

Peter Hellberg

Användarens påverkan i aktiv inlärning

Magistersavhandling i informatonssystem

Handledare: Markku Heikkilä

Anna Sell

Fakulteten för samhällsvetenskaper och

Ekonomi

Åbo Akademi

Åbo 2019

Abstrakt

Titel: Användarens påverkan i aktiv inlärning

Författare: Peter Hellberg

Maskininlärning och artificiell intelligens är aktuella ämnen nu och de utvecklas i en otrolig hastighet som är svår för de flesta att försöka hänga med i. Många personer inser inte ens i hur stor del av deras vardag maskininlärning redan är inblandat i, och mängden ökar konstant dagligen.

Syftet med denna avhandling är att ta reda på hur stor inverkan användaren kan ha i en aktiv inlärningsmodell och hur mycket data krävs för att inläringen för algoritmen skall bli pålitlig i en applikation. För att begränsa avhandlingen så har vi tillsammans med silo.ai utvecklat en bildigenkänningsapplikation som jag använder för att samla data. För att göra applikationen lättanvändlig och tydlig så har vi gjort en undersökning i bl.a. datavisualisering varefter utvecklandet av applikationen har pågått medan testandet av data har gjorts.

Resultaten visade att mängden märkt data som behövs för att algoritmen ska kunna ge ett pålitligt svar beror på bilderna som ingår i applikationen. I testversionen användes en modell med 100 hund och katt bilder samt en modell med 98 bil och flyg bilder. Det visade sig att hund och katt modellen krävde mer färdigmärkt data för att algoritmen kunde ge ett pålitligt resultat, medan bil och flyg modellen krävde betydligt mindre. Hund och katt modellen gav ett pålitligt svar med 0 fel efter 120 märkta bilder, medan bil och flyg modellen krävde endast 60 stycken märkta bilder. Det som dock bör beaktas här var att det var en stor skillnad på vilka bilder som märktes, speciellt i bil och flyg modellen.

Nyckelord: Maskininlärning, aktiv inlärning, algoritm, bildigenkänningsapplikation, datavisualisering

Innehållsförteckning

1	Inledning.....	1
1.1	Introduktion.....	1
1.2	Syfte och avgränsning.....	2
1.3	Problem och frågeställningar.....	2
1.4	Metod och insamling av data.....	2
1.4.1	Forskningsmetod.....	2
1.4.2	Insamling av data.....	3
1.5	Avhandlingens struktur.....	3
2	Maskininlärning.....	5
2.1	Övervakat lärande och oövervakat lärande.....	6
2.2	Klassificering	7
2.3	Regression.....	9
2.4	Klustring.....	10
2.5	Aktiv inlärning.....	12
2.6	Human in the loop.....	14
3	Användargränssnitt och användarupplevelse (UI & UX).....	15
3.1	Användargränssnitt (UI).....	15
3.2	användarupplevelse (UX).....	16
3.2.1	Mätning av användarupplevelsen.....	17
4	Datavisualisering.....	19
4.1	Användning av färger.....	20
4.1.1	RGB.....	21
4.1.2	Rekommenderade färgpaletter.....	22
4.2	Användning av grafer.....	23
4.3	Annoteringar.....	27
5	Utveckling och test av applikationen.....	29
5.1	Bildigenkänningsapplikationen.....	29
5.2	Analys av hund och katt-modellen.....	32
5.3	Analys av bil och flyg-modellen.....	39
5.4	Sammanfattning av resultaten.....	45
6	Diskussion och slutsatser.....	47

Källor.....49

Figurförteckning:

Figur 1 Beslutsträd

Figur 2 Obehandlad data

Figur 3 Klustrad data

Figur 4 Skillnaden mellan kluster- och raster-metoden

Figur 5 Aktiv inläring

Figur 6 Human in the loop

Figur 7 Hierarkisk uppgiftsanalys

Figur 8 Effekten av visualisering

Figur 9 Color Scale Helper

Figur 10 Data kub

Figur 11 Tidsseriegraf

Figur 12 Stapeldiagram med rullgardinsmeny

Figur 13 Nätverksdiagram

Figur 14 Annotation fas 1

Figur 15 Annotation fas 2

Figur 16 Bildigenkänning 1 (Dataset view)

Figur 17 Bildigenkänning 2 (Teaching view)

Figur 18 Testkörning 1

Figur 19 Resultat av test 1

Figur 20 Testkörning 2

Figur 21 Resultat av test 2

Figur 22 Test 3&4

Figur 23 Test 5&6

Figur 24 Test 7&9

Figur 25 Test 8

Figur 26 Test med 40+ märkningar

Figur 27 Test med 120+ märkningar

Figur 28 Resultat diagram

Figur 29 Test med 1 bil och 1 flyg

Figur 30 Test med totalt 10 märkningar

Figur 31 Test med 20 och 40 märkningar

Figur 32 Felklassificerad bil

Figur 33 Test med 80 märkningar

Figur 34 Volkswagen

Figur 35 Test med 100 och 120 märkningar

Figur 36 Koncentrerad märkning

Figur 37 Koncentrerad märkning, 100 bilder

Figur 38 Resultat för Bil/Flyg modellen

Figur 39 Sammanfattning av resultaten

Figur 40 Otydlig katt

1. Inledning

1.1 Introduktion

Maskininläring (Machine Learning) och artificiell intelligens är för tillfället aktuella ämnen inom it-världen. Med hjälp av maskininläring kan vi t.ex. hantera mera data för analys samt programmera datorerna att självständigt analysera information och resultat för att kunna förutse framtida resultat. Eftersom maskininläring och artificiell intelligens är väldigt aktuella ämnen så har jag valt att undersöka hur användaren kan påverka maskininläringen samt resultaten.

En viktig del av studien är även användarupplevelsen, eftersom detta är en väsentlig del av både hur programmet används och hur programmet kan utvecklas för bättre resultat och användarupplevelser. Det är viktigt att förstå vilka faktorer det är som påverkar användaren i vilka stadier och vilka faktorer som är mer direkta och indirekta i processen. Då utvecklarna får en bättre förståelse av helheten så blir det lättare att se problemen och hur dessa kan åtgärdas för att utveckla en bättre produkt.

Tom Tullis och Bill Albert (2013) tar upp i sin bok *“Measuring the user experience: Collecting, Analyzing, and Presenting Usability Metrics”* hur användarnas förväntningar på användbarheten har ändrats. Android och iOS operativsystemen har höjt förväntningarna till att produkterna ska direkt vara lätt användbara för användaren. Detta är ett bevis på hur viktig det är att lägga mycket vikt på undersökningen av användarupplevelsen. (Tullis och Albert, 2013)

För att möjliggöra denna undersökning har jag fått hjälp av företaget silo.ai. Silo.ai är ett företag som Peter Sarlin har grundat och tillsammans med ett team utvecklar de olika applikationer baserade på maskininläring och artificiell intelligens. Jag har fått vara med och utveckla en bildigenkänningsapplikation tillsammans ett team från silo.ai och i denna avhandling går jag igenom vad vi har gjort och vilka resultat som nåtts. För datavisualiseringsdelen har jag använt mig av applikationen SiloBrain.

1.2 Syfte och avgränsning

Syftet med denna avhandling är att få en förståelse över hur mycket data som behöver märkas för ett pålitligt resultat samt kartlägga hur stor inverkan användaren kan ha då hen märker data i en applikation som använder sig av maskininlärnings algoritmer. Eftersom vi med silo.ai utvecklar en egen applikation för testningen så avgränsar jag testning av data endast i den applikationen. Testning av applikationen kommer att utföras av mig själv, eftersom jag då kan övervaka test data och resultaten noggrannast.

Färgblindhet är en faktor som kan påverka resultaten en hel del både i datavisualisering samt i bildigenkänning ifall klassificeringen av bilder innehåller uppdelning av färger. Jag har valt att lämna denna del borta ur min avhandling eftersom den faktorn har en för stor inverkan på ämnet, så den passar bättre i en fortsatt studie.

1.3 Problem och frågeställningar

Det jag i avhandlingen strävar att få svar på är hur mycket färdigmärkt data det krävs för att applikationen skall ge ett pålitligt resultat samt hur stor inverkan personen som märker data kan ha i resultatet.

Mina forskningsfrågor lyder enligt följande:

- Kan användaren påverka resultatet med sin insats?
- Hur stor inverkan har datan som märks i resultatet?
- Hur stor inverkan har mängden data som märks i resultatet?

1.4 Metod och insamling av data

1.4.1 Forskningsmetod

Jag har valt att använda mig av en kvalitativ metod för min forskning, eftersom jag anser den vara den lämpligaste metoden. Enligt Widerberg (2002) strävar man med en kvalitativ forskningsmetod till att klargöra egenskaperna som ett visst fenomen har och skapa en djupare förståelse på området. Bo Eneroth (2011) förklarar skillnaden mellan kvantitativ och kvalitativ som att den kvantitativa metoden är mer begränsad, eftersom man måste hitta gemensamma kvaliteter för att jämföra resultaten, medan den kvalitativa metoden söker

specifika kvaliteter som kan undersökas. Och det är precis det jag vill göra i denna forskning, eftersom jag vill kolla hur användarens påverkan fungerar i denna applikation och hur mycket mängden märkt data påverkar algoritmen. Testandet utför jag genom att själv märka olika mängder och typer av data för att sedan gå igenom resultaten och analysera dem.

1.4.2 Insamling av data

Insamling av data sker genom att skapa två stycken olika dataset med 100 bilder var. Ett set består av 50 bilder på hundar och 50 bilder på katter och det andra setet består av 50 bilder på bilar och 50 bilder på flyg. Dataseten är ihopsamlade av både mig själv och produktutvecklare i silo.ai. Dessa två dataset kommer att användas i applikationen för att göra olika test genom att märka data och sedan köra applikationen, för att se hurdant resultat som ges. Dataseten kommer att testas skilt, för att sedan i slutet jämföra resultaten.

1.5 Avhandlingens struktur

Avhandlingen är uppdelad i sex kapitel där jag går igenom grunderna i maskininlärning, användarupplevelse och användargränssnitt, datavisualisering samt resultaten för empiriska delen.

I första kapitlet presenterar jag kort varför jag valt ämnet i fråga samt vad de olika delområden handlar om och deras vikt i dagens samhälle. Därefter berättar jag kort om syftet med avhandlingen samt vilka problemområden som finns. Jag går även igenom metoden som används i avhandlingen och hur insamlingen av data utförs.

I kapitel två, tre och fyra presenterar jag teorin för de tre olika delområden som är viktiga för att utveckla applikationen samt förstå hur den fungerar. Jag börjar kapitel två med att presentera vad maskininlärning är och vad den innebär i dagens samhälle. Sedan fortsätter jag med att gå djupare in på ämnet maskininlärning och om övervakat samt oövervakat lärande, vilka är de två huvudmetoderna inom maskininlärning. Dessa två huvudmetoder innebär en massa olika algoritmer, varav jag presenterar de mest allmänna algoritmerna. Sedan berättar jag även om aktiv inlärning, som är ett aktuellt ämne och är en del av applikationen som vi

utvecklat och använder oss av. Kapitel tre består av användargränssnitt (UI) och användarupplevelse (UX) som är viktiga ämnen inom utveckling av nya applikationer eftersom de baserar sig på att koncentrerar sig på användarens behov. I kapitel fyra går jag igenom ämnet datavisualisering som är en viktig del av användarupplevelsen. Detta kapitel går jag relativt djupt in på för det var en stor del av undersökningen som gjordes för silo.ai, dåvarande Infolytika, för att hitta en optimal lösning applikationerna.

Kapitel fem består av själva forskningen och dess resultat, där jag börjar med att presentera applikationens utseende och funktioner, varefter jag går över till själva datainsamlingen och presenterandet av resultaten. Datainsamlingen är indelad i två olika delar, där första delen är en hund och katt modell med 50 bilder av vardera djur art. Andra delen består av 98 bilder på bilar och flyg, med 49 bilder av vardera grupp.

I kapitel sex går jag igenom mina egna åsikter om forskningen och drar mina slutsatser angående resultaten. Jag presenterar även mitt eget intresse om möjlig fortsatt forskning inom ämnet.

2. Maskininlärning

”The field of machine learning provides a set of algorithms that transforms data into actionable knowledge” – Brett Lantz, 2015

Det har gjorts många filmer där vi kan se hur världen går under på grund av att människan har utvecklat maskinerna mer och mer, tills de slutligen klarar sig själva och tar över världen.

Detta är vad många kopplar ihop med maskininlärning och artificiell intelligens. Som tur är så har vi inte än nått denna punkt, eftersom det krävs specifika algoritmer som programmerarna skriver för att maskinerna skall utföra olika uppgifter. De flesta har inte som mål att utveckla en artificiell hjärna, utan utvecklar ett verktyg som hjälper oss att hantera och förstå den otroliga mängden data som samlas in i dagens läge. (Brett Lantz, 2015, s.1-3)

Witten och Frank (2005) tar upp hur en vanlig person har i dagens läge många olika apparater som hen kan samla in data med. Telefonen använder vi till för att spara bl.a. numror, meddelanden och bilder, datorer för att jobba med, se på filmer, surfa och dylikt. Vissa använder sig av läsplattor istället för bärbara datorer eller telefoner. De uppräknade apparaterna är endast få i mängden som människan använder sig dagligen av för att samla på sig data. På grund av att vi i dagens läge har tillgång till stora mängder sparutrymme, både på apparaterna samt i molntjänster, så har vi möjligheten att samla på oss mera information samt skjuta upp beslutet av att kasta bort en del information. (Witten, I. & Frank, E.)

