดถอยเชวงเสขน **(Linear Regression)**

การววเคราะหลการถดถอยเชวงเสตน (Linear Regression) เปกนการคนานวณหาความสตมพต นธล ระหวพางตต วแปรตต งต แตพ 2 ตต วแปร โดยเปกนการประมาณการ (Predictor, X) และตต วตอบสนอง (Response, y) ซพทงเปกนความสตมพต นธลแบบเชวงเสตน (Linear) ซพทงเปกนการคนานวณจากคพา X และ y ทพธ มธความสตมพต นธลกต น เพพพอคนานวณออกมาเปกนสมการความสตมพต นธล สมการของ Linear Regression แสดงดต งตพอไปนตธ

*y*=*ax*+*b*

**

ภาพประกอบททท **9:** แสดงสมการถดถอยเชวงเสตน Linear Regression

ทพธมา : <https://goo.gl/QL6rCo>

การจาจ ลองชชดขขอมมลเพทพอใชขในการคาจ นวณ **Linear Regression**

**import matplotlib.pyplot as plt import numpy as np plt.style.use('seaborn-whitegrid')**

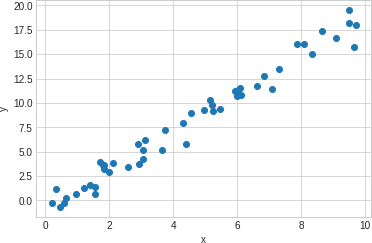
# create random data

**rng = np.random.RandomState(42) x = 10 \* rng.rand(50)**

**y = 2 \* x - 1 + rng.randn(50)**

**plt.scatter(x,y) plt.xlabel('x')**

**plt.ylabel('y') plt.show()**

****

การเรทยกใชขโมดมล **LinearRegression**

การเรธยกใชต Linear Regression ในโปรแกรม scikit-learn สามารถทนาไดตดต งนตธ

**from sklearn.linear\_model import LinearRegression**

จากนต ตนสามารถกนาหนดคพาพารามวเตอรล (Hyperparameter) เชพน

**model = LinearRegression(fit\_intercept=True) print(model)**

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False)

### จตดการขขอมมลเพทพอใชขในการคาจ นวณ

**X = x[:, np.newaxis] print(X.shape)**

(50, 1)

จากตต วอยพางขตางตตนขตอมมลทพธจะนนาไปใชตในการคนานวณ Linear Regression กนาหนดใหตมธ ขนาดเปกน (50, 1) นตพ นหมายถทงมธตต วแปร X เพธยงตต วแปรเดธยว ซพทงเรธยกวพาการคนานวณแบบ Simple Linear Regression หากมธตต วแปร X มากกวพา 1 ตต วแปรจะเรธยกวพา Multiple Linear Regression

**print X[:10]**

|  |  |
| --- | --- |
| [[ | 3.74540119] |
| [ | 9.50714306] |
| [ | 7.31993942] |
| [ | 5.98658484] |
| [ | 1.5601864 ] |
| [ | 1.5599452 ] |
| [ | 0.58083612] |
| [ | 8.66176146] |
| [ | 6.01115012] |
| [ | 7.08072578]] |

หากตตองการแสดงขตอมมลของตต วแปร X สามารถทนาไดตโดยใชตคนาสตพง print และสามารถ Slice ขตอมมลออกมาดมโดยกนาหนด index ทพธตตองการเชพน X[:10] หมายถทง การแสดงขตอมมล X ตต งต แตพ ขตอมมล ลนาดต บทพธ 0 ถทง 10

การเรทยนรขมเพทพอสรขางโมเดล **(Train the model)**

**model.fit(X,y)**

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False)

ในโปรแกรม scikit-learn สามารถใชตคนาสตพง fit เพพพอทพธจะสรตางโมเดลขตทนมาใชตงาน ดต งตต วอยพาง

ขตางตตน

การพยากรณหผลลตพธจห

## Unknown Data)

ากขขอมมลใหมท **(Predict Labels for**

เนพพ องจากตต วอยพางขตางตตนไดตจนาลองขตอมมลเพพพอนนามาใชตในการเรธยนรมตของ Linear Regression ดต งนต ตนในสพวนของการพยากรณลขตอมมลจทงทนาการจนาลองขตอมมลชกดใหมพอธก 1 ชกด เพพพอใชตสนาหรต บการ พยากรณล การจนาลองขตอมมลสามารถทนาไดตดต งนตธ

**xfit = np.linspace(-1, 11)**

# Xfit = unknown data

**Xfit = xfit[:, np.newaxis]**

**print(Xfit.shape) print(Xfit[:10])**

(50, 1)

[[-1. ]

[-0.75510204]

[-0.51020408]

[-0.26530612]

[-0.02040816]

[ 0.2244898 ]

[ 0.46938776]

[ 0.71428571]

[ 0.95918367]

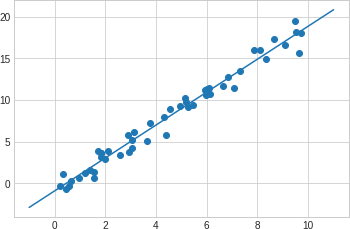
[ 1.20408163]]

ตต วอยพางขตางตตนไดตจนาลองขตอมมลจนานวน 50 จกดเพพพอใชตสนาหรต บการพยากรณล จากนต ตนนนาขตอมมลทพธ ไดตไปพยากรณลดตวยคนาสตพง predict ดต งตต วอยพางตพอไปนตธ

**yfit = model.predict(Xfit)**

ผลการพยากรณลจะถมกเกกบไวตในตต วแปร yfit ดต งนต ตนสามารถนนาผลลต พธลทพธไดตมาพลอตกราฟ เพพพอแสดงเสตน Linear ทพธไดตจากการพยากรณล ดต งตต วอยพางตพอไปนตธ

**plt.scatter(x, y) plt.plot(xfit, yfit) plt.show()**

****

ภาพประกอบททท **10:** เสตน Hyperplane ทพธไดตจากการคนานวณดตวยววธธ

Linear Regression

การพยากรณหขขอมมล **Diabetes** ดขวย **Linear Regression**

อวมพอรลตไลบรารพธทพธจนาเปกนสนาหรต บการทนา Linear Regression

**import matplotlib.pyplot as plt import numpy as np**

**from sklearn import datasets, linear\_model**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

โหลดชกดขตอมมล Diabetes

**diabetes = datasets.load\_diabetes()**

ตรวจสอบขนาดของขตอมมล

**print("shape of the dataset", diabetes.data.shape)**

('shape of the dataset', (442, 10))

ตรวจสอบขตอมมลตต งต แตพ row ลนาดต บทพธ 0 ถทง 5 และ column ลนาดต บทพธ 0 ถทง 5

**diabetes.data[0:5,0:5]**

|  |  |  |
| --- | --- | --- |
| array([[ 0.03807591, | 0.05068012, | 0.06169621, 0.02187235, -0.0442235 ], |
| [-0.00188202, | -0.04464164, | -0.05147406, -0.02632783, -0.00844872], |
| [ 0.08529891, | 0.05068012, | 0.04445121, -0.00567061, -0.04559945], |
| [-0.08906294, | -0.04464164, | -0.01159501, -0.03665645, 0.01219057], |
| [ 0.00538306, | -0.04464164, | -0.03638469, 0.02187235, 0.00393485]]) |

ในกรณธ นตธ คพา X จะถมกกนาหนดใหตใชตขตอมมล diabetes ใน column ทพธ 2

**diabetes\_X = diabetes.data[:, np.newaxis, 2]** # coloumn 2

**diabetes\_X[:10]**

array([[ 0.06169621],

[-0.05147406],

[ 0.04445121],

[-0.01159501],

[-0.03638469],

[-0.04069594],

[-0.04716281],

[-0.00189471],

[ 0.06169621],

[ 0.03906215]])

เนพพ องจากขตอมมล diabetes มธจนานวน 442 instance และไมพไดตแบพงขตอมมลออกเปกนชกดทดสอบ และชกดเรธยนรมต จทงตตองแบพงขตอมมลออกเปกนสองสพวน

# Split the data into training/testing sets **diabetes\_X\_train = diabetes\_X[:-20] diabetes\_X\_test = diabetes\_X[-20:]**

# Split the targets into training/testing sets **diabetes\_y\_train = diabetes.target[:-20] diabetes\_y\_test = diabetes.target[-20:]**

ในกรณธ นตธ ไมพไดตใชตฟต งกลชต น train\_test\_split เพพพอแบพงขตอมมล แตพไดตกนาหนดใหตขตอมมลตต งต แตพ row ลนาดต บทพธ 0 ถทงลนาดต บทพธ 422 เปกนขตอมมลสนาหรต บการเรธยนรมต และลนาดต บทพธ 423 ถทง 442 เปกนขตอมมล สนาหรต บการทดสอบ ดต งนต ตน [:-20] จทงหมายถทงขตอมมลตต งต แตพลนาดต บแรกไปจนถทงลนาดต บสกดทตาย แตพลบดตวย 20 (442-20 = 422)

ขต นต ตอนตพอไป ทนาการสรตางโมเดลดตวยคนาสตพง LinearRegression() และเรธยนรมต (Train) ดตวย ขตอมมลชกดเรธยนรมต ซพทงกกคพอ diabetes\_X\_train

**regr = linear\_model.LinearRegression() regr.fit(diabetes\_X\_train, diabetes\_y\_train)**

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False)

จากนต ตน ทนาการพยากรณลขตอมมลชกดทดสอบ diabetes\_X\_test ดตวยคนาสตพง predict()

**diabetes\_y\_pred = regr.predict(diabetes\_X\_test)**

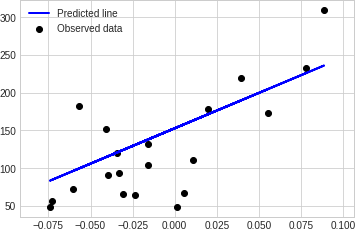
ทนาการพลอตกราฟเพพพอดมขตอมมล และเสตน Hyperplane ทพธไดตจากการคนานวณ

**import matplotlib.pyplot as plt**

**plt.scatter(diabetes\_X\_test, diabetes\_y\_test, color='black') plt.plot(diabetes\_X\_test, diabetes\_y\_pred, color='blue',**

**linewidth=2)**

**plt.legend(['Predicted line', 'Observed data']) plt.show()**

****

ปรต บแตพงกราฟใหตสวยงาม ทนาไดตดต งนตธ

**import numpy as np**

**import matplotlib.pyplot as plt**

**rng = np.random.RandomState(0) colors = rng.rand(20)**

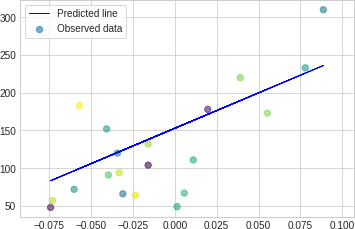
**fig = plt.figure()**

**plt.scatter(diabetes\_X\_test, diabetes\_y\_test, c=colors, s=40, alpha=0.6, cmap='viridis')**

**plt.plot(diabetes\_X\_test, diabetes\_y\_pred, color='blue', linewidth=1)**

**plt.legend(['Predicted line', 'Observed data'], frameon=True, loc='upper left')**

**plt.show()**

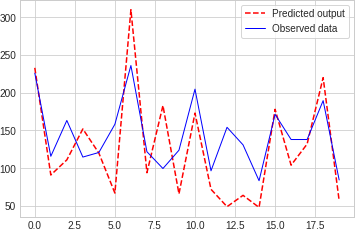
****

หากตตองการเปรธยบเทธยบระหวพาง actual target และ predict target สามารถทนาไดต ดต งนตธ

**plt.plot(diabetes\_y\_test, '--r') plt.plot(diabetes\_y\_pred, '-b', linewidth=1) plt.legend(['Predicted output', 'Observed data'],**

**frameon=True, loc='upper right')**

**plt.show()**

****

ภาพประกอบททท **11:** แสดงการพลอตคพา target จรวง (Actual Target) และ คพา target ทพธไดตจากการพยากรณล (Predicted Target)

สามารถพวมพลคพา coefficients และคพา Mean Squared Error (MSE) ออกมาเพพพอตรวจ สอบไดตดต งนตธ

# The coefficients **print('Coefficients: \n', regr.coef\_)** # The mean squared error

**print("Mean squared error: %.2f"**

**% mean\_squared\_error(diabetes\_y\_test, diabetes\_y\_pred))** # Explained variance score: 1 is perfect prediction **print('Variance score: %.2f' % r2\_score(diabetes\_y\_test,**

**diabetes\_y\_pred))**

('Coefficients: \n', array([ 938.23786125])) Mean squared error: 2548.07

Variance score: 0.47

การพยากรณหขขอมมล **Housing** ดขวย **Linear Regression**

อวมพอรลตไลบรารพธทพธจนาเปกนสนาหรต บการทนา Linear Regression

**import pandas as pd import numpy as np**

**from sklearn import linear\_model**

**from sklearn.cross\_validation import train\_test\_split from sklearn.metrics import mean\_squared\_error, r2\_score**

โหลดขตอมมล boston housing ทพธอยมพใน scikit-learn มาใชตงาน

**from sklearn.datasets import load\_boston** # housing dataset

**boston = load\_boston()**

ตต วแปร boston ถมกจต ดเกกบอยมพในรมปแบบ dictionary สามารถเรธยกดมรายละเอธยดของการจต ด เกกบขตอมมลดตวยคนาสตพง keys()

**print(boston.keys())**

['data', 'feature\_names', 'DESCR', 'target']

หากตตองการดมรายละเอธยดของชกดขตอมมล Boston housing สามารถทนาไดตโดย

**print(boston.DESCR)** หรพอคาจ สตทง **print(boston['DESCR'])**

Boston House Prices dataset

===========================

Notes

Data Set Characteristics:

:Number of Instances: 506

:Number of Attributes: 13 numeric/categorical predictive

:Median Value (attribute 14) is usually the target

:Attribute Information (in order):

* CRIM per capita crime rate by town
* ZN proportion of residential land zoned for lots over 25,000

sq.ft.

* INDUS proportion of non-retail business acres per town
* CHAS Charles River dummy variable (= 1 if tract bounds river; 0

otherwise)

* NOX nitric oxides concentration (parts per 10 million)
* RM average number of rooms per dwelling
* AGE proportion of owner-occupied units built prior to 1940
* DIS weighted distances to five Boston employment centres

town

* RAD index of accessibility to radial highways
* TAX full-value property-tax rate per $10,000
* PTRATIO pupil-teacher ratio by town
* B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by
* LSTAT % lower status of the population
* MEDV Median value of owner-occupied homes in $1000's

:Missing Attribute Values: None

หากตตองการทราบขนาดของชกดขตอมมล Boston housing สามารถทนาไดตโดย

**print("shape of the data", boston.data.shape)** #print("shape of the data", boston['data'].shape) **boston['data'][:2,:] # row = 2, col = all**

('shape of the data', (506, 13))

|  |  |  |  |
| --- | --- | --- | --- |
| array([[ | 6.32000000e-03, | 1.80000000e+01, | 2.31000000e+00, |
|  | 0.00000000e+00, | 5.38000000e-01, | 6.57500000e+00, |
|  | 6.52000000e+01, | 4.09000000e+00, | 1.00000000e+00, |
|  | 2.96000000e+02, | 1.53000000e+01, | 3.96900000e+02, |
|  | 4.98000000e+00], |  |  |
| [ | 2.73100000e-02, | 0.00000000e+00, | 7.07000000e+00, |
|  | 0.00000000e+00, | 4.69000000e-01, | 6.42100000e+00, |
|  | 7.89000000e+01, | 4.96710000e+00, | 2.00000000e+00, |
|  | 2.42000000e+02, | 1.78000000e+01, | 3.96900000e+02, |
|  | 9.14000000e+00]]) |  |  |

ขตอมมลชกด Boston housing มธจนานวน 506 instance และมธ feature ทต งต สวตน 13 attribute หากตตองการดม Label ของขตอมมลสามารถทนาไดตโดยใชตคนาสตพง boston[‘target’] และใชตคนาสตพง shape เพพพอดมขนาดของขตอมมล

**print("shape of the target", boston['target'].shape) boston['target'][:10]**

('shape of the target', (506,))

array([ 24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9])

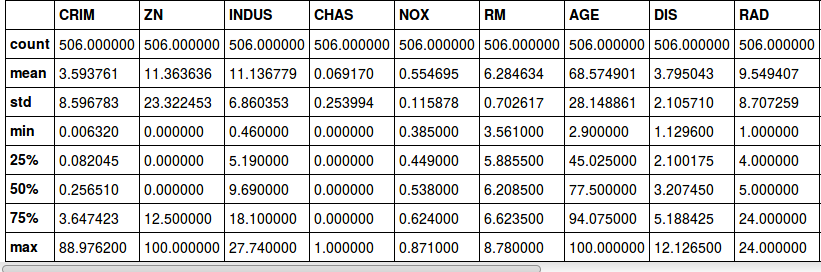
เพพพอชพวยในการแสดงขตอมมลสามารถแปลงขตอมมลใหตอยมพในรมปแบบของ pandas DataFrame

