Data Mining

1.2 Mean and sample variance

Jeg bestemte mig hurtigt for, at den hurtigste måde at finde gennemsnittet, ville være ved først at have summen af alle tallene i Train_Y, og derefter dividere dette med antal tal der var.
Derfor lavede jeg først en for-løkke, til at gå alle rækker igennem, så man ville kunne få summen af alle tal.
Summen af alle disse tal gemte jeg i en variabel, som jeg til sidst divideret med antal tal.
Jeg endte med at få et gennemsnit på -10.91161694

Det gennemsnit man fik, skulle man trække fra hver enkelt række, og derefter finde gennemsnittet af dette.
Her genbrugte jeg mange data fra tidligere. Jeg lavede igen en for-løkke så man kunne arbejde med hver række. For hver række trak jeg gennemsnittet fra, og lagde det i anden, så tallet altid ville være positivt.
Her endte jeg med et gennemsnit på 0.895243965831

1.3.1 Build model

Først downloadede jeg numpy fra http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy.
Jeg fulgte algoritme 3, side 12 i "lecture notes". Jeg indsatte "b" i min x matrice, hvilket er en søjle bestående af 1-taller. Herefter brugte jeg formlen $((X^TX)^{-1}X^Ty)$ til at finde de fem parametre som skulle findes. Dette ville give $(w^Tx + b)$, betydende at det sidste tal af de fem vil være b.
Her fik jeg tallene [[-0.79400153], [-1.2229592], [-0.32858475], [-0.78633056], [-8.14943349]]

1.3.2 Training error

Numpy bruges også i denne opgave.
Her skal man ikke bruge b, derfor starter jeg med at slette den fra matricen x. Derudover gemmer jeg de 5 parametre i 2 variabler, hvor b står for sig selv, og de andre 4 står sammen.
Herefter bruger jeg (3.4) fra "lecture notes", til at finde sum-of-squared-error. Denne bruger jeg i en while-løkke, så den tager hver række i både x og y matricerne.
For så at finde mean-squared-error skal man tage resultatet af sum-of-squared-error og dividere med længden af listen af tal.
Her får jeg 0.2747546.
Dette er et temmelig lavt tal, betydende at der ikke er så stor usikkerhed på hvilke tal der hører til hvilken farve.

1.3.3 Test error

Numpy bruges også i denne opgave.
Denne opgave er udført ligesom opgave 1.3.2, og jeg vil derfor ikke gå i detaljer hvordan den er lavet.
I denne opgave får jeg 0.27517963.
Dette er lidt højere end det jeg fik i 1.3.2, omkring 0,0004 højere. Dette vil sige at den har lidt mere

usikkerhed aka. Støj i forhold til 1.3.2. Altså er der mere usikkerhed på hvilke punkter der hører til hvilken farve. Dog er det stadig et forholdsvis lavt tal, og derfor er usikkerheden stadig ikke speciel høj.

2.1 Classification

Numpy bruges også i denne opgave.
Man skal finde den et punkt fra test_X nærmeste nabo i train_X. Ud fra den nærmeste nabo, skal man så beslutte om det er en stjerne eller en galaxe.
Først laver jeg en while-løkke i en while-løkke. Dette gør jeg så jeg kan arbejde inde i test_X og train_X på samme tid. Her bruger jeg 2 variabler til at finde nærmeste nabo og hvad det er. Den første variabel til at huske afstanden, og den anden variabel til at huske hvad det er.
For hvert punkt i test_X beregner den afstanden til hvert eneste punkt i train_X. Dette gør at den skal beregne afstanden til 5000 punkter, 5000 gange.
Når den har gjort det, og fundet nærmeste nabo til et punkt, ser den hvad det er, og gemmer dette i en liste. Til slut har man en lang liste på hvad hvilket punkt er.
Til slut ser den på hvor mange af disse der er rigtige, ved at sammenligne denne nye liste med test_Y.
Her beregner jeg det i procent, og får i procent 98.4666666667

2.2 Dimensionality reduction and visualization

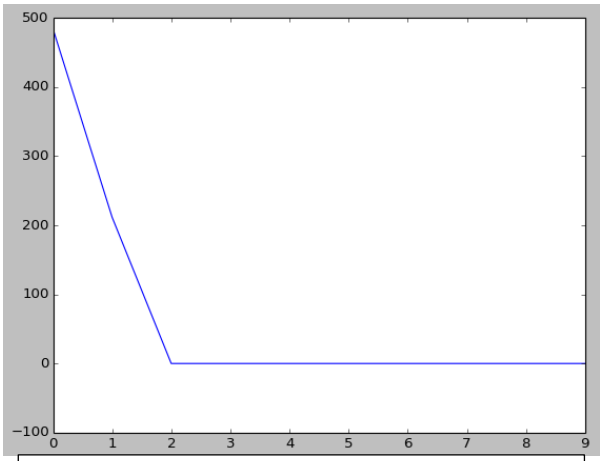
VIGTIG BEMÆRKNING: I denne opgave får jeg ikke de rigtige resultater. Dette skyldes en fejl et sted i koden til at beregne eigenspectrum. Dette forkerte resultat påvirker derefter de andre resultater, så ingen af resultaterne giver det rigtige.