Data samlas in i många olika former. Människan samlar in data med hjälp av sina sensorer: ögon, öron, näsa, tunga och nerver. Med hjälp av talförmåga och språk har vi kunnat dela med oss av våra erfarenheter och med hjälp av skrivkunskap har vi kunnat dokumentera våra erfarenheter och observationer. Detta har utvecklats till att bland annat städerna samlar in olika information om medborgarna som t.ex. födsel och skattebetalningar mm. I dagens läge har vi utvecklat elektriska sensorer som har samma funktioner som människan (hörsel, syn, lukt mm.) men som samlar in data väldigt olika från människan, eftersom de inte påverkas bl.a. av trötthet eller andra distraktioner. Vissa säger att vi för tillfället lever i en era som kallas för ”Big Data”, men som är på sätt och vis missledande eftersom denna data har alltid funnits, men nu har vi lärt oss dokumentera denna massiva mängd av data och vi har en direkt åtkomst till den via datorer. Studierna att utveckla algoritmer som omvandlar denna stora

mängd data till användbart material är det som kallas maskininläring. (Brett Lantz, 2015, s.1-3)

Seth Mottaghinejad (2018) beskriver en maskininlärnings algoritm som en självständig uppsättning av regler som löser problem genom processering av data. Med dessa algoritmer kan datorerna lära sig ny information utan att de har blivit specifikt inlärd den nya informationen. Ett exempel på detta är ifall vi vill få reda på ifall det är en katt eller hund på bilden. I en maskininlärningsalgoritm berättar vi inte att kolla på nosen, örona och svansen för att se skillnaden på en hund och katt. En maskininlärningsalgoritm tar reda på själv hur den kan komma fram till resultatet, vilket kan vara väldigt annorlunda jämfört med vad vi skulle använda för att känna igen djuret på bilden.

2.1 Övervakat och oövervakat lärande

Övervakat lärande innebär att man använder sig av kända instanser, medan i oövervakat används obekanta instanser. Shalev-Shwarz och Ben-David (2014) beskriver skillnaden av dessa två med ett vardagligt exempel om hur man kan upptäcka skräppost som kommer till vår e-post. I övervakat lärande skulle vi utgå från data där användaren redan har märkt vilka meddelande som är skräppost eller ej, varav det är lättare att skilja på kommande meddelanden till vilken grupp de hör till. I oövervakat lärande däremot utgår vi från en stor mängd data som inte har blivit märkt från tidigare och skall försöka hitta de ”ovanliga” meddelanden. (Shalev-Shwarz & Ben-David, 2014, s.22-23)

Kotsiantis (2007) beskriver övervakat lärande med att det finns inmatningsvariabler och utmatningsvariabler varav man med hjälp av en algoritm försöker ta reda på hur man nått resultatet. Oövervakat lärande däremot har endast inmatningsvariabler och uppgiften är att försöka hitta strukturen eller distributionen för att lära sig mera om datan. Man kan även se på det som om det i övervakat lärande fanns en lärare som övervakar själva lärandet och ser till att svaren som ges är rätt, medan oövervakat lärande saknar lärare, eftersom det inte finns något rätt svar. (Kotsiantis, S.B. 2007, s.3-21)

Rokach och Maimon (2008) påminner om att övervakat lärande har två stycken huvudsakliga modeller som det är viktigt att kunna skilja på, dessa är klassificering samt regression. Klassificeringsmodellen hjälper till att dela in datan i tidigare definierade grupper, som t.ex.

hyresbetalare kan delas in i folk som betalar hyran i tid och folk som är försenade med hyran. Regressionsmodellen däremot hjälper till att kartlägga datan för att t.ex. kunna förutse saker, som t.ex. efterfrågan på en viss produkt eller priset på ett hus. Dessa tas noggrannare upp i kapitel 2.2 Klassificering samt 2.3 Regression. (Rokach och Maimon, 2008, s.1-6)

2.2 Klassificering

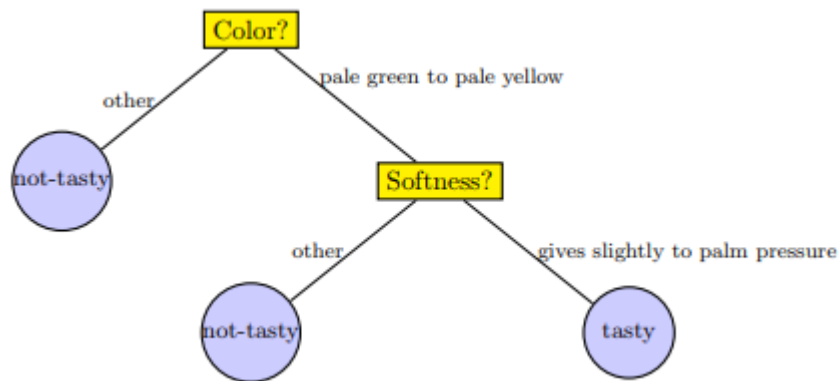
Klassificering hör till övervakat lärande, och går ut på att dela in datan i olika klasser. Det finns många olika metoder och algoritmer, ”k-Nearest Neighbor” och ”Naive Bayes” för att nämna en par stycken vanliga. Eftersom det finns många olika metoder så kan man inte enligt Kotsiantis (2007) rangordna metoderna eller prata om att någon metod är bättre än den andra, utan det gäller mera att hitta metoden som passar lämpligast till problemet som man försöker lösa, eftersom metoderna används för att undersöka flera olika sorters problem. (Kotsiantis, S.B. 2007, s.3-21)

Beslutsträd

Enligt Rokach och Maimon (2008) är ett så kallat beslutsträd (*eng. decision tree*) en vanlig metod i många olika ämnen och situationer för att göra beslut i vårt vardagliga liv, både medvetet och undermedvetet, och detta gäller även i klassificering i maskininläring. Beslutsträd används både i klassificering samt regression, och då den används i klassificering använder man sig ofta av namnet klassificeringsträd, för att förtydliga vilken sorts model som används. (Rokach och Maimon, 2008, s.5-6)

Shalev-Shwarz och Ben-David (2014) tar upp ett exempel på hur man kan ta reda på ifall en papayafrukt är mogen eller inte med hjälp av ett beslutsträd. Exemplet illustreras i bild 1 ”Beslutsträd” nedanför. Man börjar med frågan ”vilken färg har frukten?”. Svartalternativen är ”ljusgrön till ljusgul” eller ”annan färg”. Ifall svaret är ”annan färg” så är frukten troligen inte mogen vilket leder till att vi inte behöver mera information. Ifall svaret är ljusgrön så ställer vi oss följande fråga som är ”mjukhet?”. Svartalternativen ”ytan ger lite efter då man trycker på den” vilket leder till att frukten troligen är mogen och ”annat” vilket leder till att

frukten troligen inte är mogen. Denna frågekedja kan fortsätta så långt vi vill, så länge som det finns relevanta följdfrågor. (Shalev-Shwarz & Ben-David, 2014, s.250-251)



Figur 1 Beslutsträd (Shalev-Shwarz & Ben-David, 2014, s.250)

Nearest Neighbor

Nearest Neighbor, fritt översatt till ”närmaste grannen”, är en metod som hjälper att klassificera omärkt eller o grupperad data till en klass som den är mest lik, skriver Lantz (2015). Detta är en väldigt simpel men användbar metod, speciellt i fall där koncepten är väldigt svåra att definiera men lätta att tolka när man ser dem. Denna metod kan beskrivas med en av de lättaste algoritmerna som används i maskininlärning, den så kallade k-NN algoritmen. Kort och lätt beskrivet fungerar algoritmen enligt följande: bokstaven ”k” står för en variabel och algoritmen går ut på att den mäter avståndet från variabeln ”k” till de övriga variablerna. När avstånden är mätta kan den gruppera variablerna som ligger närmast varandra ihop. (Brett Lantz, 2015, s.66-69)

Kotsiantis (2007) kallar k-NN för ”Lazy-learning”algoritm eftersom denna kräver mindre uträkningstid under själva träningsfasen, men är långsammare i själva klassificerings skedet. För att kunna visualisera sig hur själva klassificeringen och räknandet av avstånd går till, så rekommenderar han att man skall tänka sig variablerna som punkter i ett 3 dimensionellt diagram. Vid själva klassificeringen har inte punkternas värden desto större betydelse, eftersom själva uträkningen går ut på att minimera avståndet mellan variablerna som hör till samma grupp och maximera avståndet mellan grupperna. (Kotsiantis, S.B. 2007, s.9-12) För att kunna mäta avstånden mellan variablerna, så behövs det en avståndsfunktion som eller formel som kan räkna ut likheten mellan variablerna. Lantz (2015) skriver att k-NN

algoritmen traditionellt använder sig av "Euclidean distance" som mäter likheten rakaste och kortaste vägen mellan två stycken variabler, utan att ta andra variabler i beaktande. En annan vanlig avståndsfunktion är "Manhattan distance", där de övriga variablerna tas i beaktande och avståndet räknas från variabel till variabel. (Brett Lantz, 2015, s.69)

Naive Bayes

Naive Bayes-metoden är en gammal metod som knippar sig med statistik och sannolikhetslära. Sannolikheten att någonting händer kan mätas med värden 0 till 1, som står för 0 procent och 100 procent och allting däremellan. Klassificering enligt Naive Bayes-metoden innebär att man använder träningsdata för att räkna ut sannolikheten på tidigare utfall baserat på bevis från tidigare funktionsvärden. Därefter, då man räknar ut sannolikheten för nya funktioner, jämför man tidigare utfall mot nya för att förutsäga den mest troliga klassen för de nya funktionerna. Metoden är pålitligare, eller noggrannare, ju mer övningsdata som samlats in. (Brett Lantz, 2015, s.89-98)

Enligt Lantz (2015) används Naive Bayes-metoden för bland annat att upptäcka intrång eller avvikelser i datornät samt för att diagnostisera medicinska tillstånd. Det är en simpel men effektiv metod som kan tillämpas i många olika fall. Metoden används även för textinlärningsalgoritmer som är användbara för att klassificera dokument eller artiklar i tidningar mm. Naive Bayes-metoden behandlar alla attribut som självständiga, vilket gör att det är en effektiv metod för att behandla dokument med ett stort antal attribut eftersom parametrarna för attributen kan läras separat, skriver McCallum och Nigam. Detta visade sig även stämma in i Vural och Göks (2017) undersökning, där de med hjälp av Naive Bayes-metoden och beslutsträd-metoden försökte förutsäga de kriminella bland alla som var misstänkta av olika brått. Naive Bayes-metoden visade sig vara betydligt effektivare än beslutsträd-metoden, speciellt i fall med mera information, eftersom Naive Bayes-metodens algoritmer är mer anpassade för större data, medan beslutsträd-metodens beräkningar blir väldigt komplicerade med större mängder data.

2.3 Regression

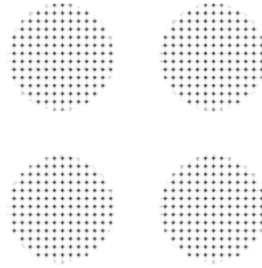
Lantz (2015) definierar regression som en metod att specificera förhållandet mellan en enda numerisk beroende variabel och en eller flera numeriska oberoende variabler. Det finns flera

olika metoder att göra detta på, varav de simplaste metoderna antar att förhållandet mellan beroende och oberoende variablerna följer ett rakt sträck. En välkänd metod heter ”simple linear regression”, som översätts rakt till enkel linjär regression. Som Bryan Burnham (2015) skriver i sin bok ”*Fundamental Statistics for the Behaviour Sciences*”, så används linjär regression mycket inom statistik för att förutspå variabler på basis av variabler som vi redan har. Han påpekar att komma ihåg att förhållandet mellan värdena i linjär regression är så gott som aldrig perfekta, vilket betyder att resultatet av de förutspådda resultaten inte heller är med exakt noggrannhet.

Enligt Lantz (2015) används regression analys för bland annat undersökningar av hur befolkningen samt individerna varierar mätt i deras olika egenskaper, för användning av olika vetenskapliga forskningar inom ekonomi, sociologi, psykologi bland många andra ämnen. Regression analys används även i undersökningar där man försöker hitta sambandet mellan en händelse och orsaken eller lösningen till denna, som t.ex. säkerhetstest eller marknadsundersökningar. (Brett Lantz, 2015, s.171-176)

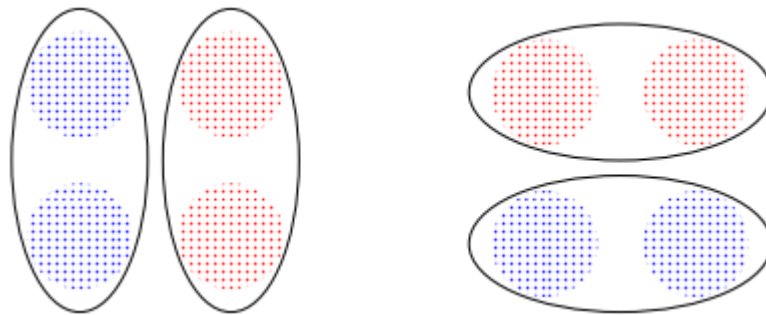
2.4 Klustring

Så kallad klustring (*eng. clustering*) tillhör oövervakat lärande, vilket innebär att det inte direkt finns något rätt svar. Vad klustring går ut på är att gruppera objekt så att liknande eller näraliggande objekt grupperas ihop. Det går att tillämpa nearest neighbor-metoden i klustring, vilket är enligt Sibson (1973) den äldsta metoden för analys av kluster. Detta är ändå inte alltid lika lätt som det låter, eftersom det kan vara en tolkningsfråga vilka objekt som hör till vilken grupp. För att illustrera detta visuellt har Shalev-Shwarz och Ben-David (2014) gjort 3 bilder, som visas nedan i Figur 2 och Figur 3, som förklarar problemet tydligare. I figur 2 har vi vår data som skall delas in i 2 olika klusters. (Shalev-Shwarz & Ben-David, 2014, s.308-309)



Figur 2 Obehandlad data (Shalev-Shwarz & Ben-David, 2014, s.308)

Figur 3 visar 2 stycken lika exemplar som båda är lika rätt, men datan i de 2 olika klustren skiljer sig märkvärdigt.