ดต งตพอไปนตธ

**df\_x = pd.DataFrame(boston.data, columns=boston.feature\_names) df\_y = pd.DataFrame(boston.target)**

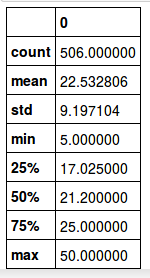
สามารถดมรายละเอธยดของขตอมมลโดยเรธยกผพาน DataFrame ดต งตพอไปนตธ

**df\_x.describe()**

****

ภาพประกอบททท **12:** ตต วอยพางขตอมมลทพธจต ดเกกบในรมปแบบของ pandas DataFrame

**df\_y.describe()**

****

เมพพอเตรธยมขตอมมลเรธยบรตอยแลตว จากนต ตนทนาการสรตางโมเดลของ Linear Regression

**import pandas as pd import numpy as np**

**from sklearn import linear\_model**

**from sklearn.cross\_validation import train\_test\_split**

**from sklearn.datasets import load\_boston** # housing dataset

**boston = load\_boston()**

ในกรณธ นตธ จะเลพอกใชตเพธยง 1 attribute/feature เทพานต ตนเพพพอสรตาง Model ในกรณธ นตธ เลพอกใชต ขตอมมลจาก column ลนาดต บทพธ 5 คพอ RM

# select column 5 = RM

**y = boston['target']**

**X = boston['data']**

**x\_train, x\_test, y\_train, y\_test = train\_test\_split(X[:,5], y, test\_size=0.2, random\_state=4)**

# create a model

**reg.fit(x\_train.reshape(-1, 1), y\_train.reshape(-1, 1))**

# prediction

**y\_pred = reg.predict(x\_test.reshape(-1, 1))**

คนาสตพง fit() คพอการสรตางโมเดล และ predict() คพอการพยากรณล ผลลต พธลทพธไดตจากการ พยากรณลถมกเกกบไวตในตต วแปร y\_pred

**import numpy as np**

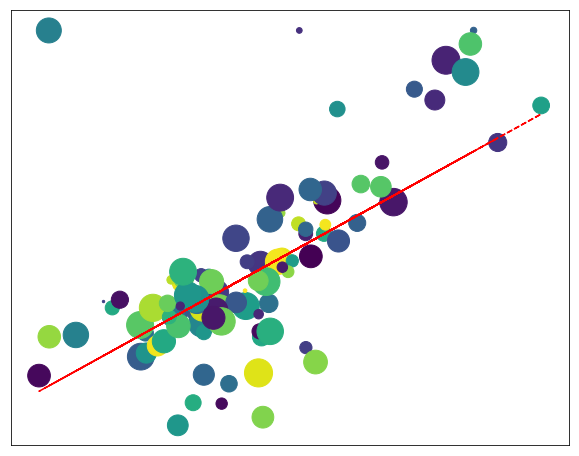
**rng = np.random.RandomState(42) colors = rng.rand(x\_test.shape[0])**

**sizes = 800 \* rng.rand(x\_test.shape[0])**

**plt.figure(figsize=(10,8))**

**plt.scatter(x\_test, y\_test, c=colors, s=sizes) plt.plot(x\_test, y\_pred, '--r')**

**plt.xticks(())**

**plt.yticks(()) plt.show()**

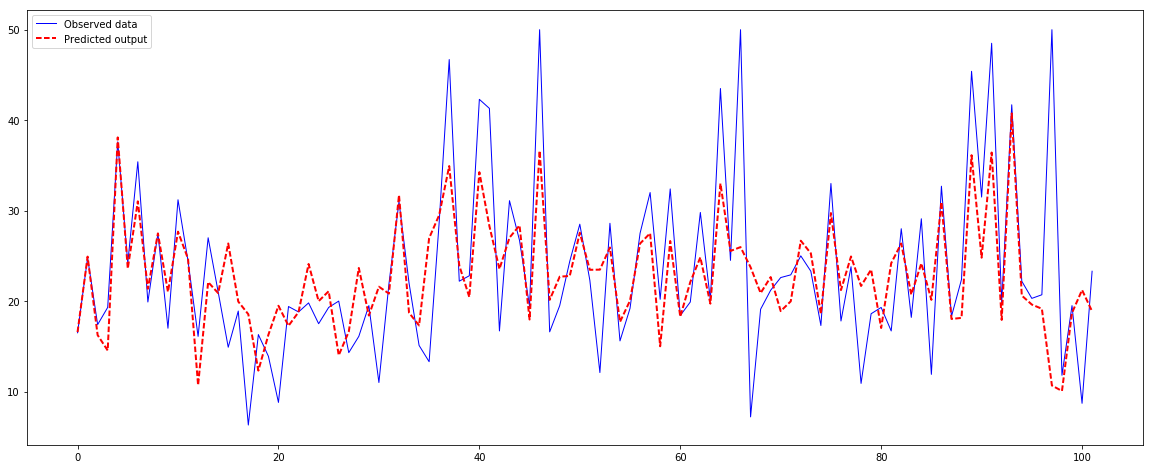
หากตตองการเปรธยบเทธยบระหวพางคพาจรวง (Actual) และคพาทพธพยากรณล (Predicted) สามารถ สรตางกราฟเพพพอแสดงขตอมมลในลต กษณะของ Visualization ไดตดต งนตธ

**import matplotlib.pyplot as plt plt.figure(figsize=(20, 8))**

**plt.plot(y\_test, linewidth=1, linestyle='solid', color='blue', label='Observed data')**

**plt.plot(y\_pred, '--r', linewidth=2, label='Predicted output') plt.legend(frameon=True, loc='upper left')**

**plt.show()**

****

แสดงคพา Coefficients, MSE และ Variance score

# The coefficients **print('Coefficients: \n', reg.coef\_)** # The mean squared error

**print("Mean squared error: %.2f"**

**% mean\_squared\_error(y\_test, y\_pred))**

# Explained variance score: 1 is perfect prediction

**print('Variance score: %.2f' % r2\_score(y\_test, y\_pred))**

('Coefficients: \n', array([[ -1.14743504e-01, 4.70875035e-02, 8.70282354e- 03,

3.23818824e+00, -1.67240567e+01, 3.87662996e+00,

-1.08218769e-02, -1.54144627e+00, 2.92604151e-01,

-1.33989537e-02, -9.07306805e-01, 8.91271054e-03,

-4.58747039e-01]]))

Mean squared error: 25.41 Variance score: 0.73

ในกรณธ ตพอไปจะใชต column ลนาดต บทพธ 2 คพอ INDUS ในการสรตางโมเดล

**X = boston['data']**

**x\_train, x\_test, y\_train, y\_test = train\_test\_split(X[:,2], y, test\_size=0.2,**

**random\_state=4)**

# create a model

**reg.fit(x\_train.reshape(-1, 1), y\_train.reshape(-1, 1))**

# prediction

**y\_pred = reg.predict(x\_test.reshape(-1, 1))**

จากนต ตนพลอตกราฟ เพพพอแสดงในลต กษณะของการ Visualization

**import numpy as np**

**import matplotlib.pyplot as plt**

# Plot outputs

**rng = np.random.RandomState(42) colors = rng.rand(x\_test.shape[0])**

**sizes = 800 \* rng.rand(x\_test.shape[0])**

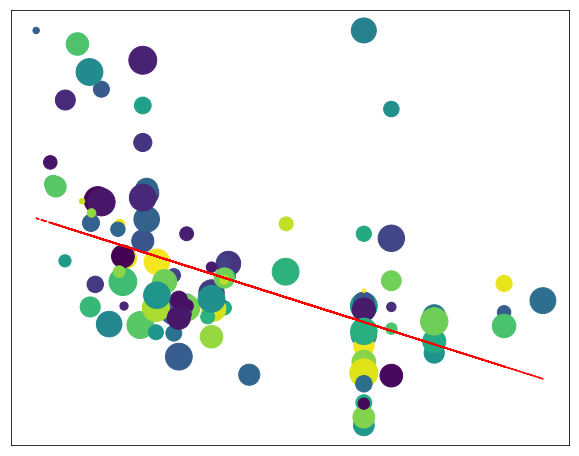
**plt.figure(figsize=(10,8))**

**plt.scatter(x\_test, y\_test, c=colors, s=sizes) plt.plot(x\_test, y\_pred, '--r')**

**plt.xticks(())**

**plt.yticks(())**

**plt.show()**

****

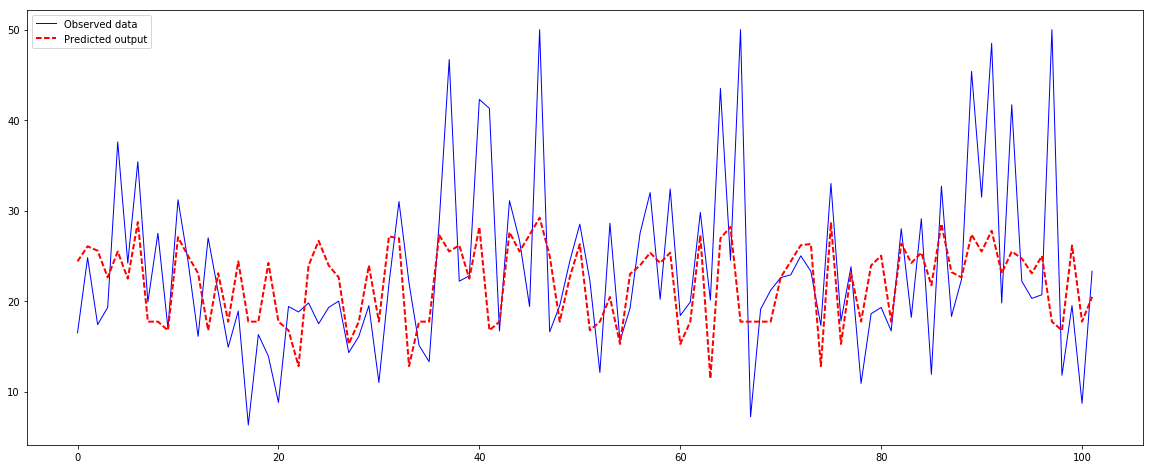
**import matplotlib.pyplot as plt**

**plt.figure(figsize=(20, 8))**

**plt.plot(y\_test, linewidth=1, linestyle='solid', color='blue', label='Observed data')**

**plt.plot(y\_pred, '--r', linewidth=2, label='Predicted output') plt.legend(frameon=True, loc='upper left')**

**plt.show()**

****

# The coefficients **print('Coefficients: \n', reg.coef\_)** # The mean squared error

**print("Mean squared error: %.2f"**

**% mean\_squared\_error(y\_test, y\_pred))**

# Explained variance score: 1 is perfect prediction

**print('Variance score: %.2f' % r2\_score(y\_test, y\_pred))**

('Coefficients: \n', array([[-0.65143285]])) Mean squared error: 75.47

Variance score: 0.19

สรชปผลการทดลองดขวยววธท**Liner Regression** กตบขขอมมลชชด

## Boston housing

จากการทดลองทต งต 2 กรณธ คพอใชตคอลต มนล RM และ INDUS ปรากฎวพา

**RM** มธคพา MSE 25.41 และ Variance score 0.73

**INDUS** มธคพา MSE 75.47 และ Variance score 0.19

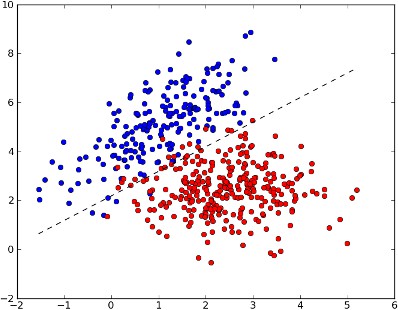
โดยทพธคพา MSE หากเขตาใกลต 0 จะมธคพา Error นตอยทพธสกด และคพา Variance score หากมธคพา เปกน 1 จะเปกนการพยากรณลทพธแมพนยนาทพธสกด จทงสรกปไดตวพา หากใชตคอลต มนล RM และ INDUS ในการ พยากรณลขตอมมล Boston housing คอลต มนล RM มธประสวทธวภาพในการพยากรณลสมงกวพา ดต งนต ตน หาก ตตองการนนาไปใชตในการสรตางโมเดล จทงแนะนนาใหตใชตคอลต มนล RM ในการสรตาง ทต งต นตธ อาจจะตตองลอง ทนาการสรตางโมเดลดตวยววธธอพพน เชพน Multiple Linear Regression เพพพอทดสอบประสวทธวภาพตพอไป

ตตวจาจ

บทททท **7** แนกแบบไบนารทท

# (ฺBinary Classifier)

ตต วจนาแนกแบบไบนารพธ (Binary Classifier) เปกนววธธการแบพงกลกพมขตอมมลออกเปกนสองกลกพม (Binary Class) จากตต วอยพางขตอมมลแบพงออกเปกนสองกลกพม (สธนนตาเงวน และสธแดง) โดยตต วจนาแนกแบบ ไบนารธจะสรตางเสตนสมมกตวหรพอเรธยกวพา hyperplane เพพพอใชตสนาหรต บการแบพงกลกพมขตอมมลทต งต สองกลกพม ออกจากกต น



ภาพประกอบททท **13:** แสดงเสตน Hyperplane ทพธใชตแบพงขตอมมลออกเปกนสองสพวน

ในกรณธ นตธ จะทดสอบการจนาแนกแบบไบนารธ โดยใชตขตอมมล MNSIT ในการทดสอบ

**from scipy.io import loadmat**

**mnist\_raw = loadmat("mldata/mnist-original.mat") mnist = {**

**"data": mnist\_raw["data"].T,**

**"target": mnist\_raw["label"][0],**

**"COL\_NAMES": ["label", "data"],**

**"DESCR": "mldata.org dataset: mnist-original",**

**}**

**X,y = mnist['data'], mnist['target']**

จากตต วอยพางขตางตตน ใชตโปรแกรม scipy ในการโหลดไฟลลนามสกกล .mat ซพทงบรรจกดตวยชกด ขตอมมล MNSIT โดยขตอมมลสามารถดาวนลโหลดไดตจาก

<https://github.com/amplab/datascience-sp14/blob/master/lab7/mldata/mnist-> [original.mat](https://github.com/amplab/datascience-sp14/blob/master/lab7/mldata/mnist-original.mat)

เมพพอโหลดขตอมมลเสรกจเรธยบรตอยสามารถเรธยกดมรายละเอธยดของขตอมมลไดตดต งนตธ

# 70K images, 28x28 pixels/image, each pixel = 0 (white) to 255 (black)

**mnist** # a dict object

{'COL\_NAMES': ['label', 'data'],

'DESCR': 'mldata.org dataset: mnist-original', 'data': array([[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

...,

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0],

[0, 0, 0, ..., 0, 0, 0]], dtype=uint8),

'target': array([0., 0., 0., ..., 9., 9., 9.])}

ขตอมมลของตต วเลขจะถมกจต ดเกกบอยมพในรมปแบบของเวกเตอรล ทพธมธขนาด 754 attribute (28x28)

ดต งนต ตน จทงกนาหนดใหตตต วแปร X เกกบขตอมมล Feature และ y เกกบ label ทพธมธจนานวนทต งต สวตน 70,000 ตต ว

**X,y = mnist['data'], mnist['target']**

**X.shape, y.shape**

((70000, 784), (70000,))

จากนต ตนทนาการแยกชกดขตอมมลออกเปกน 2 ชกด คพอ Training set และ Test set

**import numpy as np**

**X\_train, X\_test, y\_train, y\_test = X[:60000], X[60000:], \ y[:60000], y[60000:]**

**shuffle\_index = np.random.permutation(60000)**

**X\_train, y\_train = X\_train[shuffle\_index], \ y\_train[shuffle\_index]**

**print(X\_train.shape, X\_test.shape, y\_train.shape, \ y\_test.shape)**

((60000, 784), (10000, 784), (60000,), (10000,))

ชกดขตอมมล MNIST ทพธโหลดมาไดตแบพงขตอมมลเปกนทพธเรธยบรตอย โดยขตอมมลชกดทพธ 1 – 60000 เปกน ชกดเรธยนรมต โดยขตอมมลจะเรธยงจาก กลกพม (Class) 0 ถทง Class 9 และตต งต แตพ 60001 – 70000 เปกนชกด ทดสอบ ขตอมมลจะเรธยงลนาดต บจาก Class 0 ถทง Class 9 เชพนกต น ดต งนต ตน สามารถใชตคนาสตพงตพอไปนตธ สนาหรต บ แบพงขตอมมล

**X\_train, X\_test, y\_train, y\_test = X[:60000], X[60000:], \ y[:60000], y[60000:]**

## Stochastic Descent

เรพวมตตนดตวยการกนาหนดขตอมมลใหตเปกน 2 กลกพม (Class) โดยตต วอยพางกนาหนดใหตกลกพมของตต วเลข