Numpy bruges også i denne opgave.
Jeg startede med at lave en while-løkke, hvor jeg tjekkede hvilke der var galakser. Dem der var galakser gemte jeg i en liste, som jeg derefter fandt gennemsnittet i hver søjle.
Herefter fandt jeg eigenspectrum ved at lave en while-løkke der arbejdede med alle galakserne og lavede dem til en 10x10 matrice, hvor jeg så fandt gennemsnittet af denne matrice. Ud fra dette fandt jeg hhv. eigen vektoren og eigen værdierne.

Nu downloader jeg matplotlib, dateutil, pyparsing og six, alle fra hjemmesiden
http://www.lfd.uci.edu/~gohlke/pythonlibs/

Her laver jeg et plot af eigenspectrum (hvilket var forkert), hvilket kan ses på billede 1.

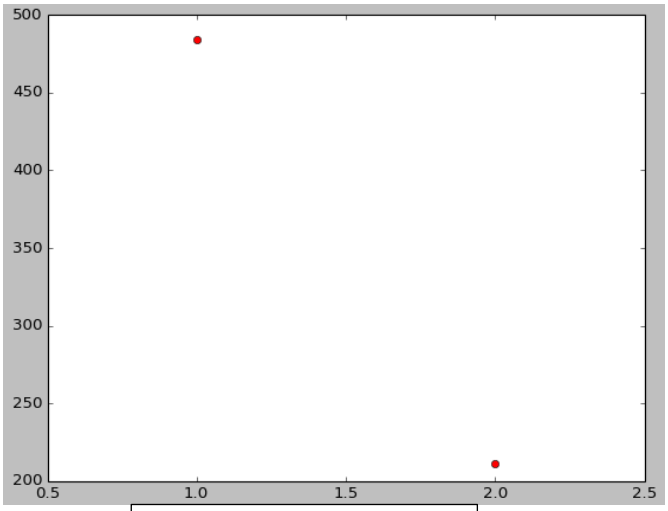
Herefter beregnede jeg mig frem til hvor mange komponenter det krævede at udfylde 90% af den samlede sum. Dette gjorde jeg ved først at finde summen fra eigen værdierne, og derefter kan dette med 0,9.
Herefter kører jeg en while-løkke til at tage 1 komponent ad gangen og se hvor mange procent det er.



Billede 1
Man kan tydeligt se at tallene på y-aksen er meget højere end de burde

Når der er minimum 90% stopper while-løkken og den udskriver hvor mange komponenter det kræver. Her kræver det 3 komponenter.

Til sidst laver jeg et scatter plot af de første 2 komponenter. Dette kan ses på billede 2.



Billede 2
De første 2 komponenter

Som sagt er der en fejl et sted i koden/udregningerne, hvilket gør det svært at beskrive resultaterne. Dog har jeg kontaktet en instruktor som, ud over at bekræfte det er forkert, sagde at selve formen på min graf (billede 1) er rigtig nok. Derved er det altså kun en lille fejl jeg har, som gør at tallene på y-aksen er alt for store, men ellers burde resten være rigtigt.
Det samme gælder så også for mit scatter plot (billede 2), og for hvor mange komponenter der skal bruges.

2.3 Clustering

VIGTIG BEMÆRKNING: Denne opgave nåede jeg desværre ikke at blive færdig med, derfor er der kun en lille del af opgaven, og man vil derfor ikke kunne se noget egentligt resultat. Jeg har kun fået printet, så man kan se hvor mange galakser der havner i hhv. liste A og liste B.

Her starter jeg med at lave 2 centrum, som er 2 tilfældige punkter fra galakse listen. Herefter tjekker ejg hvor langt der er for hvert punkt ud til centrum 1 og centrum 2. Hvis der er kortest til centrum 1, skal den tilføjes i liste A, eller bliver den tilføjet i liste B.

Mere nåede jeg desværre ikke at lave…