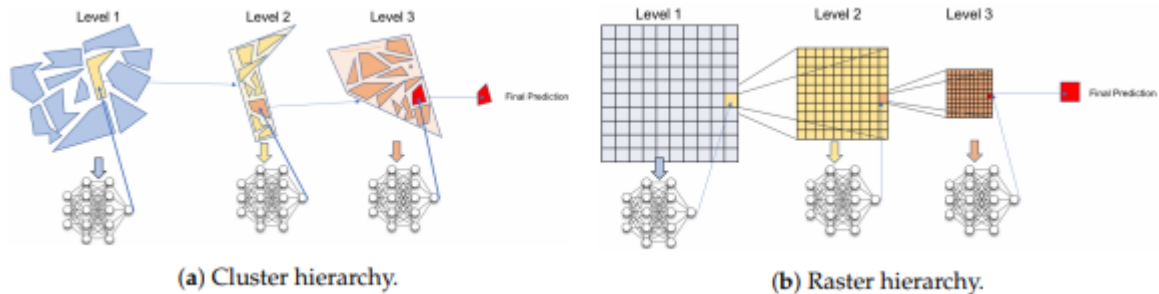


Figur 3 Klustrad data (Shalev-Shwarz & Ben-David, 2014, s.309)

Lantz (2015) skriver att klustring används främst för att upptäcka ny kunskap istället för att förutse saker. Vad man söker efter är att hitta mindre grupperingar och olika mönster av relation i större mängder data. Typiska exempel där klustring används är då man delar kunder i olika demografiska segment eller köp beteende, för att upptäcka anomalt beteende och för att underlätta läsandet av extremt stor data. (Brett Lantz, 2015, s.286-288)

Urner, Bucher, Yang och Jonietz (2018) däremot har använt klustring som en del av sin studie angående användarens förutsägning av nästa plats. I forskningen samlade de in spåringsdata av över 700 applikations användare. Metoderna som användes var raster och klustring, varav klustring visade sig vara effektivare i detta område eftersom den kombinerar närliggande platser som hör ihop enligt koordinaterna latitud och longitud, medan raster delar in områden i små rutor, vilket kan ge flera olika svar för en och samma plats, eftersom geografiska platser

sällan är i form av en kvadrat. I figur 4 visas ett exempel på skillnaden mellan (a) klustermetoden och (b) raster-metoden.

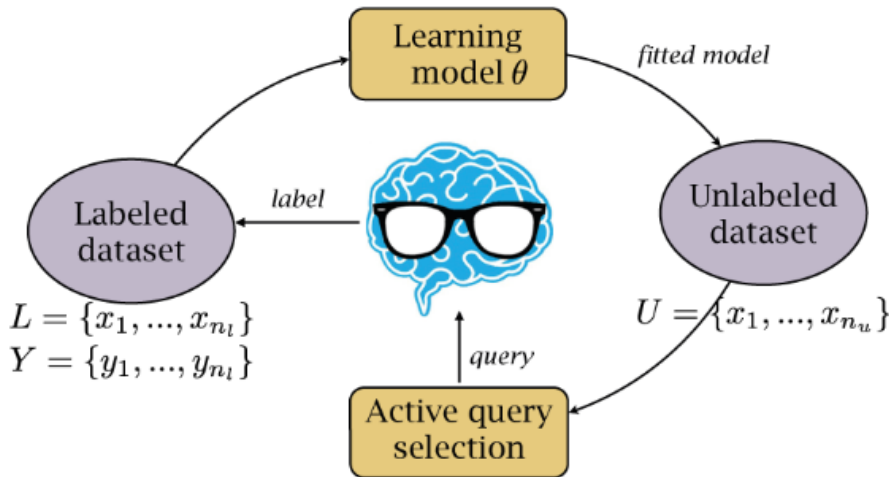


Figur 4 Skillnaden mellan kluster- och raster-metoden (Urner, Bucher, Yang & Jonietz, 2018)

För att sedan kunna använda denna spåringsdata för att förutse nästa plats innebär en del analyserande av användarens beteende. Som indata har Urner m.fl. (2018) använt sig av när (tid, dag, vardag eller veckoslut) platsen har besökts, avstånd och liknande vägar som gjorts tidigare. (Urner, Bucher, Yang & Jonietz, 2018)

2.5 Aktiv inlärning

Med ”vanlig” maskininlärning strävar man till att försöka hitta den bästa eller effektivaste algoritmen eller metoden för att behandla data som används. Med aktiv inlärning däremot strävar man efter att minska mängden data som måste behandlas genom att ställa frågor till användaren. I figur 5 ser vi en illustration av Computer Vision Lab (CVLAB) om hur aktiv inlärningsmodellen kan se ut. Modellen börjar med några exemplar av färdigt annoterad data (L & Y) som används till inlärningen av modellen θ . Därefter kommer det in ny data U som inte ännu behandlats, varifrån algoritmen plockar ut få valda värden som användaren sedan skall annotera. Värdena som plockas ut bestäms av algoritmen och metoden som används, t.ex. klassificeringsmetoden. (Konyushkova, Sznitman & Pascal, 2017)



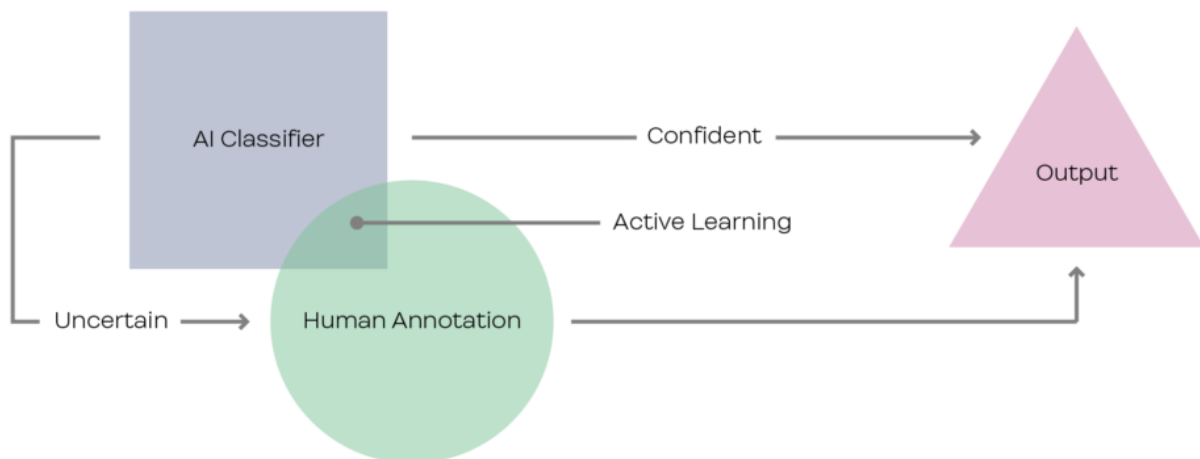
Figur 5 Aktiv inlärning (Computer Vision Laboratory CVLAB URL: <https://cvlab.epfl.ch/page-123199-en.html>)

Både Burr Settles (2009) och Tong och Koller (2001) beskriver aktiv inlärning som en effektiv metod där algoritmen kan välja från vilken data den skall lära sig. Detta innebär, enligt Settles, att inlärningsmetoden blir mer effektiv eftersom algoritmen måste gå igenom en mindre mängd data jämfört med andra algoritmer som används i övervakat lärande. I övervakat lärande kräver algoritmerna ofta hundratals eller tusentals instanser för att kunna fungera effektivt och noggrant, vilket kan i vissa fall vara både svårt och tidskrävande. (Burr Settles, 2009)

Tong och Koller (2001) tar upp metoden pool-baserad (eng. Pool-based) aktiv inlärning som Lewis och Gale (1994) introducerade för klassificering i aktiv inlärning. I denna metod har den som lär sig tillgång till en "pool" eller grupp av omärkt data varifrån hen kan med hjälp av en viss mängd instanser märka eller indela data till vilken grupp den hör. De tar upp ett vardagligt exempel på detta som är filtrering av e-post. Användaren kan göra en skräddarsydd spamfiltrering genom att gå igenom äldre e-poster och anmärka dessa till antingen spam eller icke spam. Efter att gått igenom en viss mängd så har användaren gjort en skräddarsydd spamfiltrering åt sig själv med hjälp av aktiv inlärning metoden, där algoritmen på basis av tidigare annoterad data kan själv räkna ut till vilken kategori inkommande e-post tillhör. (Tong & Koller, 2001)

2.6 Human in the loop

Human-in-the-Loop, direkt översatt till ”människan i loppet”, är ett begrepp som används i artificiell intelligens och är en del av aktiv inlärning. Figur 6 är en bild från Figure Eights artikel ”What is Human-in-the-Loop Machine Learning” där de med hjälp av bilden förklarar vad begreppet ”Human-in-the-Loop” innebär. Allting börjar vid kuben ”AI Classifier” där algoritmen kollar ifall den känner igen data eller ej. Ifall algoritmen kan ge ett säkert svar så går data raka vägen vidare till triangeln ”Output”. Ifall algoritmen inte känner igen data så skickar den en frågeställning till cirkeln ”Human Annotation” (direkt översatt ”mänsklig märkning”) där användaren skall bekräfta vad för sorts data det är frågan om, varefter den går vidare till ”Output”. Det är den delen som gör processen till ”Human-in-the-Loop”, eftersom användaren är med i processen och hjälper algoritmen.



Figur 6 Human in the loop (Figure Eight URL: <https://www.figure-eight.com/resources/human-in-the-loop/>)

3. Användargränssnitt (UI) och användarupplevelse (UX)

Användargränssnitt, som förkortas UI enligt engelska namnet User Interface, och användarupplevelse, som förkortas UX enligt engelska namnet User Experience, är två ämnen som går hand i hand då vi planerar t.ex. en ny applikation. Enligt Q Manning i Code Magazine är det svårt att komma på en helt ny applikation i dagens läge eftersom konkurrensen är stor. Det som däremot skiljer på applikationerna är användargränssnittet och användarupplevelsen. I dagens läge måste applikationen fånga användarens intresse, oberoende vad innehållet av applikationen är.

Helga Moreno (2014) skriver i sin artikel "The Gap Between UI and UX Design – Know the Difference" att skillnaden mellan UI och UX kan förklaras enligt följande; något som ser bra ut men är svårt att använda är ett exempel på bra UI men dålig UX, medan något som är väldigt användbart men ser dåligt ut är exempel på bra UX men dålig UI. Ett annat bra exempel tas upp av Keinonen, Mattelmäki, Soosalu och Säde (1998) där UI kommer mer fram i själva design skedet då produkten görs åt användaren medan UX kommer mer fram i uppföljningen då vi observerar användarens upplevelse av produkten.

3.1 Användargränssnitt (UI)

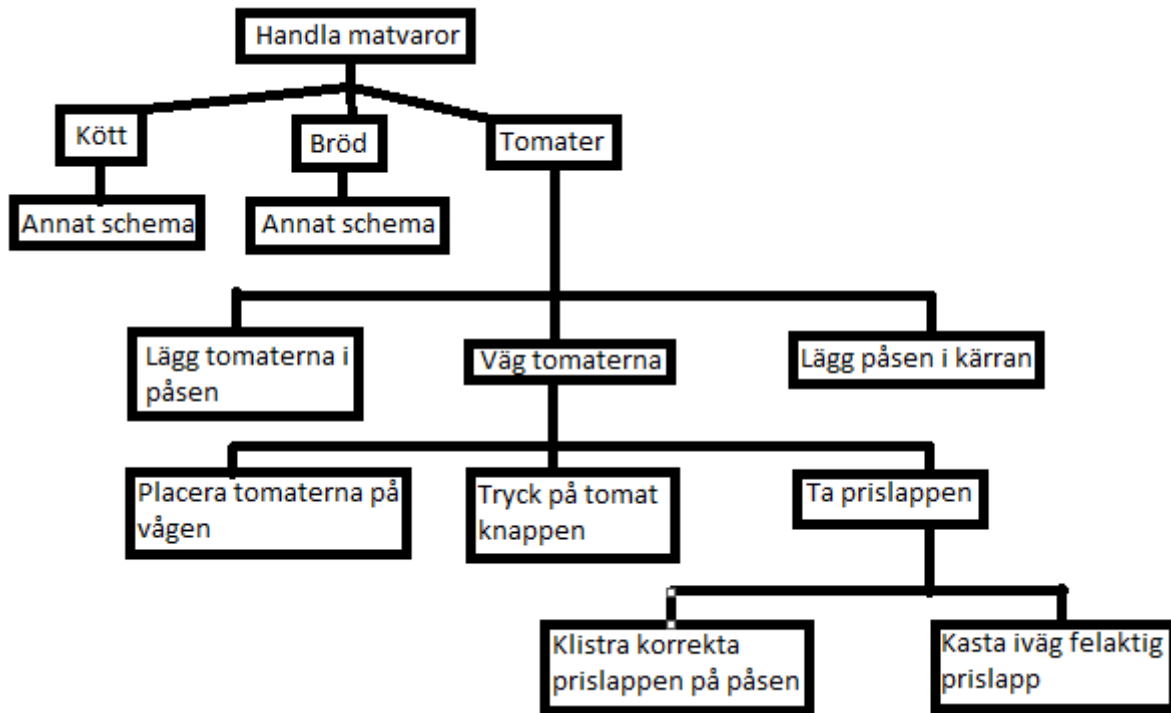
Scenarier

Scenarier är ett sätt att visualisera framtida interaktioner mellan användaren och produkten. Dessa scenarier är endast något som skall visuellt förklara vem användaren är, vilka deras behov är, hur deras upplevelse av användningen ser ut, miljön, omständigheterna, vad uppgiften är, målet och hur målet skall uppfyllas. Dessa scenarier kan göras på alla tänkbara sätt hur en berättelse kan berättas, t.ex. som serier med bilder och bildtexter. (Keinonen, Mattelmäki, Soosalu och Säde, 1998)

Användargränsschema

Användargränsschema, även känt som processflödesschema eller hierarkisk uppgiftsanalys, är ett effektivt sätt att visuellt presentera en stor mängd information samlat i en bild eller diagram, som t.ex. logiken i strukturen samt navigationen i ett interaktivt användargränssnitt.