5 มธคพาเปกน True และกลกพมตต วเลขตต งต แตพ 0-4 และ 6-9 มธคพาเปกน False

**y\_train\_5 = (y\_train == 5) # create target vectors y\_test\_5 = (y\_test == 5)**

**print(y\_train\_5.shape, y\_train\_5) print(y\_test\_5.shape, y\_test\_5)**

((60000,), array([False, False, False, ..., False, False, False]))

((10000,), array([False, False, False, ..., False, False, False]))

สรขางโมเดล **Stochastic Descent**

สนาหรต บ scikit-learn ใหตเรธยกใชตคนาสตพง SGDClassifier() เพพพอกนาหนด hyperparameter ทพธ จนาเปกน และใชตคนาสตพง fit() เพพพอสรตางโมเดล

**from sklearn.linear\_model import SGDClassifier**

**sgd\_clf = SGDClassifier(random\_state=42) sgd\_clf.fit(X\_train, y\_train\_5)**

SGDClassifier(alpha=0.0001, average=False, class\_weight=None, epsilon=0.1, eta0=0.0, fit\_intercept=True, l1\_ratio=0.15, learning\_rate='optimal', loss='hinge', n\_iter=5, n\_jobs=1, penalty='l2', power\_t=0.5, random\_state=42, shuffle=True, verbose=0, warm\_start=False)

การพยากรณหดขวยโมเดล **Stochastic Descent**

สนาหรต บโปรแกรม scikit-learn สามารถใชตคนาสตพง predict() สนาหรต บการพยากรณลขตอมมล ใน กรณธ นตธ ไดตสรตางฟต งกลชต น (Function) ในภาษา Python สามารถประกาศฟต งกลชต นดตวยคนาสตพง def ประกอบดตวย plot\_digit() และ pred\_data() เพพพอใหตเรธยกใชตงานไดตอยพางสะดวก

**def plot\_digit(X): plt.imshow( X.reshape(28, 28), cmap = plt.cm.binary,**

**interpolation="nearest")**

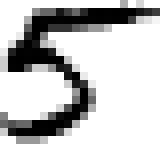
**plt.axis("off") plt.show()**

**def print\_pred\_data(clf, actual\_y, X): print("Actual : ", actual\_y) print("Prediction : ", clf.predict([X])[0])**

ดต งนต ตน การพยากรณลขตอมมลตต วเลข และแสดงรมปภาพของตต วเลขแบบ Visualization สามารถ ทนาไดตดต งตต วอยพางตพอไปนตธ

**pred\_digit = 5500 plot\_digit(X\_test[pred\_digit])**

**print\_pred\_data(sgd\_clf, y\_test\_5[pred\_digit], \ X\_test[pred\_digit])**



('Actual : ', True) ('Prediction : ', True)

จากตต วอยพางขตางตตน ไดตพยากรณลขตอมมลในชกดทดสอบลนาดต บทพธ 5500 (X\_test[5500]) ซพทง Class ทพธแทตจรวง (Actual Class) คพอ True (ตต วเลข 5) และผลการพยากรณล (Prediction Class) ไดตคนาตอบคพอ True นตพ นแสดงวพาเปกนการพยากรณลทพธถมกตตอง

**pred\_digit = 1000 plot\_digit(X\_test[pred\_digit])**

**print\_pred\_data(sgd\_clf, y\_test\_5[pred\_digit], \ X\_test[pred\_digit])**

****

('Actual : ', False) ('Prediction : ', False)

การวตดประสวทธวภาพ **(Performance Measurement)**

สนาหรต บการวตดประสวทธวภาพสามารถทนาไดตหลายววธธ ในตต วอยพางนตธ จะใชตคนาสตพง cross\_val\_score() เพพพอทดสอบและวตดประสวทธวภาพของอต ลกอรวธทม Stochastic Descent แสดง ดต งตต วอยพางตพอไปนตธ

**from sklearn.model\_selection import cross\_val\_score**

**print(cross\_val\_score(**

**sgd\_clf, X\_train, y\_train\_5, cv=3,**

**scoring="accuracy"))**

[0.9617 0.96315 0.95905]

ตต วอยพางขตางตตนใชตคนาสตพง cross\_val\_score() ในการวตดประสวทธวภาพ โดยคพามาตรฐาน (Default) ของคนาสตพงสนาหรต บการวต ดคพาความถมกตตองไดตกนาหนดใหต k-fold มธคพาเทพากต บ 3 (cross validation: cv=3) ดต งนต ตน ผลลต พธลทพธไดตกกคพอ 0.9617, 0.96315 และ 0.95905 โดยเปกนผลลต พธล ของการทดลองครต งต ทพธ 1, 2 และ 3 ตามลนาดต บ

การประเมวนประสวทธวภาพของอตลกอรวธธมดขวย **Confusion Matrix**

การประเมวนประสวทธวภาพของกต ลกอรวธทมดตวย Confusion Matrix เปกนววธธทพธแสดงใหตเหกนถทง ผลลต พธลจากการพยากรณล และทนาใหตรมตไดตวพาหากผลลต พธลผวดพลาด ผวดพลาด ณ จกดใด การทนา Confusion Matrix ใน scikit-learn ใชตคนาสตพง confusion\_matrix() สามารถเรธยกใชตไดตดต งตพอไปนตธ

**from sklearn.metrics import confusion\_matrix**

ในกรณธ นตธ ไดตสรตางฟต งกลชต นเพพพอใชตสนาหรต บการแสดงผลของ Confusion Matrix แบบ

Visualization โดยสรตางฟต งกลชต นชพพอ plot\_confusion\_matrix() แสดงดต งตพอไปนตธ

**import matplotlib.pyplot as plt import itertools**

**def plot\_confusion\_matrix(cm, classes,**

**title='Confusion matrix', cmap=plt.cm.Blues):**

**plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)**

**plt.colorbar()**

**tick\_marks = np.arange(len(classes)) plt.xticks(tick\_marks, classes, rotation=45) plt.yticks(tick\_marks, classes)**

**thresh = cm.max() / 2.**

**for i, j in itertools.product(range(cm.shape[0]), \**

**range(cm.shape[1])): plt.text(j, i, format(cm[i, j], 'd'),**

**horizontalalignment='center',**

**color='white' if cm[i,j] > thresh else 'black')**

**plt.tight\_layout() plt.ylabel('True label') plt.xlabel('Predicted label')**

ในกรณธ นตธ ไดตทดสอบโมเดลของ Stochastic Descent อธกครต งต โดยใชตคนาสตพง

cross\_val\_predict() ววธธการทดสอบ แสดงดต งตพอไปนตธ

**from sklearn.model\_selection import cross\_val\_predict y\_train\_pred = cross\_val\_predict(sgd\_clf, X\_train, \**

**y\_train\_5, cv=3)**

จากตต วอยพาง ทดสอบโดยใชตขตอมมลชกดทดสอบ (Train) เมพพอเสรกจสวตนการทดสอบผลลต พธลทพธไดต จากการพยากรณลจะถมกจต ดเกกบทพธตต วแปร y\_train\_pred จากนต ตนจทงนนาผลลต พธลจากการพยากรณล (Prediction Class) y\_train\_pred และผลลต พธลทพธแทตจรวง (Actual Class) ไปคนานวณเพพพอหา Confusion Matrix ดต งตต วอยพางตพอไปนตธ

**from sklearn.metrics import confusion\_matrix**

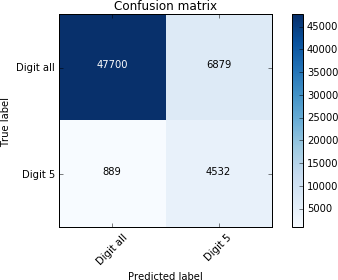
**cm = confusion\_matrix(y\_train\_5, y\_train\_pred) print(cm)**

**plt.figure()**

**classes = ['Digit all', 'Digit 5'] plot\_confusion\_matrix(cm, classes)**

[[47700 6879]

[ 889 4532]]



จากผลลต พธลขตางตตน แบพงออกเปกน 2 แถว และ 2 คอลต มนล เนพพ องจากมธทต งต สวตน 2 กลกพม ขตอมมลทพธแสดงในแนวแถว แสดงถทงกลกพมทพธถมกตตอง (True label) และขตอมมลทพธแสดงในแถวคอลต มนล แสดงถทงผลลต พธลจากการพยากรณล ตต วอยพาง เชพน ในแถวขตอมมลชกด Digit all มธการพยากรณลถมกตตองจนานวน 47700 และพยากรณลผวดเปกน Digit 5 อยมพจนานวน 6879 ตต วเลข และในแถวขตอมมลชกด Digit 5 มธการพยากรณลถมกตตองจนานวน 4532 และพยากรณลผวดเปกน Digit all จนานวน 889 ตต วเลข

จากนต ตนจทงนนาผลลต พธลทพธไดต y\_train\_pred ไปคนานวณรพวมกต บ y\_train\_5 เพพพอคนานวณหาคพา Precision,

Recall และ F1 ดต งตพอไปนตธ

**from sklearn.metrics import precision\_score, recall\_score, \**

**f1\_score**

**print("precision:",precision\_score(y\_train\_5, y\_train\_pred)) print("recall:",recall\_score(y\_train\_5, y\_train\_pred)) print("f1:",f1\_score(y\_train\_5, y\_train\_pred))**

('precision:', 0.3971606344755061)

('recall:', 0.8360081165836561)

('f1:', 0.5384980988593155)

สนาหรต บการทดสอบโดยใชต Training set นต ตนจะเปกนการทดสอบเพพพอปรต บคพาพารามวเตอรล (Tuning Parameter) ใหตเหมาะสมกต บชกดขตอมมล และเพพพอใหตมธประสวทธวภาพสมงสกด จากนต ตนจทงนนา โมเดลทพธไดตไปทดสอบกต บขตอมมลชกด Test set หรพอนนาไปใชตงานจรวงตพอไป

### การนจาโมเดลไปทดสอบกตบขขอมมลชชดทดสอบ

จากตต วอยพางกพอนหนตานตธ ใชตคนาสตพง precision\_score, recall\_score และ fa\_score เพพพอแสดง ผลลต พธล ทต งต นตธ สามารถใชตคนาสตพง classification\_report() เพพพอแสดงผลลต พธลไดตเชพนเดธยวกต น แสดงดต ง ตต วอยพางตพอไปนตธ

**from sklearn.metrics import classification\_report y\_test\_pred = sgd\_clf.predict(X\_test)**

**classes = ['Digit all', 'Digit 5'] print(classification\_report(y\_test\_5, y\_test\_pred, \**

**target\_names=classes))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Digit all | 0.99 | 0.98 | 0.98 | 9108 |
| Digit 5 | 0.80 | 0.85 | 0.83 | 892 |
| avg / total | 0.97 | 0.97 | 0.97 | 10000 |

จากนต ตนใชตเรธยกใชตฟต งกลชต น plot\_confusion\_matrix() เพพพอทนาการ Visualization ขตอมมล

**from sklearn.metrics import confusion\_matrix**

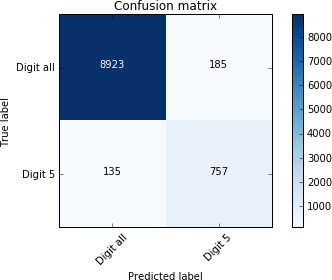
**cm = confusion\_matrix(y\_test\_5, y\_test\_pred) print(cm)**

**plt.figure()**

**classes = ['Digit all', 'Digit 5'] plot\_confusion\_matrix(cm, classes)**

[[8923 185]

[ 135 757]]



เมพพอไดตทนาการพยากรณล Test set จากนต ตนจทงตรวจสอบประสวทธวภาพของอต ลกอรวธทมโดยใชต คนาสตพง accuracy\_score() ดต งตต วอยพางตพอไปนตธ

**from sklearn.metrics import accuracy\_score**

**print("Accuracy Score", accuracy\_score(y\_test\_5, \**

**y\_test\_pred)\*100)**

('Accuracy Score', 96.8)

ผลลต พธลทพธไดตจากการพยากรณลดตวยอต ลกอรวธทม Stochastic Descent มธความถมกตตอง

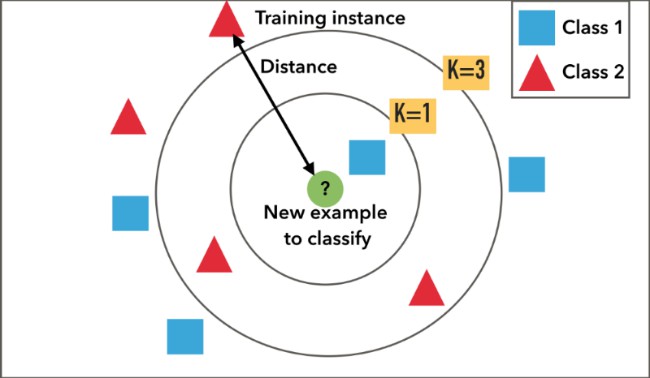
(Accuracy Score/ Accuracy Result) 96.8%

บทททท **8**

การคาจ นวณเพทพอนบขานใกลขทททสชด **k** ตาจ แหนทง

# (ฺK-Nearest Neighbors)

การคนานวณเพพพอนบตานใกลตทพธสกด k ตนาแหนพง (K-Nearest Neighbors) เปกนววธธการเรธยนรมต เครพพองจต กรทพธใชตสนาหรต บจต ดหมวดหมมพขตอมมล (Classification) และเปกนววธธทพธไมพซนาต ซตอน



ภาพประกอบททท **14:** แสดงลต กษณะการทนางานของอต ลกอรวธทม KNN

อต ลกอรวธทมจะทนาการเปรธยบเทธยบจกดใหมพ (New point) กต บจกด (Point) ทต งต หมดทพธอยมพใน Training set เพพพอหาจกดทพธใกลตเคธยงกต บจกดใหมพทพธสกด โดยกนาหนดจนานวนจกดทพธใกลตเคธยงกต บจกดใหมพ จนานวน K จกด เชพน หากกนาหนดใหต K=3 ดต งนต ตน จกดทพธใกลตเคธยงกต บจกดใหมพทต งต สวตนจนานวน 3 จกดจะถมก นนามาพวจารณา จากนต ตนทนาการกนาหนด Label/Class ใหตกต บจกดใหมพ โดยตรวจสอบกต บ Label ของทต งต 3 จกด หากพบวพามธ Label ไหนมากทพธสกด (Majority Vote) กกจะกนาหนดเปกน Label ใหตกต บจกดใหมพนต ตน

จากตตวอยทาง

หากกนาหนดใหต **K=1** จกดทพธใกลตกต บจกดใหมพทพธสกดคพอ Class สพธเหลพธยม ดต งนต ตน จกดใหมพนต ตนจะถมก กนาหนดใหตมธ Class เปกน สพธเหลพธยม

หากกนาหนดใหต **K=3** จกดทพธใกลตกต บจกดใหมพทพธสกดสามลนาดต บคพอ Class สพธเหลพธยม 1 จกด และ

สามเหลพธยม 2 จกด ดต งนต ตนจกดใหมพจะถมกกนาหนดใหตมธ Class เปกน สามเหลพธยม

### การสรขางโมเดลของ **KNN**

ตต วอยพางดต งตพอไปนตธ ทนาการสรตางโมเดลของอต ลกอรวธทม KNN โดยใชตชกดขตอมมล Iris ในการ ทดสอบ

**from sklearn.datasets import load\_iris**

**from sklearn.model\_selection import train\_test\_split**

**iris\_dataset = load\_iris() X\_train, X\_test, y\_train, y\_test =**

**train\_test\_split(iris\_dataset['data'], \**

**iris\_dataset['target'], \ random\_state=0)**

จากตต วอยพางขตางตตน ทนาการโหลดชกดขตอมมล Iris มาเพพพอทดสอบ โดยใชตคนาสตพง load\_iris() จาก นต ตนใชตคนาสตพง train\_test\_split() เพพพอแบพงขตอมมลออกเปกน 2 ชกด โดยขตอมมล Feature/Attribute ทพธแบพง แลตวจะถมกจต ดเกกบไวตทพธตต วแปร X\_train และ X\_test และขตอมมล Label/Class จะถมกเกกบไวตทพธตต วแปร y\_train และ y\_test

จากนต ตนใชตคนาสตพง KNeighborsClassifier() เพพพอกนาหนดคพาพารามวเตอรล n\_neighbors หาก ไมพกนาหนดคพาโปรแกรมจะใชตคพาเรพวมตตน (Default) ซพทงกนาหนดใหตมธคพาเทพากต บ 5

**from sklearn.neighbors import KNeighborsClassifier knn = KNeighborsClassifier(n\_neighbors=1)**

จากตต วอยพางขตางตตนไดตกนาหนดใหต n\_neighbors=1 โดยเกกบคพาทพธกนาหนดไวตทพธตต วแปร knn

จากนต ตนทนาการสรตางโมเดลโดยใชตคนาสตพง fit() ดต งตต วอยพางตพอไปนตธ

**knn.fit(X\_train, y\_train)**

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=1, n\_neighbors=1, p=2, weights='uniform')

### การพยากรณหโดยใชขโมเดลของ **KNN**

ในชกดขตอมมล Iris นต ตนจะมธ Label อยมพจนานวน 3 กลกพม ประกอบดตวย

setosa กลกพม 0

versicolor กลกพม 1

virginica กลกพม 2

ดต งนต ตน Label นตธ จะใชตเปกนตต วตรวจสอบความถมกตตองของการพยากรณล

**print("Test data: {}".format(X\_test[1]))**

**print("Label: {}".format(y\_test[1]))**

Test data: [6. 2.2 4. 1. ]

Label: 1

ตต วอยพางขตางตตนแสดงใหตเหกนขตอมมลทพธจะนนาไปทดสอบ (Test) การพยากรณลดตวยโมเดลของ

KNN

**prediction = knn.predict([X\_test[1]])**

**print("Prediction: {}".format(prediction)) print("Predicted target name: {}".format( \**

**iris\_dataset['target\_names'][prediction]))**

Prediction: [1]

Predicted target name: ['versicolor']

สนาหรต บการพยากรณลจะใชตคนาสตพง predict() จากตต วอยพางไดตสพงขตอมมล X\_test[1] เขตาไป คนานวณ และผลลต พธลทพธไดตจะจต ดเกกบไวตทพธตต วแปร prediction จากการทดลองพบวพา เมพพอสพงขตอมมล X\_test[1] เขตาไปทดสอบ ผลลต พธลทพธไดตคพอ Prediction: [1] คพอกลกพม 1 ซพทงกลกพม 1 กกคพอ versicolor ซพทงเปกนคนาตอบทพธถมกตตอง

ทนาการทดสอบขตอมมล x\_test[2] เพพพอตรวจสอบความถมกตตอง

**prediction = knn.predict([X\_test[2])**

**print("Prediction: {}".format(prediction)) print("Predicted target name: {}".format( \**

**iris\_dataset['target\_names'][prediction]))**

Prediction: [0]

Predicted target name: ['setosa']

จากการทดสอบผลลต พธลทพธไดตคพอกลกพม 0 ซพทงกกคพอกลกพม setosa

### การทดสอบประสวทธวภาพของโมเดล **KNN**

การแสดงผลลต พธลของการจต ดกลกพมขตอมมล (Classification Report) สามารถทนาไดตดต งตต วอยพาง

ตพอไปนตธ

**from sklearn.metrics import classification\_report**

**print(classification\_report(y\_test, y\_pred, target\_names=iris\_dataset['target\_names']))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| setosa | 1.00 | 1.00 | 1.00 | 13 |
| versicolor | 1.00 | 0.94 | 0.97 | 16 |
| virginica | 0.90 | 1.00 | 0.95 | 9 |
| avg / total | 0.98 | 0.97 | 0.97 | 38 |

**from sklearn.metrics import accuracy\_score accuracy\_score(y\_test, y\_pred)**

0.9736842105263158

ในการแสดงผลลต พธลจากการทดลองในโปรแกรม scikit-learn สามารถทนาไดตโดยใชตคนาสตพง เชพน classification\_report() และ accuracy\_score() เปกนตตน จากการทดสอบโมเดลของ KNN มธประสวทธวภาพทพธ 97.37%

การแสดงผลการทดลองดขวย **Confusion Matrix**

**from sklearn.metrics import confusion\_matrix print("Confusion Matrix: \n{}".format(confusion\_matrix( \**

**y\_test, \ y\_pred)))**

Confusion Matrix: [[13 0 0]

[ 0 15 1]

[ 0 0 9]]

ในการแสดงผล Confusion Matrix ใหตอยมพในรมปแบบของการ Visualization สามารถ ทนาไดตโดยการสรตางฟต งกลชต น plot\_confusion\_matrix() ดต งตต วอยพางตพอไปนตธ

**import matplotlib.pyplot as plt import itertools**

**def plot\_confusion\_matrix(cm, classes,**

**title='Confusion matrix', cmap=plt.cm.Blues):**

**plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)**

**plt.colorbar()**

**tick\_marks = np.arange(len(classes)) plt.xticks(tick\_marks, classes, rotation=45) plt.yticks(tick\_marks, classes)**

**thresh = cm.max() / 2.**

**for i, j in itertools.product(range(cm.shape[0]), \**

**range(cm.shape[1])): plt.text(j, i, format(cm[i, j], 'd'),**

**horizontalalignment='center',**

**color='white' if cm[i,j] > thresh else 'black')**

**plt.tight\_layout() plt.ylabel('True label') plt.xlabel('Predicted label')**

ในการคนานวณหาคพา Confusion Matrix สามารถใชตคนาสตพง confusion\_matrix() ดต ง ตต วอยพางตพอไปนตธ

**cm = confusion\_matrix(y\_test, y\_pred) np.set\_printoptions(precision=2)**

**print('Confusion matrix, without normalization') print(cm)**

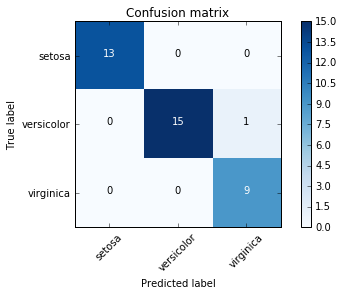
**plt.figure()**

**classes = iris\_dataset.target\_names plot\_confusion\_matrix(cm, classes)**

Confusion matrix, without normalization [[13 0 0]

[ 0 15 1]

[ 0 0 9]]

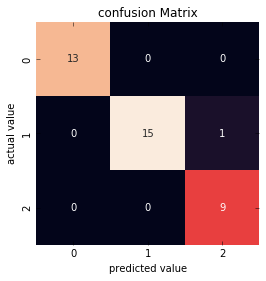


ในการแสดง Confusion Matrix นอกจากจะใชตโปรแกรม matplotlib ยต งสามารถใชต โปรแกรม Seaborn เปกนอธกหนพทงทางเลพอก แสดงดต งตพอไปนตธ

**import seaborn as sns**

**sns.heatmap(cm, square=True, annot=True, cbar=False) plt.title('confusion Matrix')**

**plt.xlabel('predicted value') plt.ylabel('actual value') plt.show()**

****

ตต วอยพาง Classification Score และ Confusion Matrix ขตางตตนไดตมาจากการกนาหนดใหต n\_neighbors = 1 เพพพอทดสอบประสวทธวภาพ สามารถกนาหนดจนานวนของ n\_neighbors ใหตเปกน 3, 5 หรพอ 7 เปกนตตน

การใชขงานอตลกอรวธธม **KNN** กตบขขอมมลโรคเบาหวาน **(Diabetes Dataset)**

ตต วอยพางตพอไปจะนนาอต ลกอรวธทม KNN มาชพวยในการพยากรณลขตอมมลโรคเบาหวาน (Diabetes) วพาผมตปพ วย (patient) เปก นโรคเบาหวาน หรพอไมพ ขตอมมลโรคเบาหวานสามารถดาวนลโหลด ไดตจากเวกบไซตล

<https://www.kaggle.com/amolbhivarkar/knn-for-classification-using-scikit-learn/data>

ตต วอยพางการนนาอต ลกอรวธทม KNN ไปใชตพยากรณลขตอมมลผมตปพ วยโรคเบาหวาน สามารถทนาไดตดต ง

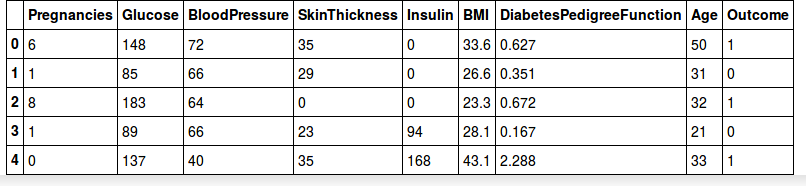
ตพอไปนตธ

**import numpy as np import pandas as pd**

**import matplotlib.pyplot as plt plt.style.use('ggplot')**

**df = pd.read\_csv('mldata/diabetes.csv') df.head()**

จากตต วอยพางขตางตตน ไฟลลทพธดาวนลโหลดจากเวกบไซตลอยมพในรมปแบบของ csv จทงใชตโปรแกรม pandas เพพพอโหลดขตอมมลและแปลงใหตอยมพในรมปแบบของ DataFrame โดยเกกบไวตทพธตต วแปร df หาก ตตองการแสดงขตอมมลโรคเบาหวานสามารถทนาไดตโดยพวมพลคนาสตพง head()



ขตอมมลโรคเบาหวานประกอบดตวย 8 Attribute และผลลต พธลทพธไดตมธคพาเปกน 0 หรพอ 1 ซพทงอยมพใน Attribute ทพธชพพอ Outcome ซพทงกกคพอ Output/Label/Class และหากตตองการทพธจะทราบจนานวนของ ขตอมมลสามารถทนาไดตโดยใชตคนาสตพง shape

**df.shape**

(768, 9)

ขตอมมลโรคเบาหวานทพธใชตในการทดลองประกอบดตวย 768 instance แตพละ instance มธ จนานวน 8 attribute และอธก 1 attribute เปกนผลลต พธล

กนาหนดขตอมมลทพธจะนนาไปใชตในอต ลกอรวธทม KNN สามารถทนาไดตโดย

**from sklearn.model\_selection import train\_test\_split**

**X = df.drop('Outcome',axis=1).values y = df['Outcome'].values X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.4, \**

**random\_state=42, stratify=y)**

จากตต วอยพางขตอมมลทพธเปกน Feature จะถมกเกกบไวตทพธตต วแปร X และขตอมมลของ Class จะถมกเกกบ ไวตทพธตต วแปร y จากนต ตนใชตคนาสตพง train\_test\_split() เพพพอแบพงขตอมมลเปกนชกดเรธยนรมต และชกดทดสอบ โดย กนาหนดใหตขนาดของชกดทดสอบมธขนาดเปกน 40% โดยการกนาหนด test\_size=0.4

**from sklearn.neighbors import KNeighborsClassifier neighbors = np.arange(1,9)**

**train\_accuracy =np.empty(len(neighbors)) test\_accuracy = np.empty(len(neighbors))**

**for i,k in enumerate(neighbors):**

#Setup a knn classifier with k neighbors

**knn = KNeighborsClassifier(n\_neighbors=k)**

#Fit the model

**knn.fit(X\_train, y\_train)**

#Compute accuracy on the training set

**train\_accuracy[i] = knn.score(X\_train, y\_train)**

#Compute accuracy on the test set

**test\_accuracy[i] = knn.score(X\_test, y\_test)**

จากตต วอยพางจะทนาการทดสอบประสวทธวภาพของอต ลกอรวธทมโดยกนาหนดจนานวนของ

n\_neighbors โดยใชตคนาสตพง neighbors = np.arange(1,9) ซพทงผลลต พธลทพธไดตจากคนาสตพงนตธ คพอ

array([1, 2, 3, 4, 5, 6, 7, 8])

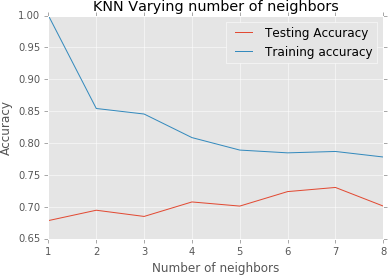
โปรแกรมจะทดสอบทต งต สวตนจนานวน 8 รอบ และเกกบผลลต พธลไวตทพธตต วแปร train\_accuracy และ

test\_accuracy เพพพอใชตในการแสดงผลลต พธล

**plt.title('KNN Varying number of neighbors') plt.plot(neighbors, test\_accuracy, label='Testing Accuracy') plt.plot(neighbors, train\_accuracy, label='Training accuracy') plt.legend()**

**plt.xlabel('Number of neighbors') plt.ylabel('Accuracy')**

**plt.show()**

****

ภาพประกอบททท **15:** กราฟแสดงประสวทธวภาพของอต ลกอรวธทม KNN เมพพอ เปลพธยนคพาพารามวเตอรล n\_neighbors

จากกราฟขตางตตน ประกอบดตวยขตอมมลจนานวน 2 เสตน เสตนสธแดง แทน Testing Accuracy และเสตนสธนนตาเงวนแทน Training Accuracy จะเหกนความแตกตพางระหวพางการทดสอบกต บ Training set และ Test set ดต งนต ตน ในการ Training ยวพงกนาหนดใหตจนานวน n\_neighbors มากเทพาไหรพผลทพธไดต จากการทดลองจะตพนาลงโดยกนาหนดใหต n\_neighbors=2 จะมธประสวทธวภาพสมงทพธสกด ในทางกลต บกต น เมพพอทดสอบกต บ Test set จนานวนของ n\_neighbors ทพธใหตประสวทธวภาพสมงทพธสกดมธคพาเทพากต บ 7 ดต งนต ตน การกนาหนดคพาพารามวเตอรล n\_neighbors จทงขตทนอยมพกต บขตอมมลทพธนนาไปใชต

สรขางโมเดล **KNN** ดขวยคทา **n\_neighbor** ทททไดขจากการทดลอง

จากการทดสอบทนาใหตไดตคพา n\_neibhbor=7 ซพทงเปกนคพาทพธดธทพธสกดในการทดสอบกต บ Test set ดต งนต ตน หากตตองการทราบประสวทธวภาพของอต ลกอรวธทม KNN จทงตตองทดสอบโดยการสรตางโมเดลอธก ครต งต ดต งตต วอยพางตพอไปนตธ

**knn = KneighborsClassifier(n\_neighbors=7) knn.fit(X\_train,y\_train)**

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski', metric\_params=None, n\_jobs=1, n\_neighbors=7, p=2,

weights='uniform')

จากนต ตนใชตคนาสตพง score() เพพพอแสดงคพาความถมกตตอง (Accuracy) ซพทงผลลต พธลทพธไดตคพอ 73.05%

**print("Accuracy", knn.score(X\_test,y\_test)\*100)**

('Accuracy', 73.05194805194806)

หากตตองการแสดงผลการทดลองดตวย Confusion Matrix สามารถทนาไดตดต งนตธ

**from sklearn.metrics import confusion\_matrix**

**y\_pred = knn.predict(X\_test) confusion\_matrix(y\_test,y\_pred)**

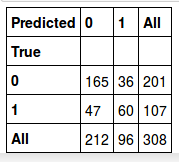
array([[165, 36],

[ 47, 60]])

คาจ สตทง **Pandas Crosstab**

ในโปรแกรม Pandas สามารถใชตคนาสตพง crosstab() เพพพอสรตาง Confusion Matrix ไดต เชพนเดธยวกต น สามารถทนาไดตดต งตต วอยพางตพอไปนตธ

**pd.crosstab(y\_test, y\_pred, rownames=['True'], colnames=['Predicted'], margins=True)**

****

แสดงผลลตพธจห ากการทดลองดขวย **Classification Report**

**from sklearn.metrics import classification\_report print(classification\_report(y\_test,y\_pred))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.78 | 0.82 | 0.80 | 201 |
| 1 | 0.62 | 0.56 | 0.59 | 107 |
| avg / total | 0.73 | 0.73 | 0.73 | 308 |

การทดสอบคทาพารามวเตอรห**(Hyperparameter Tuning)** ดขวยววธท

## Grid Search

จากการทดสอบขตางตตนทนาใหตไดตคพา n\_neighbors=7 แตพการทดสอบสามารถทนาไดตหลายววธธ เชพนววธธ Grid Search เปกนตตน ววธธการทนา Grid Search สามารถทนาไดตดต งนตธ

**from sklearn.model\_selection import GridSearchCV import numpy as np**

**param\_grid = {'n\_neighbors':np.arange(1,50,2)}**

ผลลต พธลทพธไดตจากคนาสตพง np.arange(1,50,2) แสดงดต งตพอไปนตธ

array([ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35,

37, 39, 41, 43, 45, 47, 49])

**knn = KNeighborsClassifier()**

**knn\_cv= GridSearchCV(knn,param\_grid,cv=5) knn\_cv.fit(X,y)**

GridSearchCV(cv=5, error\_score='raise', estimator=KNeighborsClassifier(algorithm='auto', leaf\_size=30,

metric='minkowski',

metric\_params=None, n\_jobs=1, n\_neighbors=5, p=2, weights='uniform'),

fit\_params={}, iid=True, n\_jobs=1,

param\_grid={'n\_neighbors': array([ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49])},

pre\_dispatch='2\*n\_jobs', refit=True, return\_train\_score=True, scoring=None, verbose=0)

เมพพอกระบวนการของการทนา Grid Search เสรกจสวตนสามารถใชตคนาสตพงดต งตพอไปนตธ เพพพอแสดง ผลลต พธล

**knn\_cv.best\_score\_**

0.7552083333333334

**knn\_cv.best\_params\_**

{'n\_neighbors': 13}

ผลลต พธลทพธไดตคพอเมพพอกนาหนดใหตพารามวเตอรล n\_neighbors=13 จะมธความถมกตตองทพธ 75.52%