I figur 7 presenteras ett simpelt exempel på hur en hierarkisk uppgiftsanalys kan se ut, i detta fall illustreras hur processen att handla en tomat går till och vilka delar som ingår i den. Dessa diagram är något som inte ska förväntas förstås av andra än själva produktutvecklarna. (Keinonen, Mattelmäki, Soosalu och Säde, 1998)



Figur 7 Hierarkisk uppgiftsanalys (Keinonen, Mattelmäki, Soosalu och Säde, 1998, egen översättning)

3.2 Användarupplevelse (UX)

Genom att studera användarupplevelsen kan även ibland resultera i att rädda liv. Anthony Andre (2003) gjorde en forskning om automatiska externa defibrillatorer (AED) som används för att återuppliva folk som fått hjärtstopp. Dessa maskiner finns utplacerade på olika allmänna folkplatser, som t.ex. flygplatser, shoppingcenter mm. I sin studie använde sig Andre av 4 stycken olika defibrillatorer och 64 stycken deltagare som aldrig använt en defibrillator tidigare. Deltagarna delades upp i fyra grupper och deras uppgift var att en i taget gå in i ett rum och försöka återuppliva en docka. Endast två grupper klarade sig felfritt, medan en annan grupp hade 25% som aldrig fick fram en elchock och vissa hade problem med andra anvisningar. Denna studie ledde till att företagen med produkter som inte fungerat felfritt

designade om produkten så att den blev lättare att använda. (Tullis, T. & Albert, B. 2013. s.5-6)

Enligt Tullis och Albert (2014) har användarupplevelsen (*eng. user experience*), förkortat UX, tre huvudsakliga karakteristiska drag som definierar själva användarupplevelsen. Den första är att det måste finnas en användare inblandad. Utan en användare finns det ingen användarupplevelse, vilket leder till att resultatet blir mer en undersökning av åsikter och preferenser istället för att se hur användaren gör och reagerar. Den andra är att användaren måste interagera med en produkt eller ett system. Själva produkten eller systemet behöver inte definieras på något speciellt sätt, enda kriteriet är att användaren måste på något sätt samverka med det. Den tredje punkten är att användarnas upplevelse intresserar och kan iakttas eller mätas. (Tullis, T. & Albert, B. 2013. s.4)

3.2.1 Mätning av användarupplevelsen

För att kunna mäta eller värdera någonting så behöver vi något slags mått. Det finns olika mått standardiserade för oss för att kunna mäta olika saker, som t.ex. längd, höjd, hastighet, vikt. För att mäta längder så använder vi oss t.ex. av meter och för tid bl.a. sekunder. Oberoende av vilket redskap vi använder oss av så är både meter och sekund alltid samma, eftersom de är färdigt standardiserade. (Tullis, T. & Albert, B. 2013. s.6-7)

Beroende på vilken bransch eller område man är så är man intresserad av olika saker att mäta. På bilbranschen t.ex. är man intresserad av bilens hästkrafter och bränsleförbrukningen medan hemma är vi intresserade av vad vågen visar eller på vilken grad termostaten är på mm. Helt som på alla andra branscher så har man även specifika intressen då man mäter användarupplevelsen. Här är vi intresserade av att mäta hur en uppgift lyckats, effektivitet, tillfredsställelse, fel och mycket mera. Många av dessa har inga standardiserade måttenheter, men då man använder sig av samma mätning för alla användaren så kan resultatet jämföras och därmed får vi ett resultat. Det som alla användarupplevelsers mätningar har gemensamt är att allting måste kunna iakttagas på något sätt, antingen indirekt eller direkt. Det som också är gemensamt för alla är att de måste vara kvantifierbara och att mätningen som är relaterad till användarupplevelsen presenteras i numeriskt format, som t.ex. procentenhet eller dylikt. (Tullis, T. & Albert, B. 2013. s.7-8)

Varför vill man mäta användarupplevelsen?

Orsaken till varför man vill mäta användarupplevelsen är för att vi ska få information som stöder våra framtida utvecklingsbeslut. Vi vill få svar på frågor som t.ex. kommer användaren att rekommendera produkten vidare eller är den nya produkten mer effektiv än förra versionen eller vilka är de mest relevanta problemen för användaren. (Tullis, T. & Albert, B. 2013. s.7-8)

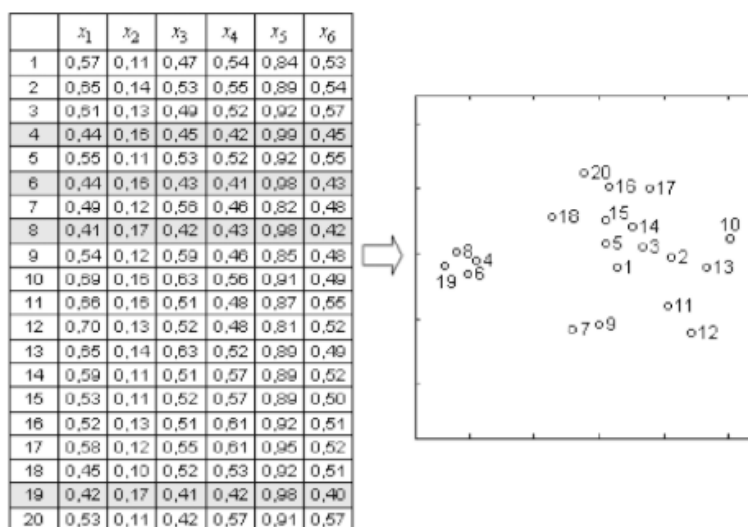
Hur använder man data

Det finns två stycken olika sätt att använda data som man samlat under studien. Ett sätt är formativ användbarhet och den andra är summativ användbarhet. I formativ användbarhet använder man sig av data och resultat för att förbättra produkten under utvecklingsskedet, för att få fram det ultimata resultatet. I summativ användbarhet väntar man på att produkten är klar och därefter samlar data för att evaluera slutresultatet. (Tullis, T. & Albert, B. 2013. s.42-47)

4. Datavisualisering

I min kandidatavhandling Peter Hellberg (2017) *Användning av färger och grafer i datavisualisering* gjorde jag en undersökning i datavisualisering åt silo.ai, dåvarande Infolytika, där vi försökte hitta optimala lösningar till applikationer som silo.ai höll på att utveckla. Bildigenkänningsapplikationen som används i undersökningen i denna avhandling har också fått sin början från undersökningen jag gjort tidigare för silo.ai, och därmed har jag bestämt mig för att tillägga teorin bakom datavisualiseringen även till denna avhandling.

Som Assam och Evergreen (2013) definierar datavisualisering så förlitar den sig på tre stycken olika kriterier. De beskriver det som en mängd kvantitativ eller kvalitativ data som genom olika processer förvandlas till en bild som är lättare för läsaren att tolka. Detta stöds även av Dzemyda, Kurasova och Medvedev (2007) i deras forskning. Figur 8 nedanför ”Visualization power” kan vi se ett av deras exempel, där de illustrerar hur data kan klargöras tydligare i ett diagram jämfört med en tabell. (Peter Hellberg, 2017)



Figur 8 Effekten av visualisering (Dzemyda, G., Kurasova, O. & Medvedev, V. (2007) s.26)

Användarkrav, samarbetsverktyg samt datorstöd är de tre kraven som Fayyad, Grinstein och Wiese (2002) anser vara väsentliga vid planering av datavisualisering. För att kunna sammanfatta och presentera det som användaren vill se så måste vi förstå dess krav på förhand. Vi måste ta reda på vilket sätt som är logiskt för användaren att läsa denna data, vad är det som är relevant för användaren och på vilket sätt det hör ihop. På grund av personers olika nivåer på kunskaper så måste systemen byggas upp så att de är lättillgängliga och

lättanvändbara för användaren eller användarna. Samarbetsverktyget, som är den tredje punkten, kommer i användning då vi hamnar behandla stora mängder data som inte kan behandlas ensam. Någon sorts kommunikationsverktyg är behändigt i dessa fall, för att underlätta genomgången och behandlandet av materialet. (Fayyad, Grinstein, Wierse, 2002, s.87-92. Peter Hellberg, 2017)

Enligt Saari (2011) har ungefär 8 procent av männen och 0,5 procent av kvinnorna i Finland någon sorts färgblindhet. Den vanligaste färgblindheten är röd-grön, vilket kan orsaka problem med hemsidor och spel bl.a. för att den vanligaste färgkoden som används är RGB som står för röd grön blå. Det som dock förbättras med färgblindhet är mörkerseendet och detaljseendet i monokroma miljöer, enligt Sacks (1995). Sacks skriver att personer med färgblindhet har lättare att urskilja figurerna i enfärgade miljöer. Detta är dock ett ämne som jag inte kommer att gå djupare in på i denna avhandling, eftersom ämnet blir för brett.

4.1 Användning av färger

Tuula Nieminen (2009) skriver i sin bok "*Visuaalinen markkinointi*" att människan kan skilja på 128 stycken olika nyanser som kan modifieras med 16-23 stycken renlighetsgrader. Hundratals olika alternativ kan skapas med hjälp av ljusstyrkan. Den psykologiska effekten som olika färgkombinationer kan åstadkomma kan läras genom att studera olika färger och dess klassificering. Till exempel kan en "visuell kakofoni" (eng. visual excitement) åstadkommas genom att kombinera olika ljusa färger. (Tuula Nieminen, 2009, s.187. Peter Hellberg, 2017)

Semantiskt resonant (eng. semantically resonant) är en term som används vid referering av färgval för tidigare inlärd koncept enligt Sharon Lin (2013). Enligt Lin används semantiskt resonanta färgpaletter för att aktivera den delen av hjärnan där informationen proceseras mer automatiskt, vilket resulterar i att användarens läsning av data effektiveras. Förutom att själva läsningen av data försnabbas så kommer också användaren bättre ihåg det som hen läst, eftersom semantiken förbättrar det visuella minnet. (Sharon Lin, 2013, s.401-410. Peter Hellberg, 2017)

I min kandidatavhandling *Användning av färger och grafer i datavisualisering* (Peter Hellberg, 2017) testade jag användningen av semantiskt resonanta färgval med en applikation

som silo.ai hade, som hette RiskDashBoard och innehöll data om länders ekonomiska inflation. I testet valde jag att gå med färger som kopplades ihop med ländernas flaggor, som t.ex. röd färg för Kina, blå färg för Sverige osv. Detta fungerade bra så länge som variablerna är frångskilda, som i stapeldiagram t.ex., men direkt då variablerna börjar korsas varandra eller går parallellt nära varandra, så fungerar inte metoden så bra. I dessa fall rekommenderar jag att man håller sig till färger som är tydligt olika för att underlätta läsandet. (Peter Hellberg, 2017)

Stephen Few (2008) skriver om användning av färger i sin artikel Visual Business Intelligence Newsletter. Tydligheten och betydelsen på visualiseringen kan ändras märkvärdigt med hjälp av smarta färgval. Few räknar upp några regler för användning av färger vid visning av data som han anser vara viktiga. De två första reglerna behandlar valet av bakgrundsfärgen. För att skapa en tydlig färgupplevelse som tillfredsställer ögat så måste man ta i beaktande färgkombinationerna på de olika elementen som är framme, som t.ex. bakgrunden på tabellerna eller ramarna runtomkring. Tredje och fjärde reglerna handlar om användning av färger och valet av dessa. Enligt Few (2008) lönar det sig endast använda färger vid tillfällen som man vill uttrycka något speciellt eller uppmärksamma läsaren med något speciellt. (Stephen Few, 2008, s.2-7. Peter Hellberg, 2017)

På en allmän nivå rekommenderar Few (2008) att man håller sig till enkla och tydliga grafer istället för att försöka skapa något futuristiskt med t.ex. 3D-diagram. För diagrammens axlar och värden som visas vid axlarna, med andra liknande komponenter som inte tillhör data, rekommenderar Few (2008) att man använder sig av neutrala färger för att hålla dem tydliga och åtskilda för att hålla koncentrationen på det väsentliga. För diagrammens linjer rekommenderas tunna gråa linjer och för bakgrunden vit färg. (Stephen Few, 2008, s.8-12. Peter Hellberg, 2017)

4.1.1 RGB

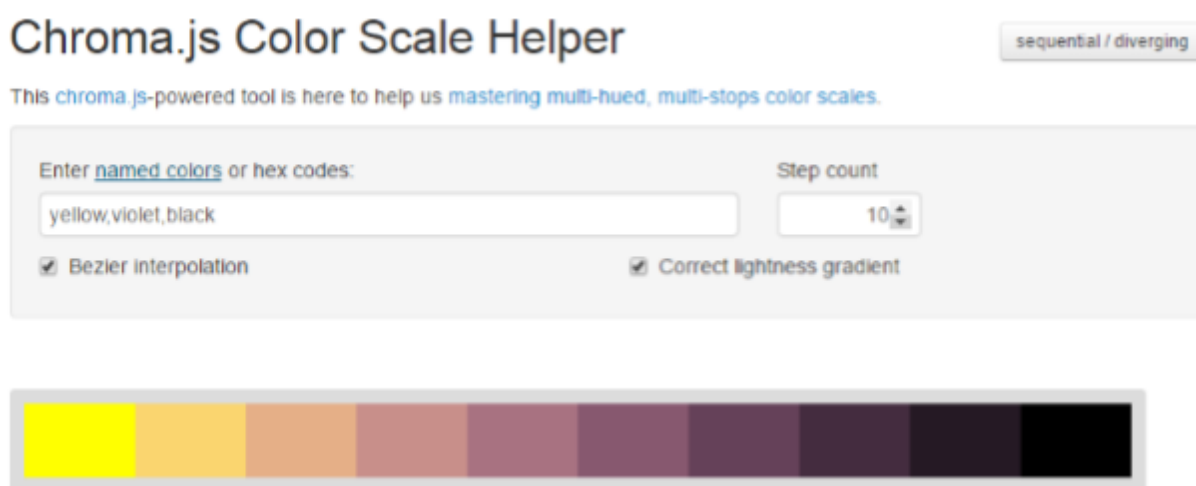
Den mest allmänna färgkoden som används för RGB är RGB (0, 0, 0) som är helsvart och RGB (255, 255, 255) som är helvit, denna används även i Javascript som silo.ai använder sig av. I graferna som användes i SiloBrain applikationen, och överlag i digitala bilder, använder man sig av färgkoder som kallas RGB-koder. RGB står för röd (*eng. red*), grön (*eng. green*) och blå (*eng. blue*). Dessa kan visas på olika sätt på skärmen beroende på hurdan skärm det är

frågan om. Äldre skärmar som t.ex. CRT-skärmar produceras färgerna genom fosforutsläpp med hjälp av röda, gröna och blåa färgstrålar, medan i mer moderna skärmar som t.ex. TFT-skärmar och projektorer lyser ljuset genom färgade filter. (Maureen Stone, 2003, s.44. Peter Hellberg, 2017)

4.1.2 Rekommenderade färgpaletter

Maureen Stone (2006) rekommenderar användning av färdiga färgpaletter vid uppdelning av färg, som t.ex. ”ColorBrewer 2.0”. Denna palett är designad för vit bakgrund och hjälper vid planering av färger, eftersom den är färdigt utvecklad och det tar lång tid att börja utveckla själv en egen. Stone anser det vara bäst att dela upp färgerna enligt hurdan funktion man vill åstadkomma, som t.ex. olika kontrastbyten för att markera något speciellt i visualiseringen, värdekontroll för kontrasten för att öka tydligheten samt analogi för gruppering av material. (Maureen Stone, 2006, s.6-9. Peter Hellberg, 2017)

Ett bra verktyg för att hitta flera färger vid tillfällen där mängden av entiteter ökar, är Chroma.js Color Scale Helper som vi kan se i figur 9. Verktøget kombinerar färger som du kan välja ut och ger färdiga RGB färgkoder för detta. Med hjälp av stegräknaren till höger i Figur 9 väljer användaren mängden färgkombinationer som verktøget skall producera.



Figur 9 Color Scale Helper (Chroma.js Color Scale Helper URL:

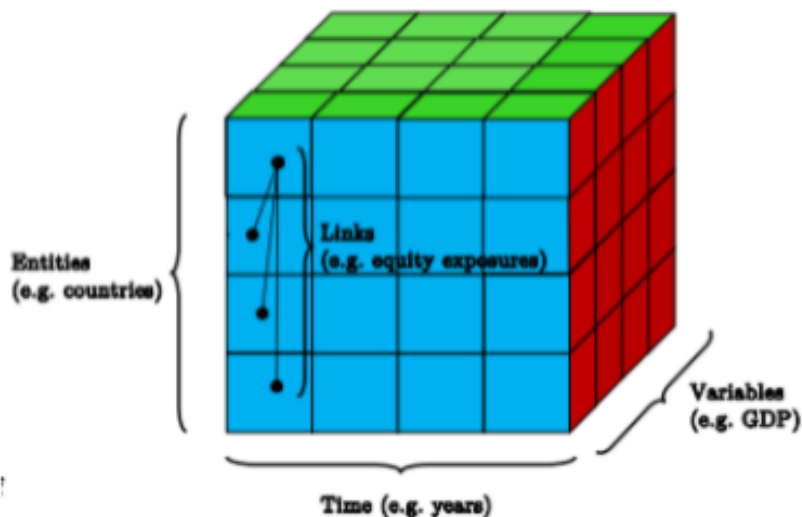
<http://gka.github.io/palettes/#colors=yellow,violet,black|steps=295|bez=1|coL=1>)

Med hjälp av att blanda färgerna från färgpaletter i Scale Helper kan vi åstadkomma fler färger än vi kan använda. Enligt Nieminen (2009) så kan människan endast skilja 128 nyanser, vilket gör mängden som Color Scale Helper erbjuder onödigt för människoögat, men den kan vara användbar för användning i bildigenkänningsapplikationer eller maskininlärning överlag, eftersom datorn är kapabel att känna igen betydligt fler nyanser än människan.

4.2 Användning av grafer

Stephen M. Kosslyn (2006) beskriver i sin bok *Graph Design for the Eye and Mind* hjärnans funktion och dess begränsningar. Enligt Kosslyn kan dessa begränsningar undvikas med hjälp grafer, eftersom hjärnan har lättare att uppta informationen som är presenterad med hjälp av väluppbyggda grafer och tabeller. (Stephen Kosslyn, 2006, s.17- 20. Peter Hellberg, 2017)

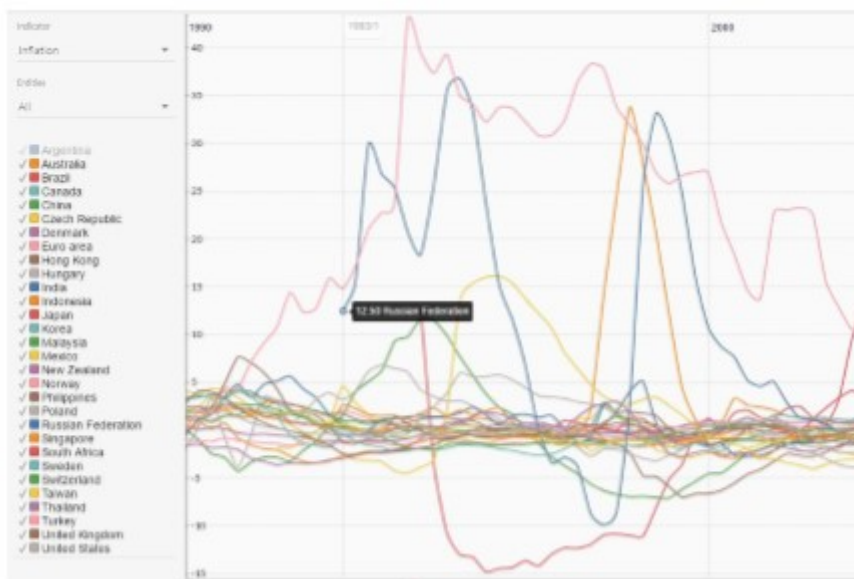
Peter Sarlin (2013) förklarar visuellt med hjälp av en data kub med fyra dimensioner hur mångdimensionell data kan uppdelas, som vi kan se i figur 10. Som vi kan se i figur 10 nedan så har vi entiteterna på axeln till höger som går lodrät, som i detta exempel är länder (eng. countries). Vågräta axeln längst nere i mitten på bilden består av tid, som i detta fall är år (eng. years). Den tredje axeln längst till höger på bilden består av variabler, t.ex. länders BNP värde (eng. GDP). Den fjärde dimensionen som vi kan se i mitten av kuben på bilden är förbindelser, t.ex. olika indikatorer för riskfaktorer (eng. equity exposures). (Peter Sarlin 2013, s.68-69. Peter Hellberg, 2017)



Figur 10 Data kub (Peter Sarlin (2013) Bild 4.1 s.69)

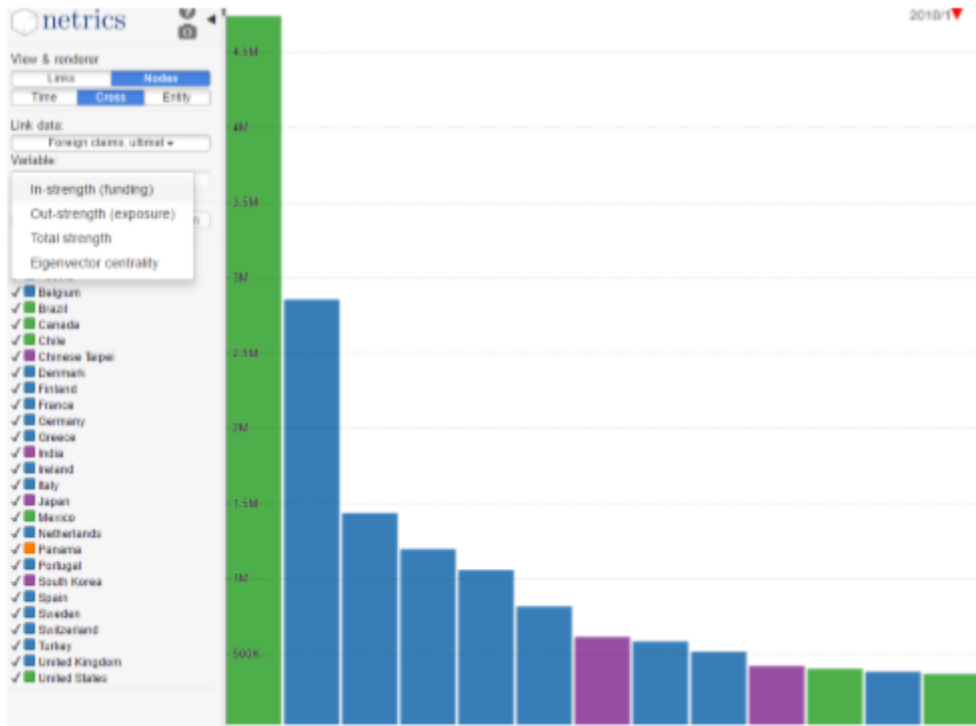
Orsaken till varför det är bra att använda sig av kuben är för att den är lätt att dela upp i olika sidor som underlättar att visualisera hur data kan användas. Blåa sidan, som är längst fram på kuben i figur 10, består av flera entiteter för en viss tidsperiod för en viss variabel. På högra kanten av kuben har vi den röda sidan, som representerar data av flera entiteter och variabler på en viss bestämd tid. Den gröna sidan ovanpå kuben består av flera variabler inom olika tider för en entitet, medan den sista dimensionen består av sammanhanget mellan entiteterna under en viss tid. (Peter Sarlin 2013, s.68-69. Peter Hellberg, 2017)

Visualiseringen av vilken sorts data som skall användas, och till vilken typ av graf, underlättas med hjälp av kubens indelning. Sarlin (2013) anser tidsserier passa både blåa och gröna sidan i kuben i figur 10. Eftersom syftet är att se hur data har ändrats under en viss tidsperiod, så är tidsseriegrafen en bra graf att använda sig av på grund av sin tydlighet, som vi kan se i figur 11. Övriga alternativ som jag kunde tänka mig före denna typ av data kunde vara ett spridningsdiagram (eng. Scatter Plot) eller ett Gantt diagram, men de är inte lika tydliga som en tidsseriegraf. (Peter Hellberg, 2017)



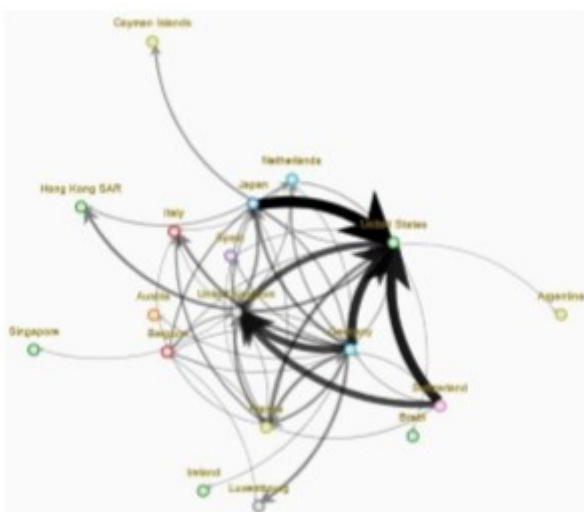
Figur 11 Tidsseriegraf (SiloBrain, Infolytika, 2017)

Eftersom röda sidan av kuben består av grupper som vanligen innehåller en stor mängd information, både entiteter och variabler, så kräver den även att diagrammet ska vara anpassat för detta. Stapeldiagram, som vi kan se i figur 12, är ett bra diagram som jag fick bekanta mig med i applikationen Netrics som Silo.ai hade. I denna applikation har de lagt till en rullgardinsmeny för att underlätta användarens val av variabler. Ett exempel på denna lösning ser vi i figur 12 nedan, som är från Netrics applikationen. Informationen visas tydligt och det är lätt för användaren att förstå vad det är som visas. På högra sidan av bilden i figur 12 ser vi själva stapeldiagrammet som visar data och till vänster om denna har vi en stapel där vi kan göra en hel del olika val för att ändra på vad som skall visas. Högst uppe i stapeln kan användaren välja vilka grupper som skall kombineras i kuben, och under detta kan själva variablerna väljas. Som bonus har silo.ai lagt till under en meny där användaren kan kryssa vilka länder som ska tas i beaktande i resultaten som visas. (Peter Hellberg, 2017)



Figur 12 Stapeldiagram med rullgardinsmeny (Netrics, Infolytika, 2017)

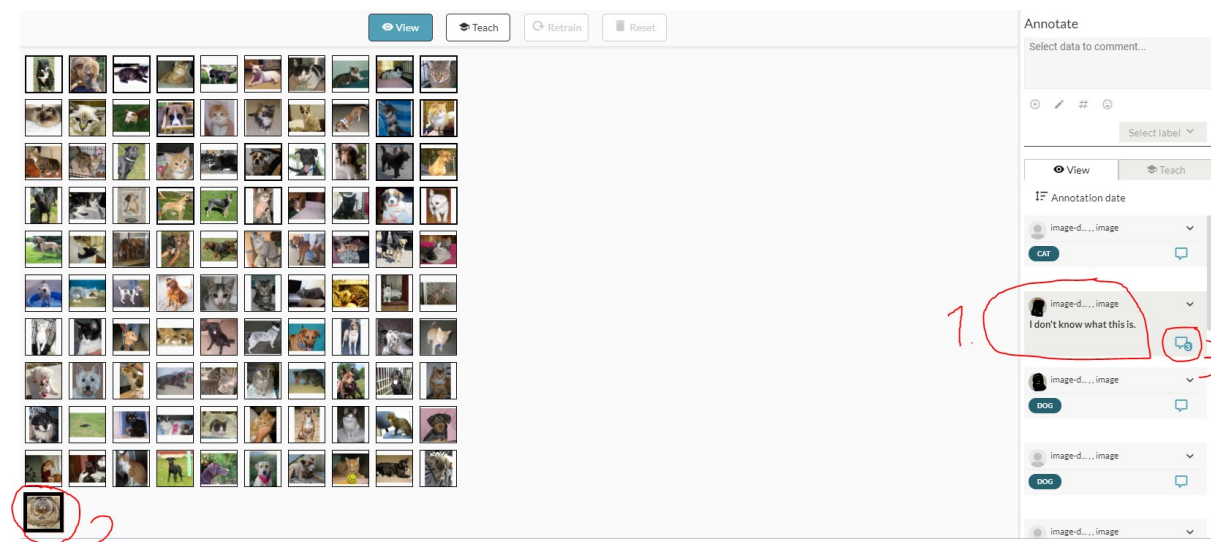
För den fjärde dimensionen av kuben, som består av förbindelser (eng. connections), använder sig silo.ai av ett nätverksdiagram som vi ser i Figur 13 under. Diagrammet är specialiserat på att visa hur olika entiteter eller variabler är kopplade till varandra. Denna typ av diagram är behändigt då man går in på noggrannare detaljer, men ifall det finns en stor mängd data så kan diagrammet snabbt se väldigt otydligt ut då man ser på helheten. (Peter Hellberg, 2017)



Figur 13 Nätverksdiagram (SiloBrain, Infolytika, 2017)

4.3 Annoteringar

I vissa fall kan det vara att resultaten måste diskuteras eller kollas av flera personer. Då är annotationer behändiga, eftersom den problematiska eller oklara platsen/bilden/punkten kan annoteras, alltså märkas ut, så att övriga personer som delar denna applikation kan se annotationen direkt och ta ställning till den ifall de kan. I figur 14 kan vi se ett exempel ur vår bildigenkännings testapplikation, med 3 olika punkter markerade, där denna metod har använts. Vi ser i högra spalten vid siffran ”1.” kommentaren ”I don’t know what this is.”, vilket betyder att personen kommenterat ”jag vet inte vad detta är”. Då vi för markören över denna punkt så lyser annotationen upp bland data, punkt 2 i bilden. Detta är för att visuellt visa åt användarna var själva annotationen finns bland data så de lättare kan se den. I punkt 3 kan vi även se att denna annotation har 3 kommentarer.