### การจตดหมวดหมมทชชดขขอมมล **MNIST** ดขวยอตลกอรวธธม **KNN**

ตต วอยพางตพอไปนตธ เปกนการใชตอต ลกอรวธทม KNN เพพพอจต ดหมวดหมมพชกดขตอมมล MNIST โดยใชต โปรแกรม scipy ในการโหลดขตอมมล

**from scipy.io import loadmat**

**mnist\_raw = loadmat("mldata/mnist-original.mat") mnist = {**

**"data": mnist\_raw["data"].T,**

**"target": mnist\_raw["label"][0],**

**"COL\_NAMES": ["label", "data"],**

**"DESCR": "mldata.org dataset: mnist-original",**

**}**

**X, y = mnist['data'], mnist['target']**

จากตต วอยพางขตางตตนใชตคนาสตพง loadmat() ในการโหลดขตอมมลนามสกกล .mat และเกกบขตอมมลไวตทพธ ตต วแปร X และ Label ไวตทพธตต วแปร y ในกรณธ นตธ จะทนาการสลต บขตอมมล (shuffle) เพพพอใหตขตอมมลทพธจะนนา มา Visualization มธความหลากหลาย

**import numpy as np**

**shuffle\_index = np.random.permutation(70000) X, y = X[shuffle\_index], y[shuffle\_index]**

จากตต วอยพางทนาการสลต บขตอมมลทต งต หมด 70,000 ชกด ดตวยคนาสตพง permutation() โปรแกรมจะ สลต บและคพนคพาตนาแหนพง (index) เกกบไวตในตต วแปร shuffle\_index

**X\_train, X\_test, y\_train, y\_test = X[:20000], X[69000:], y[:20000], y[69000:]**

**print(X\_train.shape, X\_test.shape, y\_train.shape, \ y\_test.shape)**

((20000, 784), (1000, 784), (20000,), (1000,))

จากนต ตน เลพอกขตอมมลเพพพอใชตในการเรธยนรมต (X\_train) จนานวน 20,000 ชกด (x[:20000]) และ เลพอกขตอมมลเพพพอใชตในการทดสอบ (X\_test) จนานวน 1,000 ชกด (x[69000:]) โดยรมปภาพตต วเลขมธ ขนาด 28x28 จทงทนาใหต Feature ทพธใชตมธทต งต สวตน 784 attribute

ในความเปกนจรวงขตอมมล mnist-origianl.mat จะกนาหนดใหตขตอมมลชกดทพธ 1-60000 ใชตสนาหรต บ เรธยนรมต และชกดทพธ 60001-70000 ใชตสนาหรต บทดสอบ

## KNN Classifier

ในขต นต ตอนนตธ ทนาการเลพอกใชตโมเดล KNN เพพพอทดสอบกต บชกดขตอมมล MNIST โดยมธขตอมมล สนาหรต บเรธยนรมตจนานวน 20,000 instance และ ขตอมมลสนาหรต บทดสอบจนานวน 1,000 instance การเลพอกโมเดล KNN และกนาหนดพารามวเตอรล n\_neighbors=1 ทนาไดตดต งตต วอยพางตพอไปนตธ

**from sklearn.neighbors import KNeighborsClassifier**

**knn = KneighborsClassifier(n\_neighbors=1) knn.fit(X\_train, y\_train)**

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski', metric\_params=None, n\_jobs=1, n\_neighbors=1, p=2, weights='uniform')

จากนต ตนจทงสรตางโมเดลดตวยคนาสตพง fit() โมเดลทพธสรตางจะถมกเกกบไวตในตต วแปรทพธชพพอ knn

**from sklearn.metrics import accuracy\_score**

**y\_model = knn.predict(X\_test) accuracy\_score(y\_test, y\_model)**

0.97

จากตต วอยางขตพ างตตน นาโน มเดล knn ไปพยากรณลขตอมมล X\_test ดวยต คาน สตพง predict() และ

แสดงผลความถมกตตองดตวยคนาสตพง accuracy\_score() และมธความถมกตตองทพธ 0.97 หรพอ 97% สามารถ คนานวณคพา และแสดงตาราง Confusion Matrix ไดตดต งตพอไปนตธ

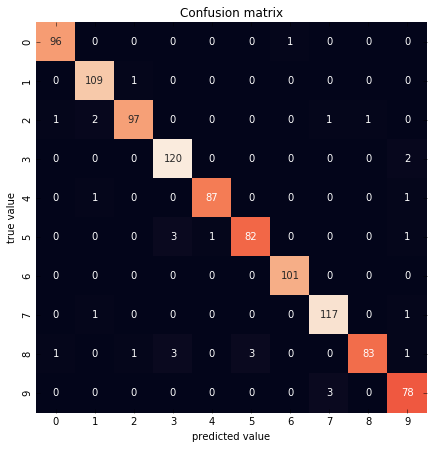
**from sklearn.metrics import confusion\_matrix import seaborn as sns**

**mat = confusion\_matrix(y\_test, y\_model)**

**plt.figure(figsize=(7,7)) plt.title("Confusion matrix")**

**sns.heatmap(mat, square=True, annot=True, cbar=False, fmt="d") plt.xlabel('predicted value')**

**plt.ylabel('true value') plt.show()**



จากตต วอยพาง เสตนทแยงคพอผลลต พธลของโมเดล KNN ทพธพยากรณลถมกตตอง และตต วเลขทพธกระจาย ทตพว Confusion Matrix คพอตต วเลขทพธพยากรณลผวดพลาด

สกดทตายแลตวสามารถแสดงขตอมมลการพยากรณลแบบ Visualization ไดตดต งตพอไปนตธ

**import matplotlib.pyplot as plt**

**fig, axes = plt.subplots(10, 10, figsize=(8, 8),**

**subplot\_kw={'xticks':[], 'yticks':[]}, gridspec\_kw=dict(hspace=0.1, wspace=0.1))**

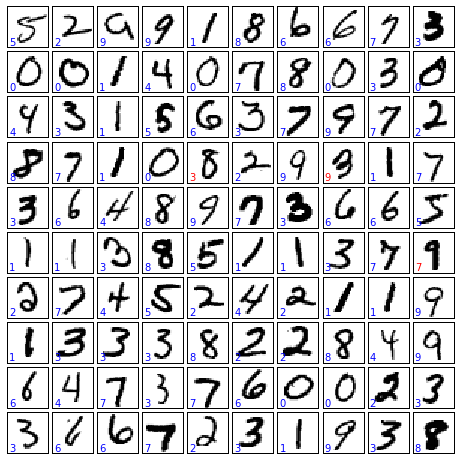
**for i, ax in enumerate(axes.flat): ax.imshow(X\_test[i].reshape((28,28)), cmap='binary', \**

**interpolation='nearest') ax.text(0.05, 0.05, str(int(y\_model[i])), \**

**transform=ax.transAxes, \**

**color='blue' if(y\_test[i] == y\_model[i])\ else 'red')**

**plt.show()**



จากตต วอยพาง ตต วเลขสธนนตาเงวนคพอผลลต พธลจากการพยากรณลทพธถมกตตอง และตต วเลขสธแดงคพอการ พยากรณลผวดพลาด

บทททท**9** การจตดหมวดหมมทขขอมมลดขวย **Naive Bayes (**ฺ**Naive Bayes Classification)**

Naive Bayes Classification เปกนการจต ดหมวดหมมพขตอมมลแบบทพธมธการเรธยนการสอน

(Supervised Learning) โดยใชตหลต กความนพาจะเปกนเขตามาชพวยคนานวณ

ตต วอยพางตพอไปนตธ ใชตไลบรารพธ Seaborn ในการเรธยกใชตขตอมมล Iris Dataset

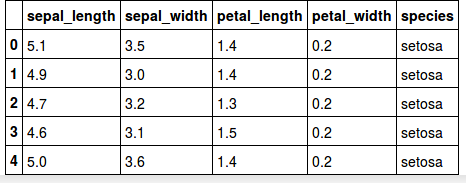
**import seaborn as sns**

**iris = sns.load\_dataset('iris') print("Type", type(iris))**

**iris.head()**

('Type', <class 'pandas.core.frame.DataFrame'>)

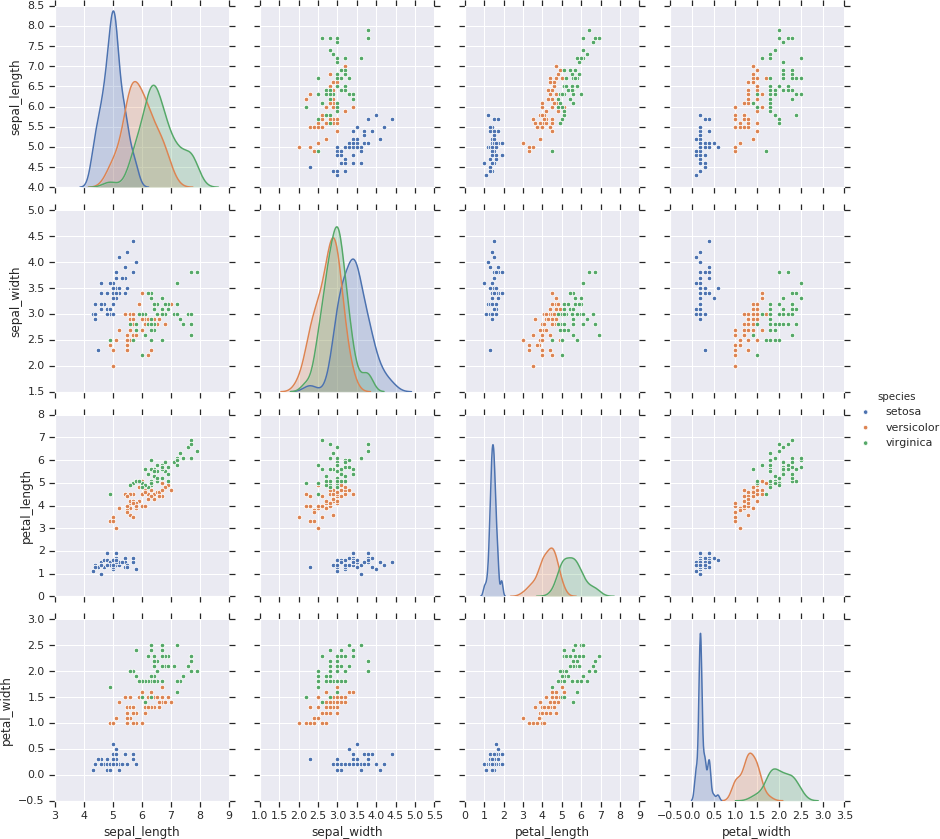
ขตอมมลทพธโหลดมาจะจต ดเกกบอยมพในตต วแปร iris โดยทพธตต วแปร iris จะเกกบขตอมมลแบบ pandas DataFrame



สามารถใชตไลบรารพธ Seaborn เพพพอ Visualization ขตอมมล Iris ไดต ววธธการพลอตทนาไดต ดต งตพอไปนตธ

**import matplotlib.pyplot as plt sns.set()**

**sns.pairplot(iris, hue='species',size=3) plt.show()**

****

### การเตรทยมขขอมมลเพทพอใชขในการเรทยนรขม

ชกดขตอมมล Iris มธจนานวน 150 instance และแบพงออกเปกน 3 กลกพม ในตต วอยพางตพอไปนตธ จะใชต คนาสตพง train\_test\_split เพพพอแบพงขตอมมลออกเปกนชกด Train และ Test

**from sklearn.cross\_validation import train\_test\_split**

**Xtrain, Xtest, ytrain, ytest = train\_test\_split(X\_iris, \**

**y\_iris, random\_state=1)**

จากนต ตนสามารถพลอตขตอมมล iris เพพพอดมการกระจายของขตอมมล แตพทต งต นตธ จะตตองแปลง Label ทพธ อยมพในรมปแบบของตต วอต กษร ใหตเปกนตต วเลขเสธยกพอน

**ytrain[ytrain.iloc[0:] == 'versicolor'] = 1 ytrain[ytrain.iloc[0:] == 'virginica'] = 2 ytrain[ytrain.iloc[0:] == 'setosa'] = 3**

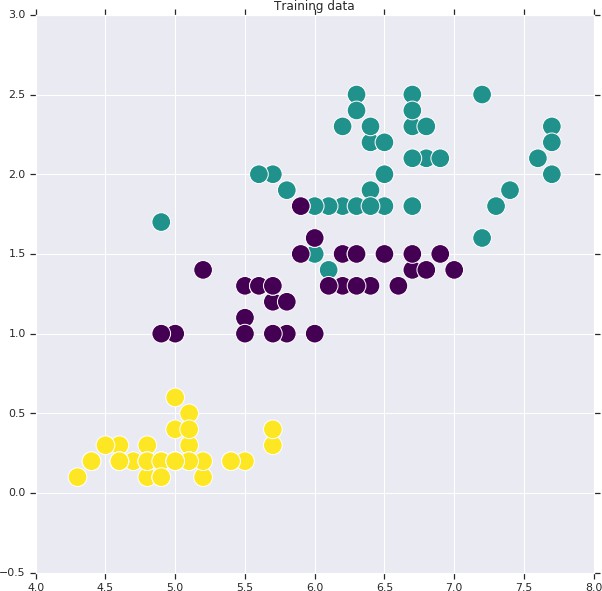
เมพพอแปลง Label ใหตมธคพาเปกน 1,2 และ 3 จากนต ตนจทงพลอตขตอมมล iris โดยใชตคนาสตพงดต งตพอไปนตธ

**import matplotlib.pyplot as plt**

**plt.figure(figsize=(10,10)) plt.scatter(Xtrain.iloc[:,0:1], Xtrain.iloc[:,3:4], \**

**c=ytrain[:], s=350, cmap='viridis') plt.title('Training data')**

**plt.show()**

****

สรขางโมเดล **Naive Bayes**

สามารถใชตคนาสตพง GaussianNB() ซพทงกกคพอ Gaussian Naive Bayes ในการสรตางโมเดล และ คนาสตพง fit() สนาหรต บเรธยนรมตขตอมมล

**from sklearn.naive\_bayes import GaussianNB**

**model = GaussianNB() model.fit(Xtrain, ytrain)**

GaussianNB(priors=None)

### พยากรณหขขอมมลดขวยโมเดล **Naive Bayes** และแสดงประสวทธวภาพของ โมเดล

การพยากรณลทนาไดตโดยใชตคนาสตพง predict() และการแสดงผลความถมกตตองใชตคนาสตพง accuracy\_score() สกดทตายแลตวใชต Confusion Matrix เพพพอดมลต กษณะการพยากรณลขตอมมล ดต งตต วอยพางตพอไปนตธ

**from sklearn.metrics import accuracy\_score**

**y\_model = model.predict(Xtest) accuracy\_score(ytest, y\_model)**

0.9736842105263158

ผลลต พธลทพธไดตจากการพยากรณลคพอ 97.37% จากนต ตนใชตคนาสตพงดต งตพอไปนตธ เพพพอแสดงคพา

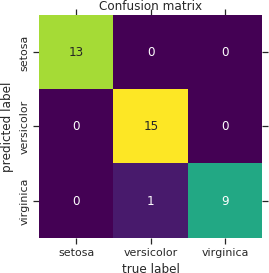
Confusion Matrix

**from sklearn.metrics import confusion\_matrix mat = confusion\_matrix(ytest, y\_model)**

**sns.heatmap(mat.T, square=True, annot=True, fmt='d', \ cbar=False, cmap='viridis',**

**xticklabels=['setosa', 'versicolor', 'virginica'],\ yticklabels=['setosa', 'versicolor', 'virginica'])\**

**plt.title('Confusion matrix') plt.xlabel('true label') plt.ylabel('predicted label'); plt.show()**



บทททท**10** การววเคราะหหองคหประกอบหลตก **(**ฺ**Principal Component Analysis)**

การววเคราะหลองคลประกอบหลต ก (Principal Component Analysis: PCA) เปกนววธธทพธใชตเพพพอ ววเคราะหลขตอมมลทพธมธหลายตต วแปร (Variable) เพพพอหาความสตมพต นธลของตต วแปรเหลพานต ตน ทนาใหตเกวดการ ลดขนาดเมตรวกซล (Matrix) ทพธมธความซต บซตอนเลกกลงงพายตพอการอธวบาย[1](#_bookmark74) นต กววจต ยจทงใชตววธธ PCA เพพพอ นนามาลดขนาดของคกณลต กษณะพวเศษ (Feature) ใหตมธเลกกลง [2](#_bookmark75) ทนาใหตลดเวลาในการสรตางโมเดล

### การสรขางโมเดล **PCA**

ตต วอยพางตพอไปนตธ ใชตชกดขตอมมล Iris ในการทดสอบ ซพทงชกดขตอมมล Iris มธทต งต สวตน 4 attribute ซพทง โหลดขตอมมลโดยใชตไลบรารพธ Seaborn