Figur 14 Annotation fas 1

I figur 15 nedan kan vi se hur det ser ut då vi trycker på annotationen för att få reda på vad som har skrivits angående denna annotation. Uppe på högra sidan kan vi se personen som har gjort annotationen, information angående punkten som har annoterats (vilken bild och när) och personens kommentar. Under detta kan vi sedan se de övriga personers kommentarer och när de har skrivit dem.



Figur 15 Annotation fas 2

5. Utveckling och test av applikationen

Själva undersökningen av datavisualisering började redan våren 2017 då jag skrev min kandidatavhandling, varefter den har under hösten 2018 utvecklats mer specifikt för bildigenkänningsapplikationen. Under utvecklingen av applikationen har vi uppfyllt de tre karakteristiska dragen som Tullis och Albert (2013) räknar upp att måste finnas då användarupplevelsen ska mätas. Första kravet var att det måste finnas en användare, i detta fall hade vi i alla fall 3 användare. Andra kravet är att användarna måste interagera med en produkt eller system, i detta fall bildigenkänningsapplikationen. Tredje och sista kravet är att användarens upplevelse intresserar och kan iakttas eller mätas, i detta fall iakttagits och mätts. Vi har delvis använt oss av metoden formativ användbarhet som Tullis och Albert (2013) beskriver som en metod för att förbättra användarens upplevelse. I denna metod använder man sig av data och resultat för att förbättra produkten under utvecklingsskedet, för att få fram det ultimata resultatet. I utvecklingen av applikationen har vi mätt mängden musklickningar för att försöka förenkla processen att utföra en körning samt märkning av data. Denna data har vi samlat in under testningsperioden av applikationen före undersökningen samt medan testningsperioden för undersökningen har pågått. Insamling av data har pågått under perioden januari till maj 2019.

Jag börjar kapitlet med att presentera hur applikationen ser ut samt hur den fungerar. Därefter presenterar jag resultaten av applikationens testningar samt mina analyser om dem. I första testningsdelen av applikationen har jag använt mig av 100 stycken bilder på hundar och katter. I andra delen har jag använt mig av 98 stycken bilder på bilar och flyg.

5.1 Bildigenkänningsapplikationen

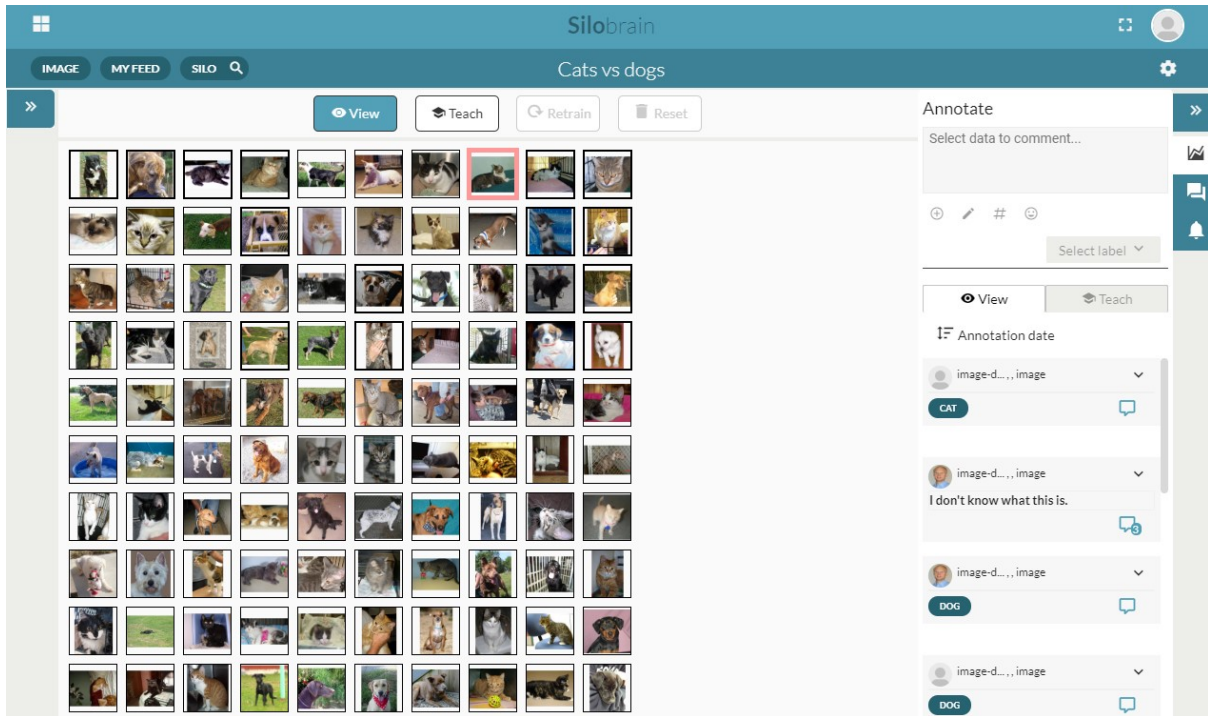
Vi började med att göra en undersökning i datavisualisering, där vi tagit i beaktande hur applikationer är lätt användbara och förstådda för användaren samt hur data kan visuellt presenteras på ett tydligt sätt. Dessa undersökningar gjordes på en allmän nivå för applikationer som silo.ai gör, vilket betyder att vi även tagit dem i beaktande för bildigenkänningsapplikationen. Största delen av datavisualiseringsforskningen gjorde jag redan till min kandidatavhandling, varefter jag har fortsatt studera vidare ämnet för att kunna tillämpa materialet för testapplikationen som är denna bildigenkänningsapplikation. Samtidigt har vi tagit i beaktande övriga faktorer som gör användarupplevelsen bättre, som t.ex.

placering av olika knappar, zoomning och flera liknande användargränssnitt.

Användargränssnitt (UI) metoden som vi använde oss av är scenarier, som Keinonen, Mattelmäki, Soosalu och Säde (1998) beskriver som en bra metod för att visuellt planera funktionen mellan användaren och applikationen. I datavisualiseringens planering gjordes scenarier med hjälp av Microsoft Office Word och SiloBrain applikationen i Java och i bildigenkänningsapplikationen använde jag mig av penna och papper för att visualisera hur olika sidor i applikationen skall se ut och hur knappar borde placeras med mera.

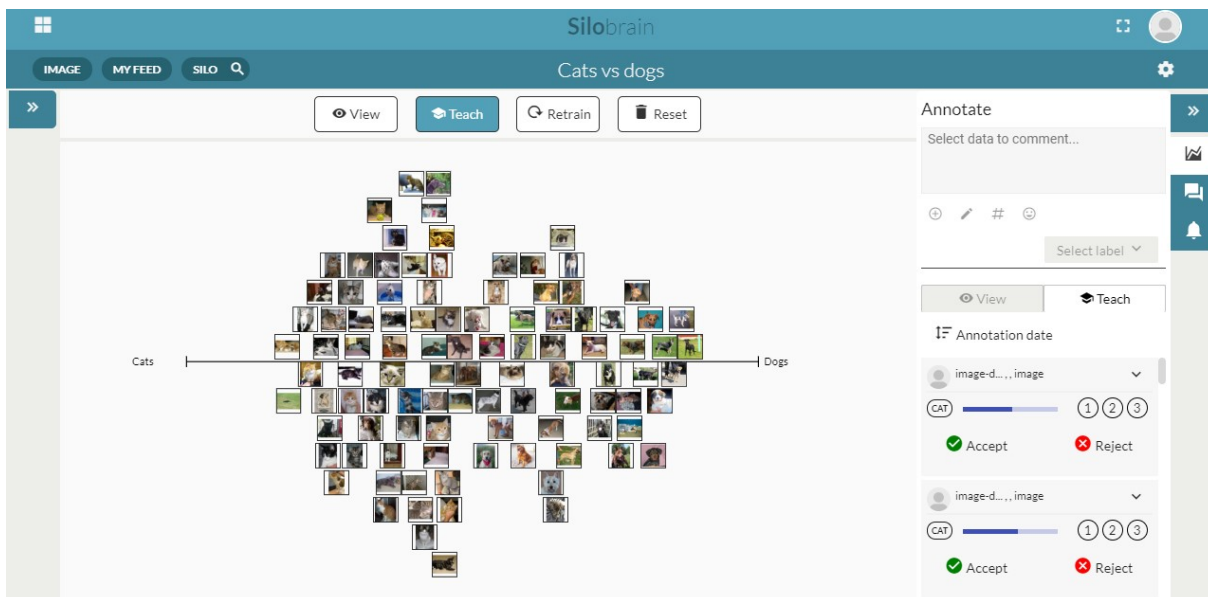
Algoritmen som har använts till applikationen är en klassifikationsalgoritm som påminner om "Nearest Neighbor" algoritmen, som enligt Lantz (2015) beskrivning är en väldigt simpel men användbar metod, speciellt i fall där koncepten är väldigt svåra att definiera men lätta att tolka när man ser dem. Denna metod kan beskrivas med en av de lättaste algoritmerna som används i maskininlärning, den så kallade k-NN algoritmen. Kort och lätt beskrivet fungerar algoritmen enligt följande: bokstaven "k" står för en variabel och algoritmen går ut på att den mäter avståndet från variabeln "k" till de övriga variablerna. När avstånden har mätts kan den gruppera variablerna som ligger närmast varandra ihop. (Brett Lantz, 2015, s.66-69)

I figur 16 kan vi se hur applikationens "framsida" ser ut. Som vi kan se uppe i mitten på bilden så heter denna flik "View", vilket innebär att användaren kan på denna flik se data presenteras. Ifall användaren vill så kan hen gå igenom bild för bild manuellt och kolla vad som visas i bilden. Det är även i denna flik som användaren märker data genom att trycka på en bild och välja mellan alternativen katt eller hund. Varje enskild märkning registreras och kan ses i högra spalten under rubriken "Annotation date". Här kan användaren även korrigera eller ta bort sina tidigare märkningar.



Figur 16 Bildigenkänning 1 (Dataset view)

Nedan i figur 17 ser vi den andra vyn på applikationen i fliken ”Teach”, där resultatet av körningen ses. I mitten av skärmen ser vi hur bilderna har delats upp och användaren kan här zooma in/ut och flytta på diagrammet hur hen vill. Det går även att trycka fram en enskild bild från diagrammet vilket ger information om bilden som t.ex. med vilken procentuell säkerhet bilden har resulterat som katt/hund.



Figur 17 Bildigenkänning 2 (Teaching view)

Då vi jämför figur 16 och figur 17 så kan vi se att knapparna "Retrain" och "Reset" inte är aktiva då fliken "View" är vald. Dessa knappar är endast aktiva i "Teach" vyn, eftersom man med "Reset" kan nollställa sin körning ifall man vill börja från början, eller så kan man välja "Retrain", som direkt översatt betyder "träna på nytt", ifall man gjort någon ändring och vill se hur det påverkar resultatet. Denna funktion är jag inte helt nöjd med personligen, eftersom jag inte anser den vara den mest optimala lösningen. Jag skulle vilja att den gjordes tydligare, men har inte kommit på någon lösning till detta, så därmed har inte ändringar till den gjorts.

Efter att applikationen har utvecklats så att vi varit nöjda med utseendet samt funktionen för egen användning, så har jag gjort tester på hur applikationen fungerar. Jag har använt mig av två olika data set, för att kunna jämföra resultaten från de olika testen. Ett data set består av 50 bilder på hundar och 50 bilder på katter. Det andra består av 49 bilder på bilar och 49 bilder på flyg.

5.2 Analys av hund och katt-modellen

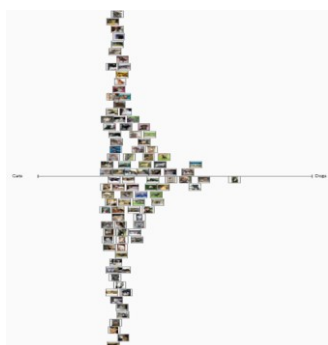
Jag började med att märka en katt och en hund. Detta gjorde jag 2 gånger, för att se ifall det hade någon skillnad vilka bilder jag märkte. På första omgången märkte jag bilderna som kan ses i figur 18. Dessa 2 bilder var de 2 första bilderna i datasetet. Då vi jämför dessa 2 bilder så kan vi konstatera att de är relativt olika. Bakgrunden i bilden är annorlunda och annan färg, profilen på djuren är olika eftersom hundprofilen är framifrån medan kattprofilen är från sidan. Djuren är dock båda svarta, vilket kan påverka resultatet.



Figur 18 Testkörning 1

Resultatet av första testkörningen kan vi se i figur 19. Det var inte något överraskande, alla bilder ligger runt 60% på katt, vilket betyder att applikationen i princip inte kan säga ifall det

är katt eller hund på bilden. Av någon orsak ligger resultaten lite mer på kattsidan, men det är svårt att säga vad det beror på. Bilder med grön bakgrund ligger mer på hundsidan, vilket tyder på att det möjligtvis har haft någon vikt i klassificeringen.



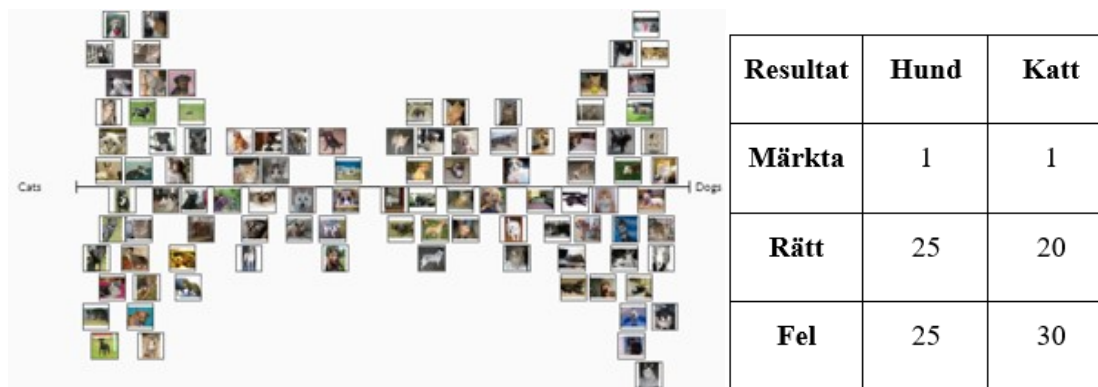
Figur 19 Resultat av test 1

Testkörning 2 försökte jag välja ut varsin bild på katt o hund som skiljer dem åt tydligt och har klara karakteristiska drag för sin art. Bilderna jag valt kan ses i figur 20, hunden är tagen med sidoprofil med en tydlig lång nos, vilket är relativt karaktäriserande för hundar, medan katten har spetsiga öron och en kurad rygg.



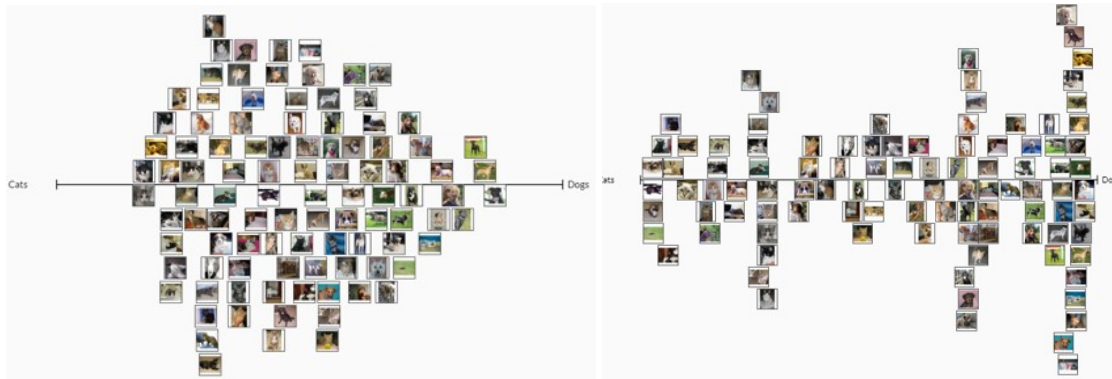
Figur 20 Testkörning 2

Resultatet som syns i figur 21 ger en klar uppdelning på bilderna och med en hög självsäkerhets procent. Då man zoomar in på resultaten och går igenom uppdelningen så ser man dock att bilderna är blandade på båda sidorna, och det finns ungefär lika många katter som är märkta som hundar och vice versa. Det intressanta som kan tolkas från detta är att det visar en klar skillnad på resultaten beroende på vad som har märkts före körningen, även om det inte är ett pålitligt resultat ännu, eftersom 55 bilder blev fel kategoriserade.



Figur 21 Resultat av test 2

Efter detta övergick jag till att märka 5 bilder av vardera art, för att se hur stor skillnad detta har jämfört med tidigare test. Även här har jag provat med olika bilder, test 3 som ses på vänstra sidan av figur 22, består av 5 första bilderna i datasetet av varsin art. På högra sidan av figur 22 ser vi test 4 där jag valt 5 bilder av varsin art med karaktäriserande drag. Vi kan se tydliga skillnader i resultaten, både från tidigare test med 2 märkta bilder samt sinsemellan med 10 märkta bilder. I test 3 är bilderna mer ihop klumpade, vilket berättar att den procentuella säkerheten på resultaten inte är så hög, jämfört med test 4 där bilderna är mer utspridda och bilderna längst till höger och vänster på grafen har väldigt höga procentuella säkerheter enligt applikationen. Då jag zoomar in och kollar noggrannare på resultaten så kan jag konstatera att det fortfarande finns en hel del fel märkta bilder. I test 3 finns det 23 felmärkta bilder och på test 4 finns det 21 felmärkta bilder. Med noggrannare analys av resultaten i test 4 så kan jag konstatera att resultatet är säkrare med katterna även om fler katter blev felmärkta, antagligen eftersom dessa oftast har spetsiga öron och har tydligt klassats enligt det. Uppdelningen på hundarna kan man se en större osäkerhet på, eftersom applikationen troligen inte hittat ett klart tecken på uppdelningen. Ifall djuret är sittandes, men utan spetsiga öron så har den klassats som hund. Också de djur som har en stående sidoprofil har klassats som hundar.

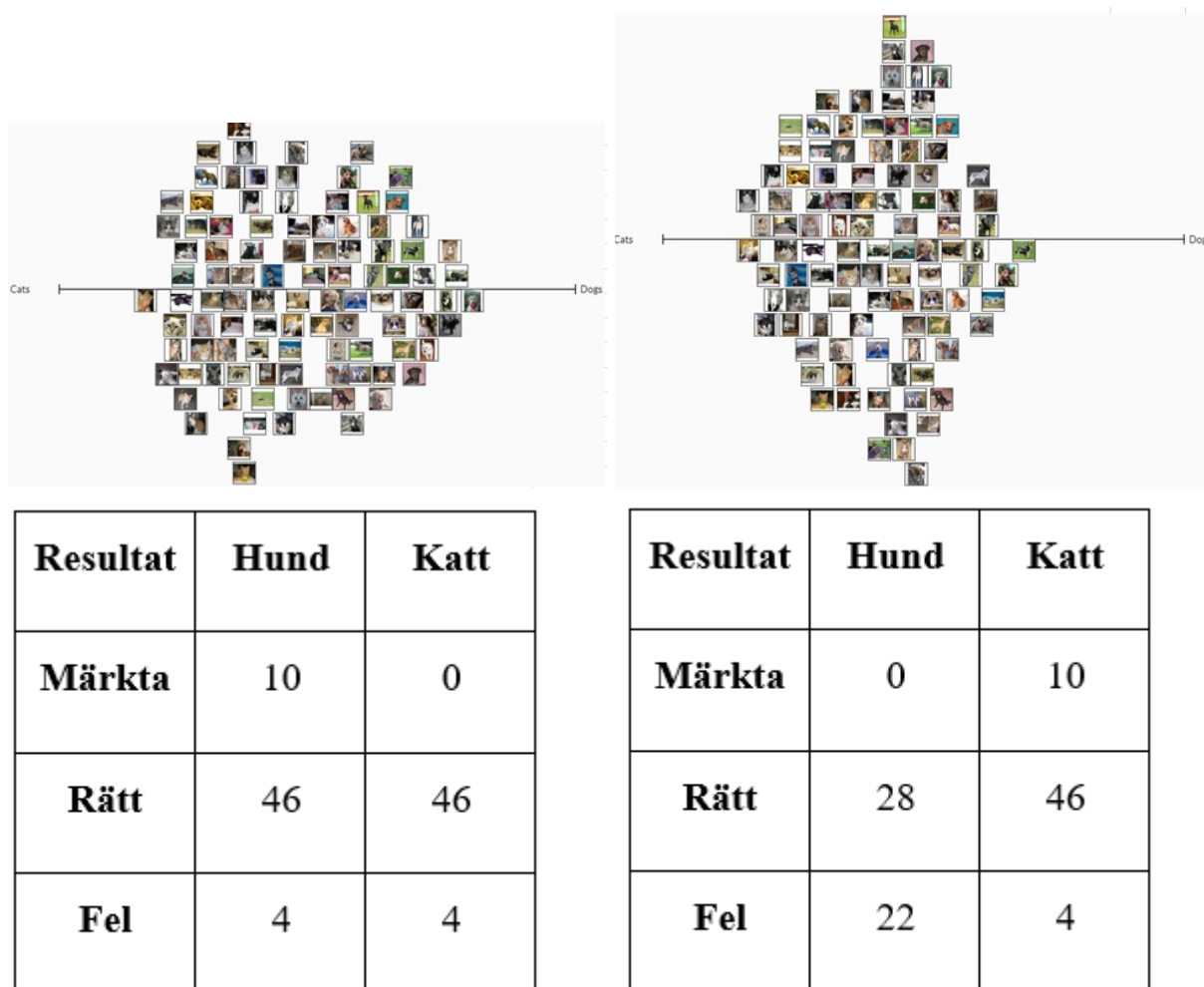


Resultat	Hund	Katt
Märkta	5	5
Rätt	37	40
Fel	13	10

Resultat	Hund	Katt
Märkta	5	5
Rätt	41	38
Fel	9	12

Figur 22 Test 3&4

Efter detta provade jag med att märka först 10 hundar och inga katter, vars resultat vi kan se till vänster i figur 23, och sedan 10 katter och inga hundar, vars resultat vi kan se till höger i figur 23. Skillnaden på resultatet mellan dessa två var väldigt stor. Resultatet av att märka hundarna var relativt bra, endast 8 bilder totalt var fel klassificerade, fastän den procentuella säkerheten inte var så hög. Däremot var resultatet av att märka 10 katter väldigt dåligt. Totalt 26 felmärkta bilder, varav 22 stycken var hundar. Dvs. att endast 28 hundar av 50 blev rätt klassificerade och den högsta procentuella säkerheten för hundar var 71%. Detta tyder på att applikationen har lättare att skilja på katt och hund då den har mer information angående hunden.



Figur 23 Test 5&6

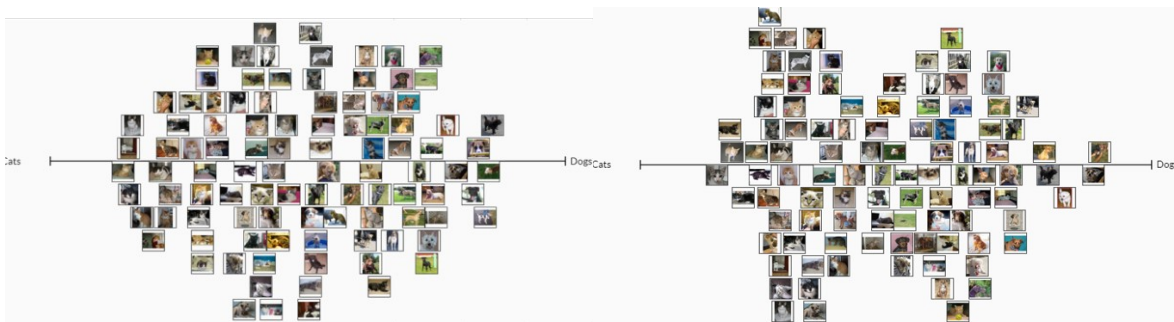
Testandet fortsatte med att jag till följande gjorde 3 olika testkörningar med 20 totala märkningar, varav de två första körningar var med 10 hundar och 10 katter märkta, medan den tredje körningen var med 11 hundar och 9 katter. Som vi kan se i figur 24 och figur 25 så börjar det vara en tydligare spridning på klassificeringen, även om resultaten var ganska växlande. I första körningen där jag försökte märka bilder med klara skillnader på hund och katt, bilden till vänster i figur 24, hade resultatet 15 fel totalt varav 6 var hundar och 9 katter. Den procentuella säkerheten var dock ganska låg på de felklassificerade bilderna och de flesta rörde sig runt 50%. I andra körningen märkte jag på måfå 10 hundar och 10 katter och då blev klassificeringen mer självsäker, som vi kan se i figur 25, men kvalitén på klassificeringen är dålig. Totalt fanns det 21 felklassificerade bilder, 14 hundar och 7 katter, och den procentuella säkerheten på vissa bilder var över 70%. Resultatet på klassificeringen ändrade igen en hel del i tredje körningen, till höger i figur 24, där jag märkt 11 hundar och 9 katter. Nu fanns det 16

fel klassificeringar varav 9 var katter och 7 hundar. Felen rör sig runt 50% och majoriteten av bilderna är relativt otydliga även för människoögat.

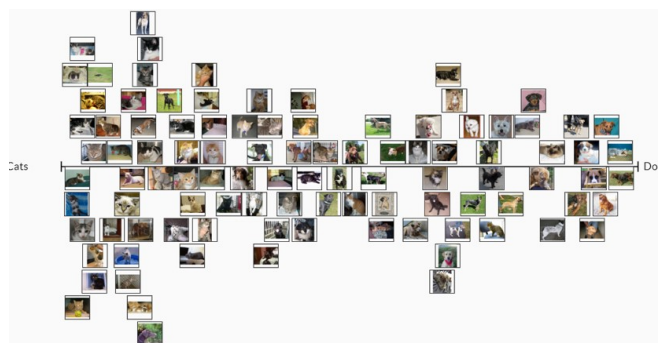
Det är svårt att säga vad skillnaden på resultaten beror på, men utgående från resultaten av de tidigare körda testen så verkar det fortfarande som att resultatet blir bättre då vi har fler märkta hundar. Också skillnaden på att märka bilder med klara skillnader tyder på ett bättre resultat hittills i testningen.