**import seaborn as sns**

**iris = sns.load\_dataset('iris') X\_iris = iris.drop('species', axis=1) y\_iris = iris['species']**

ขตอมมล feature ทต งต 4 column จะถมกจต ดเกกบไวตทพธตต วแปร X\_iris และขตอมมล Label จะถมกเกกบ อยมพทพธตต วแปร y\_iris

**from sklearn.decomposition import PCA**

**model = PCA(n\_components=2) model.fit(X\_iris)**

PCA(copy=True, iterated\_power='auto', n\_components=2, random\_state=None, svd\_solver='auto', tol=0.0, whiten=False)

1. การววเคราะหลสพวนประกอบสนาคต ญ: <http://www.edu.tsu.ac.th/major/eva/files/journal/PRINCIPAL.pdf>
2. PCA: https://[www.gotoknow.org/posts/566063](http://www.gotoknow.org/posts/566063)

การสรตางโมเดลจะเรธยกใชตคนาสตพง PCA() เพพพอกนาหนดคพาพารามวเตอรล และคนาสตพง fit() เพพพอเรธยน รมตลต กษณะของขตอมมล โดยโมเดลจะถมกเกกบไวตทพธตต วแปรชพพอ model

สามารถตรวจสอบจนานวนของ component ทพธไดตจากการคนานวณดตวย PCA โดยใชตคนาสตพงดต ง

ตพอไปนตธ

**print("check number of components", model.n\_components\_)**

('check number of components', 2)

ขต นต ตอนตพอไปคพอเปลพธยนรมปแบบของขตอมมล (Transform the data) ใหตตรงกต บรมปแบบของ

scikit-learn จากตต วอยพางใชตววธธการดต งตพอไปนตธ

**X\_2D = model.transform(X\_iris)**

**print("show first row of data", X\_2D[0,:])**

**print model.transform(X\_iris.iloc[0,:]. as\_matrix().reshape(1, -1))**

จากนต ตนทดสอบขตอมมล X\_2D โดยเลพอกแถวทพธ 0 มาแสดง ดต งตต วอยพางตพอปนตธ

**print("show first row of data", X\_2D[0,:])**

('show first row of data', array([-2.68412563, 0.31939725]))

ตต วอยพางตพอไปแสดงใหตเหกนถทงการแปลงขตอมมล X\_iris เฉพาะแถวทพธ 1 โดยใชตคนาสตพง

transform()

**print model.transform(X\_iris.iloc[0,:].as\_matrix().reshape(1, -1))**

[[-2.68412563 0.31939725]]

### การเพทวมขขอมมลจากตตวแปรเขขาไปเกกบเพทวมใน **DataFrame**

จากตต วแปร iris ทพธโหลดมาโดยใชตไลบรารพธ seaborn สามารถเพพวมขตอมมลเขตาไปใหมพไดต ดต ง ตต วอยพางตพอไปนตธ

**iris['PCA1'] = X\_2D[:,0]**

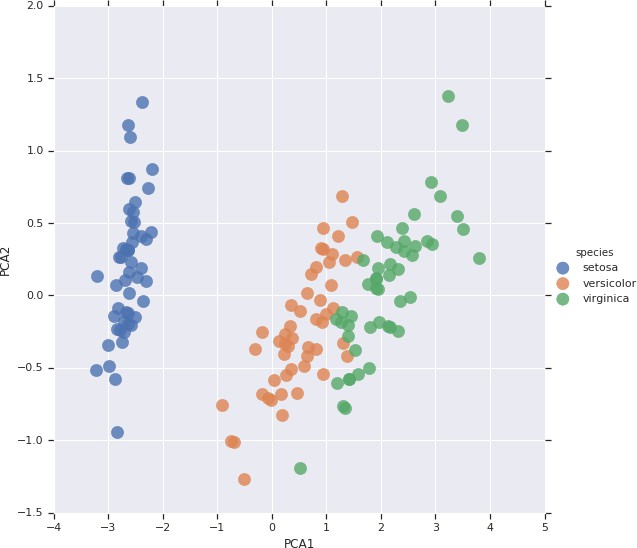
**iris['PCA2'] = X\_2D[:,1]**

จากตต วอยพางขตางตตน ไดตเพพวมขตอมมล X\_2D เขตาไปในตต วแปร iris จนานวน 2 คอลต มนล โดย กนาหนดใหตชพพอของขตอมมลคพอ PCA1 และ PCA2

การพลอตขตอมมลดตวยไลบรารพธ seaborn เพพพอดมลต กษณะการกระจายของขตอมมลทพธไดตจากการ คนานวณดตวยววธธ PCA

**import seaborn as sns**

**sns.lmplot("PCA1", "PCA2", hue='species', data=iris, \ fit\_reg=False, size=8, scatter\_kws={"s": 150})**

****

เปลททยนจาจ นวนของ **Components**

สามารถเปลพธยนจนานวนของ Components ดตวยการกนาหนดทพธตต วแปร n\_components

ดต งตต วอยพางตพอไปนตธ

**from sklearn.decomposition import PCA**

**model = PCA(n\_components=5) model.fit(X\_iris)**

**X\_2D = model.transform(X\_iris)**

ValueError Traceback (most recent call last)

<ipython-input-58-acad37c053c8> in <module>() 2

1. model = PCA(n\_components=5) # 2. Instantiate the model with hyperparameters

----> 4 model.fit(X\_iris) # 3. Fit to data. Notice y is not specified!

5 X\_2D = model.transform(X\_iris) # 4. Transform the data to two dimenstions

/usr/local/lib/python2.7/dist-packages/sklearn/decomposition/pca.pyc in fit(self, X, y)

1. Returns the instance itself.
2. """

--> 307 self.\_fit(X)

308 return self

309

/usr/local/lib/python2.7/dist-packages/sklearn/decomposition/pca.pyc in

\_fit(self, X)

1. # Call different fits for either full or truncated SVD
2. if svd\_solver == 'full':

--> 368 return self.\_fit\_full(X, n\_components)

1. elif svd\_solver in ['arpack', 'randomized']:
2. return self.\_fit\_truncated(X, n\_components, svd\_solver)

/usr/local/lib/python2.7/dist-packages/sklearn/decomposition/pca.pyc in

\_fit\_full(self, X, n\_components)

1. raise ValueError("n\_components=%r must be between 0 and "
2. "n\_features=%r with svd\_solver='full'"

--> 383 % (n\_components, n\_features)) 384

385 # Center data

ValueError: n\_components=5 must be between 0 and n\_features=4 with svd\_solver='full'

ในกรณธ ทพธกนาหนดใหต n\_components มธคพาเทพากต บ 5 จะเกวด error ดต งขตอความ error ขตางตตน ทต งต นตธ เนพพ องจากชกดขตอมมล Iris มธเพธยง 4 attribute เทพานต ตน ดต งนต ตน ในการคนานวณดตวยววธธ PCA จะตตอง คนานท งถทงจนานวนของ attribute เปกนหลต ก ซพทงไมพสามารถกนาหนดใหต n\_components มธคพามากกวพา จนานวนของ attribute

**from sklearn.decomposition import PCA**

**model = PCA(n\_components=3) model.fit(X\_iris)**

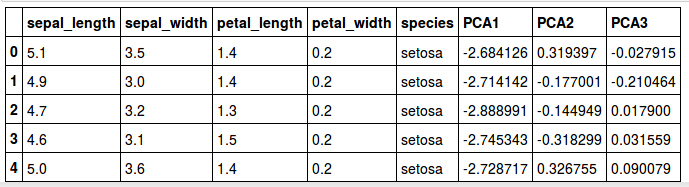
**X\_2D = model.transform(X\_iris)**

**iris['PCA1'] = X\_2D[:,0]**

**iris['PCA2'] = X\_2D[:,1]**

**iris['PCA3'] = X\_2D[:,2]**

**iris.head()**



จากตต วอยพางขตางตตน ทดสอบดตวยการกนาหนด n\_components ใหตมธคพาเทพากต บ 3 และแสดง ขตอมมลทพธไดตจากการคนานวณดตวยววธธ PCA ซพทงเกกบอยมพในตต วแปร PCA1, PCA2 และ PCA3

สรขางโมเดล **Naive Bayes** ดขวยคชณลตกษณะพวเศษทททไดขจาก **PCA**

เพพพอทดสอบประสวทธวภาพของอต ลกอรวธทม ในกรณธ นตธ ไดตนนาคกณลต กษณะพวเศษทพธไดตจากการ คนานวณดตวยววธธ PCA มาใชตเพพพอสรตางโมเดลดตวยววธธ Naive Bayes

**from sklearn.cross\_validation import train\_test\_split X\_iris = iris.drop('species', axis=1)**

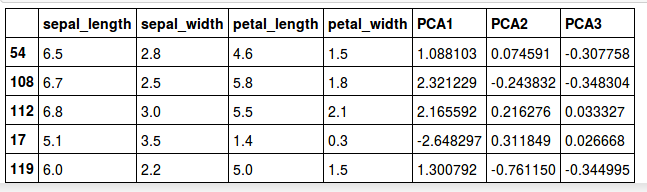
**y\_iris = iris['species']**

**Xtrain, Xtest, ytrain, ytest = train\_test\_split(X\_iris,\**

**y\_iris, random\_state=1)**

เรพวมตตนดตวยการแบพงขตอมมลออกเปกนชกดเรธยนรมต และชกดทดสอบ ดตวยคนาสตพง train\_test\_split() จากนต ตนใชตคนาสตพง Xtrain.head() เพพพอดมขตอมมลและลนาดต บของขตอมมลทพธผพานการสลต บจากคนาสตพง train\_test\_split()

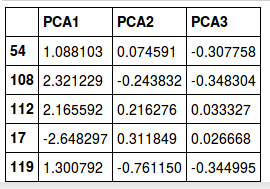
**X\_train.head()**

****

ขต นต ตอนถต ดไป เลพอกขตอมมลในคอลต มนล PCA1, PCA2, PCA3 เพพพอใชตสนาหรต บการเรธยนรมต และ การทดสอบ

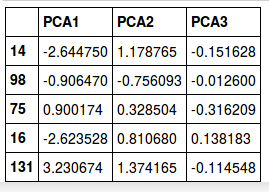
**Xtrain = Xtrain.ix[:,['PCA1','PCA2','PCA3']]**

**Xtrain.head()**

****

**Xtest = Xtest.ix[:,['PCA1','PCA2','PCA3']]**

**Xtest.head()**

****

จากนต ตนนนาขตอมมลจากตต วแปร Xtrain และ Xtest ขตางตตน มาสรตางโมเดลดตวยววธธ Naive Bayes ดต งตต วอยพางตพอไปนตธ

**from sklearn.naive\_bayes import GaussianNB**

**model = GaussianNB() model.fit(Xtrain, ytrain)**

GaussianNB(priors=None)

### การพยากรณหคชณลตกษณะพวเศษทททไดขจาก **PCA** ดขวยอตลกอรวธธม **Naive Bayes** และประสวทธวภาพจากการพยากรณห

ทนาการพยากรณลขตอมมลดตวยคนาสตพงดต งตพอไปนตธ

**y\_model = model.predict(Xtest)**

จากนต ตนทนาการคนานวณหาคพาความถมกตตอง ดต งตต วอยพางตพอไปนตธ

**from sklearn.metrics import accuracy\_score accuracy\_score(ytest, y\_model)**

0.9473684210526315

ความถมกตตองทพธไดตจากการนนาคกณลต กษณะพวเศษของ PCA จนานวน 3 components ไปทนาการ สรตางโมเดล และทดสอบ ปรากฎวพามธความถมกตตอง 94.74%

**from sklearn.metrics import confusion\_matrix mat = confusion\_matrix(ytest, y\_model)**

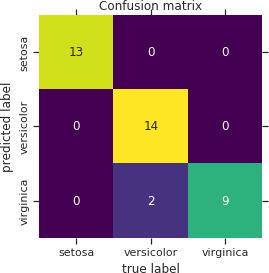
**sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False, cmap='viridis',**

**xticklabels=['setosa', 'versicolor', 'virginica'], yticklabels=['setosa', 'versicolor', 'virginica'])**

ขตอมมล

สกดทตายแสดงผลการคนานวณคพา Confusion matrix เพพพพอดมความถมกตตองของการพยากรณล

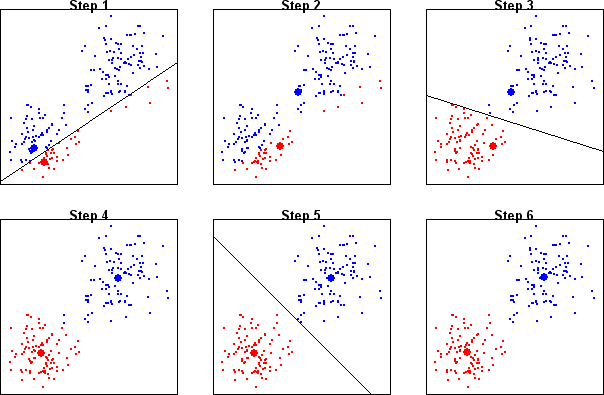
**plt.title('Confusion matrix') plt.xlabel('true label') plt.ylabel('predicted label'); plt.show()**

****

บทททท**11** การจตดกลชทมดขวยอตลกอรวธธม **K-Means**

# (ฺK-Means Clustering)

อต ลกอรวธทม K-Means เปกนววธธทพธใชตสนาหรต บคตนหาจนานวนของคลต สเตอรล (Cluster) จากขตอมมลทพธ ไมพปรากฏคลาส (Class) หรพอไมพมธ Label ซพทงจะเรธยกวพา Unlabeled Data ววธธนตธ จทงเปกนววธธ Unsupervised Learning หรพอววธธ Clustering



ภาพประกอบททท **16:** ตต วอยพางการจต ดกลกพมดตวยววธธ K-Means โดยแสดงใหตดมทธละขต นต ตอน

จากภาพประกอบทพธ [16](#_bookmark80) หากดมดตวยตาเปลพาขตอมมลสามารถแบพงออกเปกน 2 กลกพม โดยจนาลองใหต เปกนกลกพมสธแดง และกลกพมสธนนตาเงวน

**Step 1** ตตองสกพมเลพอกคพากลาง (Centroid) เพพพอใชตเปกนตต วแทนของกลกพมสธแดง และสธนนตาเงวน

เพพพอใชตสนาหรต บคนานวณหาคพาระยะหพาง (Distance Measurement) เพพพอเปรธยบเทธยบระหวพาง จกดขตอมมลและคพา Centroid ทต งต 2 กลกพม (สธแดง และสธนนตาเงวน) หากจกดนต ตนใกลตกลกพมใดทพธสกดจะถมก กนาหนด (Assign) ใหตอยมพในกลกพมนต ตน เชพน หากใกลตกลกพมสธแดง จกดนต ตนกกจะถมกกนาหนดใหตเปกนสธแดง

**Step 2** เมพพอเปลพธยนคพาใหตกต บทกกจกดขตอมมลเปกนทพธเรธยบรตอย จากนต ตนใหตคนานวณเพพพอหาคพา Centroid ใหมพ ในกรณธ ทพธคนานวณหาคพา Centroid ของกลกพมสธนนตาเงวน จะนนาขตอมมลทกกจกดทพธเปกน สธนนตาเงวนมาคนานวณ เชพนเดธยวกต นกต บสธแดง ดต งนต ตน จะไดตจกด Centroid ใหมพ ทพธใชตเปกนตต วแทนของกลกพม ตพอไป

**Step 3** คนานวณหาคพา Distance ระหวพาง Centroid ทต งต 2 กลกพม และจกดขตอมมลทกกจกด เพพพอจต ด กลกพมขตอมมลใหมพ ทนาไปจนกระทตพงจกดแตพละจกดไมพมธการเปลพธยนแปลง หรพอมธการเปลพธยนแปลงนตอยทพธสกด (Step 4-5) โปรแกรมจะหยกดการทนางาน

**Step 6** เมพพอไมพมธการเปลพธยนแปลงกลกพมของขตอมมล ใหตคนานวณหา Centroid เพพพอใชตเปกนโมเดล สามารถนนาไปทดสอบกต บชกดขตอมมล Test set และหากไดตผลดธ กกสามารถนนาไปใชตงานจรวงไดต

### จาจ ลองขขอมมลเพทพอใชขในอตลกอรวธธม **K-Means**

**import matplotlib.pyplot as plt import numpy as np**

**from sklearn.datasets.samples\_generator import make\_blobs**

**rng = np.random.RandomState(0) colors = rng.rand(300)**

**X, y\_true = make\_blobs(n\_samples=300, centers=4,**

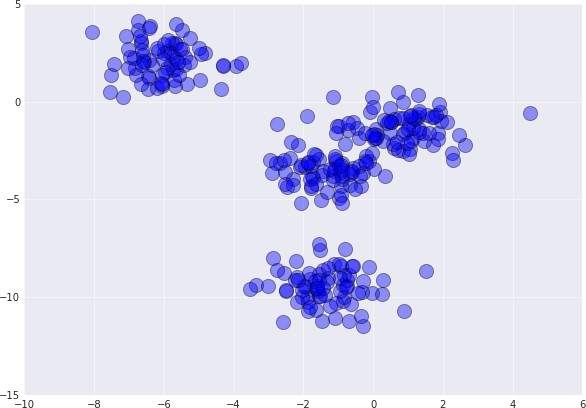
**cluster\_std=0.85, random\_state=2)**

**plt.style.use('seaborn-darkgrid') plt.figure(figsize=(10,7))**

**plt.scatter(X[:, 0], X[:, 1], s=200, alpha=0.4, cmap='viridis');**

**plt.show()**

จากตต วอยพาง ทนาการจนาลองขตอมมลดตวยคนาสตพง make\_blobs() โดยกนาหนดใหตมธขตอมมลทต งต สวตน 300 ชกดขตอมมล (n\_samples) และแบพงขตอมมลออกเปกน 4 กลกพม (centers) จากนต ตนทนาการ Visualization เพพพอดมลต กษณะการกระจายของขตอมมล ดต งรมปภาพตพอไปนตธ



สรขางโมเดลของอตลกอรวธธม **K-Means**

**from sklearn.cluster import Kmeans**

**kmeans = Kmeans(n\_clusters=4) kmeans.fit(X)**

KMeans(algorithm='auto', copy\_x=True, init='k-means++', max\_iter=300, n\_clusters=4, n\_init=10, n\_jobs=1, precompute\_distances='auto', random\_state=None, tol=0.0001, verbose=0)

การพยากรณหดขวยอตลกอรวธธม **K-Means**

สนาหรต บการพยากรณลดตวย K-Means ในโปรแกรม scikit-learn จะใชตฟต งกลชต น predict()

เชพนเดธยวกต บอต ลกอรวธทมอพพน ๆ

**y\_kmeans = kmeans.predict(X) print("output", y\_kmeans[0:20])**

('output', array([0, 0, 1, 0, 3, 2, 0, 2, 1, 0, 3, 0, 0, 2, 2, 3, 1, 2, 3, 0],

dtype=int32))

จากนต ตนแสดงผลการพยากรณลจนานวน 20 ผลลต พธล เพพพอตรวจสอบคนาตอบ จากนต ตนทนาการ พลอตกราฟ เพพพอดมจกด Centroid และการแบพงกลกพมขตอมมล

**plt.figure(figsize=(10,7))**

# plot group of data

**plt.scatter(X[:, 0], X[:, 1], c=y\_kmeans, s=250, alpha=0.5, cmap='viridis')**

**centers = kmeans.cluster\_centers\_**

# plot cluster/centroid

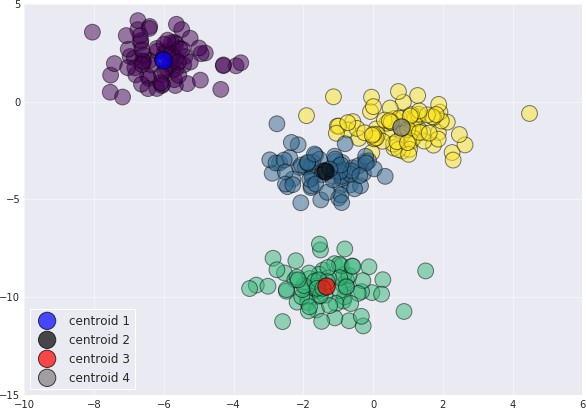
**plt.scatter(centers[0, 0], centers[0, 1], c='blue', s=300, alpha=0.7, label='centroid 1')**

**plt.scatter(centers[1, 0], centers[1, 1], c='black', s=300, alpha=0.7, label='centroid 2')**

**plt.scatter(centers[2, 0], centers[2, 1], c='red', s=300, alpha=0.7, label='centroid 3')**

**plt.scatter(centers[3, 0], centers[3, 1], c='gray', s=300, alpha=0.7, label='centroid 4')**

**plt.legend(frameon=True, loc='lower left') plt.show()**

****

### สรขางขขอมมลใหมทเพทพอทดสอบการแบทงกลชทมดขวยอตลกอรวธธม **K-Means**

ทนาการสรตางขตอมมลจนานวน 10 ชกดเพพพอใชตสนาหรต บการทดสอบโมเดลของ K-Means ดต งตต วอยพาง

ตพอไปนตธ

**X\_test, y\_test\_true = make\_blobs(n\_samples=10, centers=4,**

**cluster\_std=0.85, random\_state=2)**

**X\_test.shape**

(10, 2)

ขต นต ตอนถต ดไป ทนาการพลอตขตอมมลทพธสรตางขตทนมาใหมพ เพพพอใหตทราบวพาขตอมมลทพธสรตางขตทนมาไป กระจายอยมพจกดใดของกราฟ สามารถทนาไดตดต งตพอไปนตธ

**plt.figure(figsize=(10,7))**

# plot group of data

**plt.scatter(X[:, 0], X[:, 1], c=y\_kmeans, s=250, alpha=0.5,**

**cmap='viridis') centers = kmeans.cluster\_centers\_**

# plot cluster/centroid

**plt.scatter(centers[0, 0], centers[0, 1], c='blue', s=300, alpha=0.7, label='centroid 1')**

**plt.scatter(centers[1, 0], centers[1, 1], c='black', s=300, alpha=0.7, label='centroid 2')**

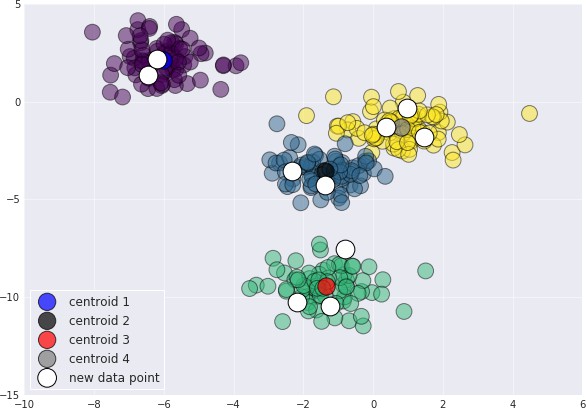
**plt.scatter(centers[2, 0], centers[2, 1], c='red', s=300, alpha=0.7, label='centroid 3')**

**plt.scatter(centers[3, 0], centers[3, 1], c='gray', s=300, alpha=0.7, label='centroid 4')**

# plot new data point

**plt.scatter(X\_test[:, 0], X\_test[:, 1], c='white', s=350, alpha=1, label='new data point')**

**plt.legend(frameon=True, loc='lower left') plt.show()**



### พยากรณหขขอมมลทททสรขางขขธนมาใหมทดขวยอตลกอรวธธม **K-Means**

ในขต นต ตอนการพยากรณลจะทนาการพยากรณลขตอมมล X\_test ทพธมธจนานวนทต งต สวตน 10 ชกดขตอมมล โดยผลลต พธลทพธไดตจากการพยากรณลจะเกกบไวตทพธตต วแปร y\_test\_kmeans

**y\_test\_kmeans = kmeans.predict(X\_test) y\_test\_kmeans**

array([1, 3, 1, 3, 2, 2, 0, 3, 2, 0], dtype=int32)

จากนต ตนพลอตกราฟเพพพอแสดงผลแบบ Visualization

**plt.figure(figsize=(10,7))**

# plot group of data

**plt.scatter(X[:, 0], X[:, 1], c=y\_kmeans, s=250, alpha=0.5, cmap='viridis')**

**centers = kmeans.cluster\_centers\_**

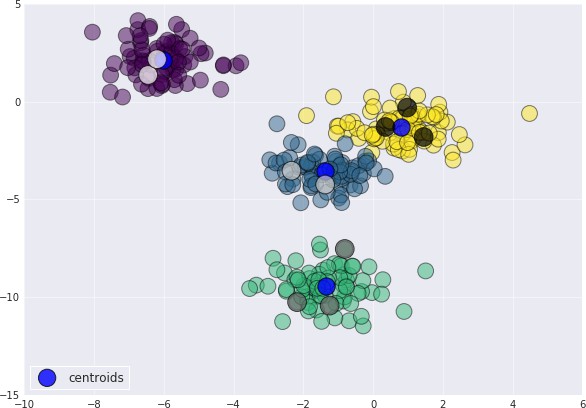
# plot cluster/centroid

**plt.scatter(centers[:, 0], centers[:, 1], c='blue', s=300, alpha=0.8, label='centroids')**

# plot new data point

**plt.scatter(X\_test[:, 0], X\_test[:, 1], c=y\_test\_kmeans, s=350, alpha=0.7)**

**plt.legend(frameon=True, loc='lower left') plt.show()**

****

บทททท**12**

การรขมจาจ ใบหนขา

# (ฺFace Recognition)

สนาหรต บการรมตจนาใบหนตา ในกรณธ นตธ ไดตทดสอบกต บชกดขตอมมล lfw โดยใชตววธธ Principal Component Analysis (PCA) เพพพอลดขนาดของ Feature และสพงตพอไปยต ง Support Vector Machine (SVM) เพพพอเรธยนรมตและสรตางโมเดล การรมตจนาใบหนตาสามารถทนาไดตดต งตพอไปนตธ

**from sklearn.datasets import fetch\_lfw\_people**

**faces = fetch\_lfw\_people(min\_faces\_per\_person=60) print(faces.target\_names) print(faces.images.shape)**

['Ariel Sharon' 'Colin Powell' 'Donald Rumsfeld' 'George W Bush' 'Gerhard Schroeder' 'Hugo Chavez' 'Junichiro Koizumi' 'Tony Blair']

(1348, 62, 47)

ขตอมมลใบหนตาทพธนนามาใชตในการรมตจนา (Recognition) สามารถโหลดไดตจาก scikit-learn โดย ใชตคนาสตพง fetch\_lfw\_people จากนต ตนทนาการพลอตเพพพอดมรมปภาพบกคคล และรายชพพอของแตพละบกคคล ทพธ นนามาใชตในการรมตจนาใบหนตา

**import matplotlib.pyplot as plt**

#plt.figure(figsize=(15,15))

**fig, ax = plt.subplots(3, 5)**

**for i, axi in enumerate(ax.flat): axi.imshow(faces.images[i], cmap='bone') axi.set(xticks=[], yticks=[])**

**axi.set\_ylabel(faces.target\_names[faces.target[i]].split() [-1], color='black')**

**plt.show()**



ภาพประกอบททท **17:** ตต วอยพางใบหนตาจากชกดขตอมมล lfw ทพธนนามา ใชตในการรมตจนาใบหนตา

### สรขางโมเดลของอตลกอรวธธม **SVM**

ในกรณธ นตธ การสรตางโมเดลดตวย SVM จะใชตขตอมมลทพธไดตจากการคนานวณดตวย PCA โดย กนาหนดใหต n\_components=150 ดต งนต ตน ขตอมมลทต งต หมด 150 component จทงเรธยกวพาเปกน คกณลต กษณะพวเศษ (Feature Extraction) ทพธจะสพงตพอไปยต ง SVM เพพพอเรธยนรมตและสรตางออกมาเปกน โมเดล ดต งตต วอยพางตพอไปนตธ

**from sklearn.svm import SVC**

**from sklearn.decomposition import RandomizedPCA from sklearn.pipeline import make\_pipeline**

**pca = RandomizedPCA(n\_components=150, whiten=True, random\_state=42)**

**svc = SVC(kernel='rbf', class\_weight='balanced') model = make\_pipeline(pca, svc)**

จากนต ตน แบพงขตอมมลออกเปกน 2 สพวน เพพพอใชตสนาหรต บการเรธยนรมต และทดสอบโดยใชตคนาสตพง

train\_test\_split()

**from sklearn.cross\_validation import train\_test\_split Xtrain, Xtest, ytrain, ytest = train\_test\_split(faces.data,**

**faces.target, random\_state=42)**

จากตต วอยพางขตางตตน ในการสรตางโมเดลนต ตน เปกนการใชตคพาเบตพองตตนในการสรตางโมเดล ดต งนต ตน ประสวทธวภาพอาจจะไมพดธมาก ดต งนต ตน ควรทพธจะปรต บเปลพธยนพารามวเตอรล (Tuning Parameter) เพพพอใหต โมเดลทพธสรตางนต ตนมธประสวทธวภาพดธทพธสกด สามารถทนาไดตโดยใชตคนาสตพง GridSearchCV()

**from sklearn.grid\_search import GridSearchCV**

**param\_grid = {'svc** **C': [1, 5, 10, 50],**

**'svc** **gamma': [0.0001, 0.0005, 0.001, 0.005]}**

**grid = GridSearchCV(model, param\_grid)**

**%time grid.fit(Xtrain, ytrain) print(grid.best\_params\_)**

CPU times: user 1min 14s, sys: 1min, total: 2min 15s Wall time: 33.8 s

{'svc gamma': 0.001, 'svc C': 5}

คพาพารามวเตอรลทพธใชตในการปรต บเปลพธยนไดตแกพคพา **svc** **C** และ **svc** **gamma** ซพทงเปกนคพา พารามวเตอรลของ RBF Kernel โดยคพาทพธดธทพธสกดในการทดสอบคพอ **svc** **gamma = 0.001** และ **svc** **c = 5** จากนต ตนนนาคพาทต งต 2 คพาไปสรตางโมเดล ดต งนตธ

**model = grid.best\_estimator\_**

โมเดลทพธสรตางดตวยตต วแปรทพธดธทพธสกดจะถมกจต ดเกกบไวตทพธตต วแปรชพพอวพา model จากนต ตนจทงสามารถนนา ไปเพพพอพยากรณลขตอมมลชกดทดสอบ ดต งตต วอยพางตพอไปนตธ

**yfit = model.predict(Xtest)**

โดยผลลต พธลทพธไดตจากการพยากรณลจะถมกจต ดเกกบไวตทพธตต วแปร yfit จากนต ตนสามารถ

Visualization รมปภาพใบหนตา และแสดงคนาตอบในการพยากรณล ดต งตพอไปนตธ

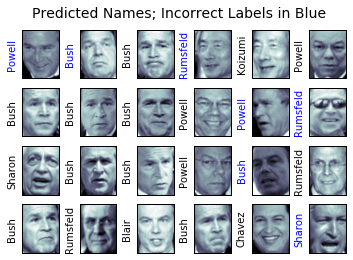
**fig, ax = plt.subplots(4, 6)**

**for i, axi in enumerate(ax.flat): axi.imshow(Xtest[i].reshape(62, 47), cmap='bone') axi.set(xticks=[], yticks=[]) axi.set\_ylabel(faces.target\_names[yfit[i]].split()[-1],**

**color='black' if yfit[i] == ytest[i] else 'blue')**

**fig.suptitle('Predicted Names; Incorrect Labels in Blue',**

**size=14);**



ภาพประกอบททท **18:** ผลลต พธลทพธไดตจากการพยากรณลใบหนตาดตวย ววธธ PCA และ SVM

จากภาพประกอบทพธ [18](#_bookmark88) รายชพพอสธนนตาเงวน คพอการพยากรณลทพธผวดพลาด และรายชพพอสธดนาคพอการ พยากรณลทพธถมกตตอง

การวตดประสวทธวภาพของการรขมจาจ ใบหนขา **(Classification Report)**

การวตดประสวทธวภาพของการรมตจนาใบหนตา ในกรณธ นตธ ใชตคนาสตพง classification\_report() เพพพอทพธ จะตรวจสอบคพา precision, recall และ f1-score

**from sklearn.metrics import classification\_report print(classification\_report(ytest, yfit,**

**target\_names=faces.target\_names))**

**precision recall f1-score support**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ariel Sharon | 0.61 | 0.73 | 0.67 | 15 |
| Colin Powell | 0.82 | 0.88 | 0.85 | 68 |
| Donald Rumsfeld | 0.76 | 0.84 | 0.80 | 31 |
| George W Bush | 0.88 | 0.88 | 0.88 | 126 |
| Gerhard Schroeder | 0.75 | 0.78 | 0.77 | 23 |
| Hugo Chavez | 1.00 | 0.70 | 0.82 | 20 |
| Junichiro Koizumi | 0.90 | 0.75 | 0.82 | 12 |
| Tony Blair | 0.89 | 0.81 | 0.85 | 42 |
| avg / total | 0.85 | 0.84 | 0.84 | 337 |

การแสดงความถมกตของของการพยากรณหดขวย **Confusion Matrix**

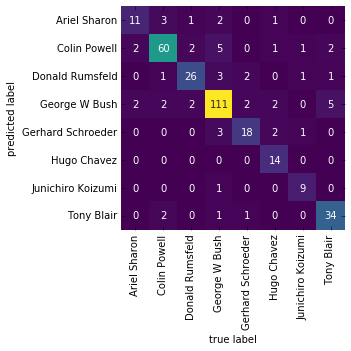
**from sklearn.metrics import confusion\_matrix import seaborn as sns**

**mat = confusion\_matrix(ytest, yfit)**

**sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False, cmap='viridis',**

**xticklabels=faces.target\_names, yticklabels=faces.target\_names)**

**plt.xlabel('true label') plt.ylabel('predicted label');**

****

อตตราความถมกตของ **(Accuracy Result)** ของการพยากรณหรมปภาพ ใบหนขา

การคนานวณความถมกตตอง (Accuracy Result) ของอต ลกอรวธทม SVM สามารถใชตคนาสตพง

accuracy\_score() โดยผลลต พธลทพธไดตจากการพยากรณลคพอ 83.97%

**from sklearn.metrics import accuracy\_score**

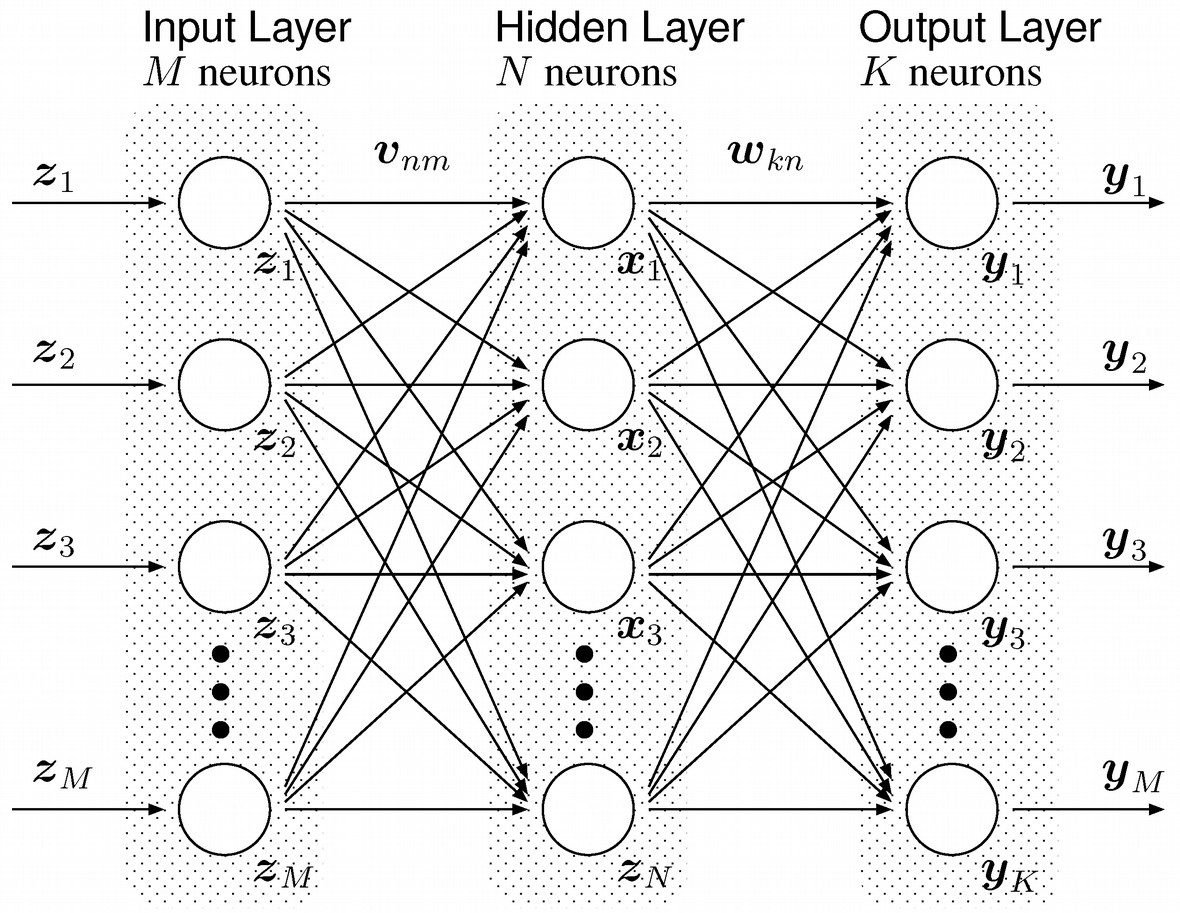
**print("Accuracy", accuracy\_score(ytest, yfit)\*100)**

('Accuracy', 83.97626112759644)

บทททท**13** การรขมจาจ ตตวอตกษร

# (ฺCharacter Recognition)

การรมตจนาตต วอต กษร (Character Recognition) ดตวยนววรอลเนกตเววรลก แบบ Multi-Layer Perceptron (MLP) ในกรณธ นตธ ทดสอบกต บชกดขตอมมล MNIST โดยคกณลต กษณะพวเศษ (Feature Extraction) ทพธนนามาใชตคพอคพาสธเทาของแตพละพวกเซล ดต งนต ตน รมปภาพขนาด 28x28 พวกเซลจะมธ คกณลต กษณะพวเศษจนานวน 784 attribute ทพธจะนนาไปเรธยนรมต



ภาพประกอบททท **19:** ตต วอยพางโครงสรตางของ Multi-Layer Perceptron (MLP)

จากภาพประกอบทพธ [19](#_bookmark93) โครงสรตางของนววรอลเนกตเววรลกแบบ MLP ประกอบดตวย 3 ชต นต (Layer) คพอ Input Layer, Hidden Layer และ Output Layer โดยทต งต 3 ชต นต จะมธการเชพพอมตพอกต น ของโหนดอยพางสมบมรณล (Fully-Connected)

เมพพอนนามาปรต บใชตกต บขตอมมลรมปภาพตต วอต กษรทพธมธขนาด 28x28 พวกเซล ดต งนต ตน ในชต นต Input Layer จทงกนาหนดใหตมธจนานวน 784 โหนด และในชต นต Output Layer ถมกกนาหนดใหตเปน 10 โหนด

ตามจนานวนของตต วเลข 0-9 สพวนในชต นต Hidden Layer นต ตน จะตตองทนาการคนานวณเพพพอหาจนานวน โหนดทพธเหมาะสมทพธสกดกต บชกดขตอมมล

ในกรณธ นตธ ใชตชกดขตอมมล MNIST ทพธมธจนานวน 70,000 ตต วเลข โดยใชตโปรแกรม scipy ในการ

โหลด

**from scipy.io import loadmat**

**mnist\_raw = loadmat("mldata/mnist-original.mat") mnist = {**

**"data": mnist\_raw["data"].T,**

**"target": mnist\_raw["label"][0],**

**"COL\_NAMES": ["label", "data"],**

**"DESCR": "mldata.org dataset: mnist-original",**

**}**

**X,y = mnist['data'], mnist['target']**

**X.shape, y.shape**

((70000, 784), (70000,))

จากนต ตนใชตคนาสตพงดต งตพอไปนตธ เพพพอ Visualization รมปภาพตต วเลข

**import matplotlib.pyplot as plt**

**some\_digit = X[36000]**

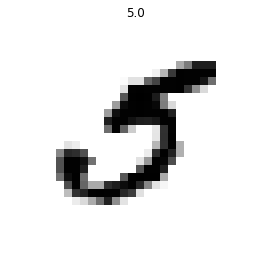
**some\_digit\_image = some\_digit.reshape(28, 28)**

**plt.imshow(**

**some\_digit\_image, cmap = plt.cm.binary,**

**interpolation="nearest")**

**plt.title(y[36000]) plt.axis("off") plt.show()**

****

เนพพ องจากในชกดขตอมมลดต งกลพาว ตต งต แตพชกดขตอมมลทพธ 1-60,000 และชกดขตอมมลทพธ 60,001-70,000 ขตอมมลไดตถมกจต ดเรธยงตามตต วเลข 0-9 ดต งนต ตน หากตตองการขตอมมลบางสพวนมาเพพพอทดสอบ จทงควรทพธจะ สลต บ (Shuffle) ขตอมมลเสธยกพอน การสลต บขตอมมลสามารถทนาไดตดต งตพอไปนตธ

**import numpy as np**

**shuffle\_index = np.random.permutation(70000) X, y = X[shuffle\_index], y[shuffle\_index]**

**X\_train, X\_test, y\_train, y\_test = X[:60000], X[60000:],**

**y[:60000], y[60000:]**

**print(X\_train.shape, X\_test.shape, y\_train.shape,\ y\_test.shape)**

((60000, 784), (10000, 784), (60000,), (10000,))

จากนต ตนสามารถแสดงขตอมมล โดยแสดงแบบ Visualization และแสดง Label ประกอบ ดต งนตธ

**import matplotlib.pyplot as plt**

**fig, axes = plt.subplots(10, 10, figsize=(8, 8),**

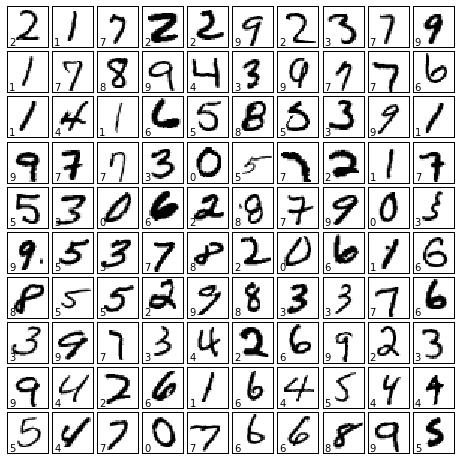
**subplot\_kw={'xticks':[], 'yticks':[]}, gridspec\_kw=dict(hspace=0.1, wspace=0.1))**

**for i, ax in enumerate(axes.flat): ax.imshow(X\_train[i].reshape((28,28)), cmap='binary',**

**interpolation='nearest') ax.text(0.05, 0.05, str(int(y\_train[i])),**

**transform=ax.transAxes, color='black')**

**plt.show()**

****

### สรขางโมเดลของอตลกอรวธธม **MLP**

การใชตงาน MLP ใชต scikit-learn จะตตองใชตโมดมล MLPClassifier โดยในตต วอยพางไดต กนาหนดใหต Hidden Layer มธขนาด 100 โหนด (hidden\_layer\_sizes) และทนาการทดสอบจนานวน 10 รอบ (max\_iter) และกนาหนดอต ตราการเรธยนรมตทพธ 0.001 (learning\_rate\_init) จากนต ตนใชตคนาสตพง fit() เพพพอสรตางโมเดล และเกกบโมเดลไวตทพธตต วแปร mlp

**from sklearn.neural\_network import MLPClassifier**

**mlp = MLPClassifier(hidden\_layer\_sizes=(100,), max\_iter=10,**

**alpha=1e-4,**

**solver='sgd', verbose=10, tol=1e-5, random\_state=1, learning\_rate\_init=0.001)**

**mlp.fit(X\_train, y\_train)**

Iteration 1, loss = 1.81871326

Iteration 2, loss = 1.05944149

Iteration 3, loss = 0.69554161

Iteration 4, loss = 0.50766222

Iteration 5, loss = 0.38525626

Iteration 6, loss = 0.33714014

Iteration 7, loss = 0.30388212

Iteration 8, loss = 0.28367218

Iteration 9, loss = 0.26573927

Iteration 10, loss = 0.25603315

MLPClassifier(activation='relu', alpha=0.0001, batch\_size='auto', beta\_1=0.9, beta\_2=0.999, early\_stopping=False, epsilon=1e-08, hidden\_layer\_sizes=(100,), learning\_rate='constant', learning\_rate\_init=0.001, max\_iter=10, momentum=0.9, nesterovs\_momentum=True, power\_t=0.5, random\_state=1, shuffle=True, solver='sgd', tol=1e-05, validation\_fraction=0.1, verbose=10, warm\_start=False)

### การวตดประสวทธวภาพของการเรทยนรขม

ในกรณธ นตธ ใชตคนาสตพง score() เพพพอดมผลการทดลองกต บขตอมมลชกดเรธยนรมต และขตอมมลชกดทดสอบ โดยใชตคนาสตพงดต งตพอไปนตธ

**print("Training set score: %f" % mlp.score(X\_train, y\_train)) print("Test set score: %f" % mlp.score(X\_test, y\_test))**

### การพยากรณหและวตดประสวทธวภาพของการขมจาจ

ในการพยากรณลใหตใชตคนาสตพง predict() เพพพอพยากรณลขตอมมล X\_test ดต งตต วอยพางตพอไปนตธ

**yfit = mlp.predict(X\_test)**

**from sklearn.metrics import classification\_report tn = ['0','1','2','3','4','5','6','7','8','9']**

**print(classification\_report(y\_test, yfit,**

**target\_names=tn))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.95 | 0.96 | 0.95 | 971 |
| 1 | 0.98 | 0.96 | 0.97 | 1121 |
| 2 | 0.92 | 0.91 | 0.92 | 971 |
| 3 | 0.91 | 0.89 | 0.90 | 1016 |
| 4 | 0.95 | 0.90 | 0.92 | 1010 |
| 5 | 0.90 | 0.90 | 0.90 | 896 |
| 6 | 0.94 | 0.95 | 0.95 | 1025 |
| 7 | 0.95 | 0.93 | 0.94 | 1080 |
| 8 | 0.83 | 0.91 | 0.87 | 942 |
| 9 | 0.89 | 0.92 | 0.90 | 968 |
| avg / total | 0.92 | 0.92 | 0.92 | 10000 |

แสดงผลลตพธจห ากการพยากรณหดขวย **Confusion Matrix**

from sklearn.metrics import confusion\_matrix import seaborn as sns

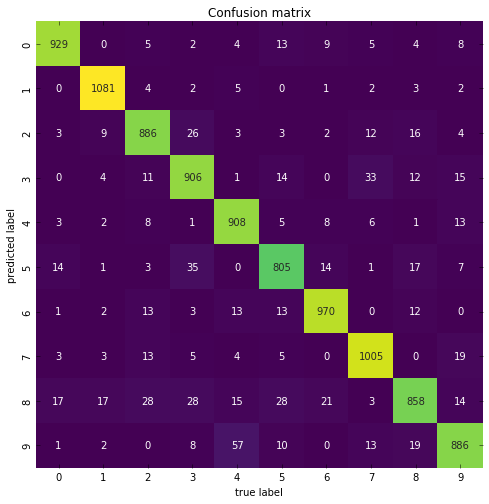
mat = confusion\_matrix(y\_test, yfit)

fig, ax = plt.subplots(figsize=(8,8))

sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False, cmap='viridis', xticklabels=tn,

yticklabels=tn, ax=ax)

plt.title('Confusion matrix') plt.xlabel('true label') plt.ylabel('predicted label');



แสดงอต ตราความถมกตตองจากการพยากรณล โดยผลลต พธลทพธไดตจากการพยากรณลคพอ 92.34%

**from sklearn.metrics import accuracy\_score print("Accuracy", accuracy\_score(y\_test, yfit)\*100)**

('Accuracy', 92.34)

การ **Visualization** รมปภาพตตวเลข และแสดงผลการพยากรณห

เพพพอใหตสะดวกตพอการเปรธยบเทธยบระหวพางคพาทพธแทตจรวง และคพาทพธพยากรณล สามารถ

Visualization ขตอมมลไดตดต งตพอไปนตธ

**import matplotlib.pyplot as plt**

**fig, axes = plt.subplots(10, 10, figsize=(8, 8),**

**subplot\_kw={'xticks':[], 'yticks':[]}, gridspec\_kw=dict(hspace=0.1, wspace=0.1))**

**for i, ax in enumerate(axes.flat): ax.imshow(X\_test[i].reshape((28,28)), cmap='binary',**

**interpolation='nearest')**

**# actual class**

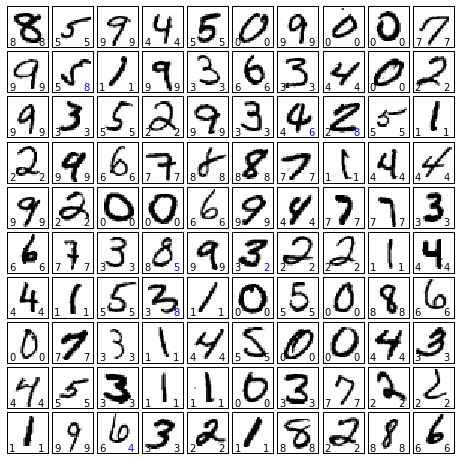
**ax.text(0.05, 0.05, str(int(y\_test[i])), transform=ax.transAxes, color='black')**

**# predict class**

**ax.text(0.75, 0.05, str(int(yfit[i])), transform=ax.transAxes,**

**color='black' if yfit[i] == y\_test[i] else 'blue')**

**plt.show()**

****

ภาพประกอบททท **20:** เปรธยบเทธยบระหวพางคพาทพธแทตจรวง และคพาทพธไดตจากการ พยากรณล

จากภาพประกอบทพธ [20](#_bookmark98) ในแตพละรมปจะมธตต วเลขกนากต บ 2 ตต วเลข โดยตต วเลขทางดตานซตายมพอคพอ Label ของตต วเลขนต ตน ๆ สพวนตต วเลขทางดตานขวาคพอผลลต พธลทพธไดตจากการพยากรณล ดต งนต ตน หาก เปกนการพยากรณลทพธผวดพลาดจะแสดงเปกนสธนนตาเงวน แตพหากพยากรณลถมกตตองกกจะแสดงเปกนสธดนา