Resultat	Hund	Katt
Märkta	10	10
Rätt	44	41
Fel	6	9

Resultat	Hund	Katt
Märkta	11	9
Rätt	43	41
Fel	7	9



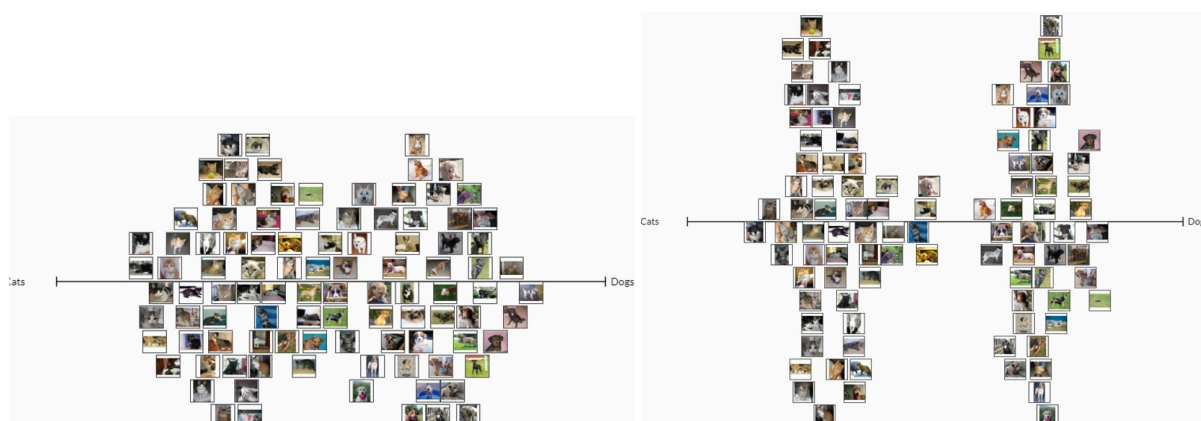
Figur 24 Test 7&9



Resultat	Hund	Katt
Märkta	10	10
Rätt	36	43
Fel	14	7

Figur 25 Test 8

Efter detta gick jag över till att märka 20 hundar och 20 katter och fortsatte att öka på mängden märkt data vartefter jag kört och kollat på resultatet. Ända upp till 100 märkta bilder totalt var resultaten relativt lika då man tog i beaktande mängden felklassificerade bilder, medan uppdelningen blev tydligare visuellt. I figur 26 ser vi till vänster resultatet där jag märkt 20 hundar och 20 katter medan på högra sidan har vi resultatet av 50 hundar och 50 katter märkta. En sammanfattning av resultaten är att mängden felklassificerade bilder varierade mellan 1 och 4, men som vi kan se på resultaten i figur 26 så har uppdelningen av de två olika kategorierna blivit betydligt tydligare. Nu ser man en klar skillnad på vilka som anses vara hundar och vilka katter, och den procentuella säkerheten har också stigit en del. Mängden bilder runt 50% säkerhet har minskat betydligt medan den högsta säkerheten hålls relativt samma runt 90%.

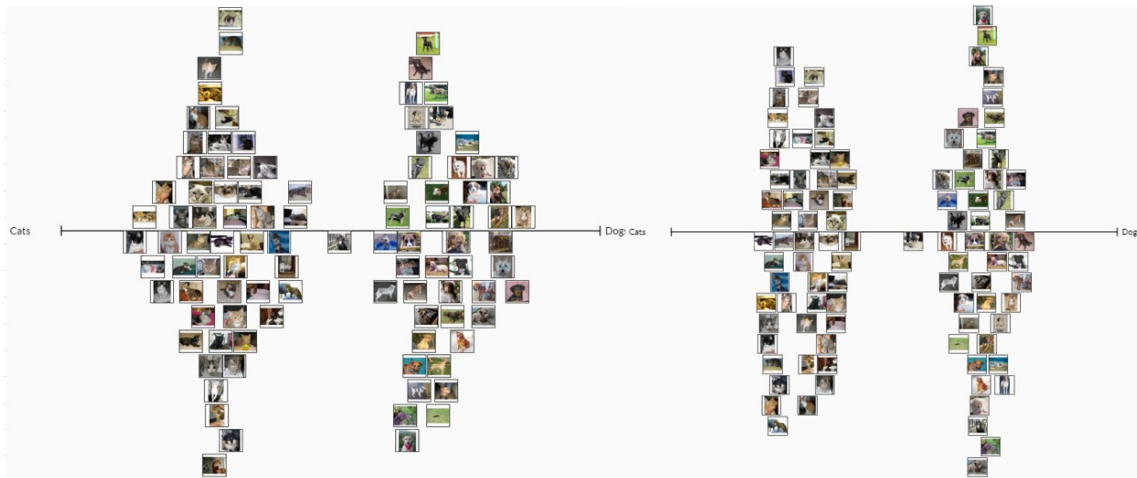


Resultat	Hund	Katt	Resultat	Hund	Katt
Märkta	20	20	Märkta	50	50
Rätt	49	49	Rätt	49	50
Fel	1	1	Fel	1	0

Figur 26 Test med 40+ märkningar

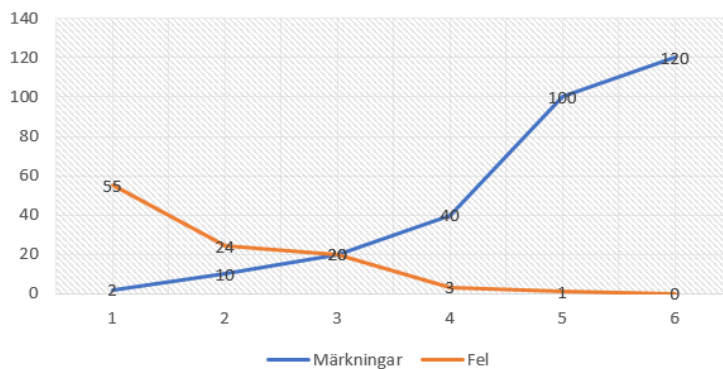
Efter detta fortsatte jag öka på mängden märkt data och gjorde flera omkörningar. Resultatet ändrade inte mera så mycket, oberoende av mängden märkt data och gjorda omkörningar. Resultaten hade inga felklassificerade bilder och uppdelningen var tydlig som kan ses i figur

27, där bilden till höger är med 120 märkta bilder och en omkörning medan bilden till höger har 220 märkta bilder och 4 omkörningar.



Figur 27 Test med 120+ märkningar

Sammanfattningen av statistiken kan vi se i diagrammet i figur 28, där vi ser 2 olika linjer, där den blåa linjen visualiserar antalet märkningar och den orangea linjen visualiserar antalet fel. Som vi kan se i diagrammet, och som konstaterades tidigare i kapitlet, så får vi ett relativt pålitligt resultat redan vid 40 märkningar med tanke på att max 4% av 100 bilder blev felklassificerade då. Dock blev inte uppdelningen av klasserna tydlig förrän 100 märkningar, vilket vi kan se till höger i figur 26, där det har blivit ett tydligt mellanrum mellan klasserna och endast en missklassificerad bild.

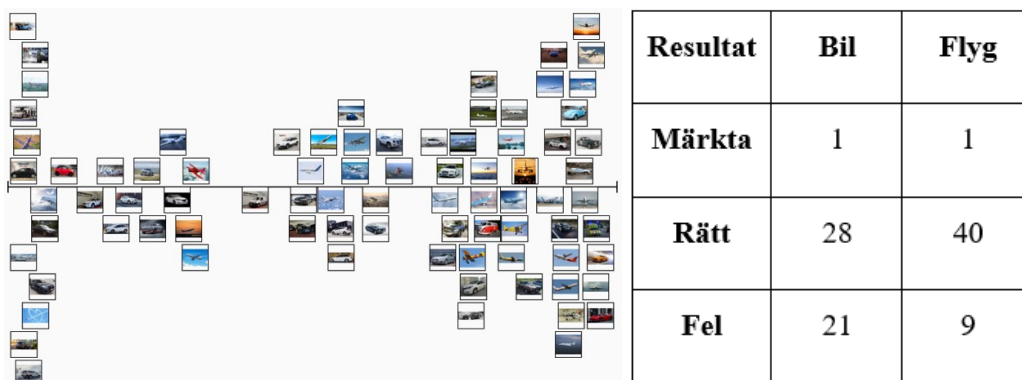


Figur 28 Resultat diagram

5.3 Analys av bil och flyg-modellen

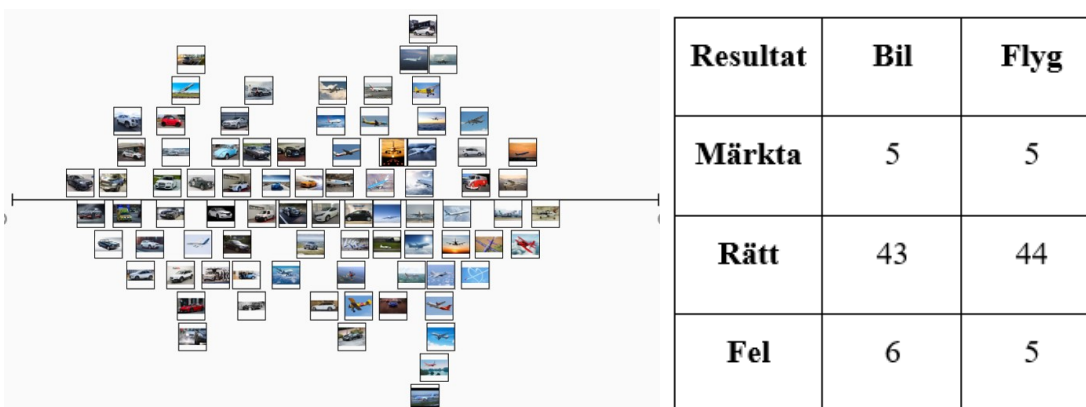
Jag började testet likadant som med hundar och katter, det vill säga med att märka en bil och ett flyg. För att jämföra ifall resultaten skiljer sig på något sätt, så försökte jag med olika bilder, för att se ifall algoritmen tolkar dem på olika sätt, men det visade sig att resultatet var

så gott som samma oberoende av vilka bilder som märktes. Resultatet av detta kan ses i figur 29 nedanför. Vänstra sidan av diagrammet är bilar och högra sidan flyg. Som vi ser så verkar applikationen relativt säker på klassificeringen, men som med hund och katt versionen så har även denna en hel del fel. Totalt fanns det 30 felklassificerade bilder varav 9 var flyg och 21 bilar. Detta syftar på att algoritmen har lättare att skilja på hur ett flyg ser ut.



Figur 29 Test med 1 bil och 1 flyg

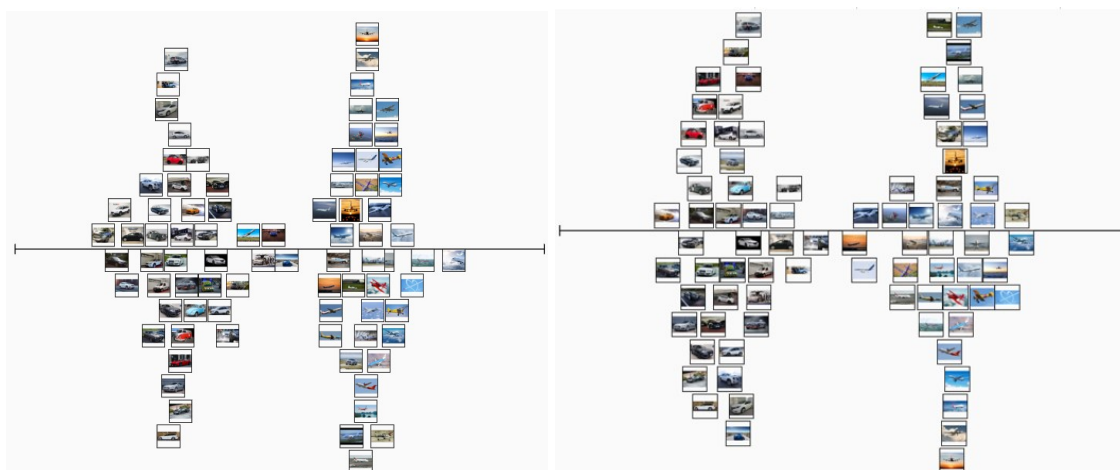
Efter detta gick jag över till att märka 5 bilar och 5 flyg och resultatet av detta kan vi se i figur 30 nedan. Säkerheten av klassificeringen är inte lika hög mera som vi ser, men däremot har resultaten förbättras betydligt jämfört med endast varsin märkning. Nu fanns det endast 11 felklassificeringar varav 5 var flyg och 6 var bilar. Även här skiljs inte resultaten mycket åt beroende på vilka bilder som märktes, däremot skiljs resultaten mycket jämfört med hund och katt versionen, eftersom där var det en stor skillnad på beroende på hurdana bilder som märktes.



Figur 30 Test med totalt 10 märkningar

Nedan i figur 31 ser vi resultaten av 2 följande testkörningar, där bilden till höger är resultatet av 20 märkta bilder och bilden till höger med 40 märkta bilder. Som vi ser så är skillnaden till

10 märkta bilder (figur 30) väldigt stor, eftersom vi nu har en klar skillnad på klassificeringarna, medan resultatet mellan 20 och 40 märkta bilder inte är så stor. Resultaten i figur 31 har en klar uppdelning mellan bilar och flyg och resultaten är även goda då jag går igenom bilderna för att se hur många felklassificerade bilder det finns. Med totalt 20 märkta bilder, 10 bilar och 10 flyg, fanns det 4 felklassificerade bilder varav 3 var bilar och endast 1 var flyg. Då jag märkte 10 bilder till av vardera klass så kan vi se att klassificeringen har blivit lite klarare, eftersom utrymmet mellan bilderna har blivit större vilket betyder att den procentuella säkerheten i klassificeringen ökat. Även antalet felklassificerade bilder minskade från 4 bilder till endast 1 bild, och det var en bil. Detta stöder ännu teorin om att algoritmen har en klarare uppfattning på hur ett flyg ser ut, eftersom alla resultat hittills har varit starkare för flygen än bilar.



Resultat	Bil	Flyg	Resultat	Bil	Flyg
Märkta	10	10	Märkta	20	20
Rätt	46	48	Rätt	48	49
Fel	3	1	Fel	1	0

Figur 31 Test med 20 och 40 märkningar

Efter 40 märkningar fortsätter jag med att öka igen med 10 märkningar av vardera klass och gör en körning, tills jag uppnår 80 märkningar. I figur 33 ser vi resultatet av totalt 80 märkningar som är en klar förbättring från 40 märkningar som vi kan se i figur 31.

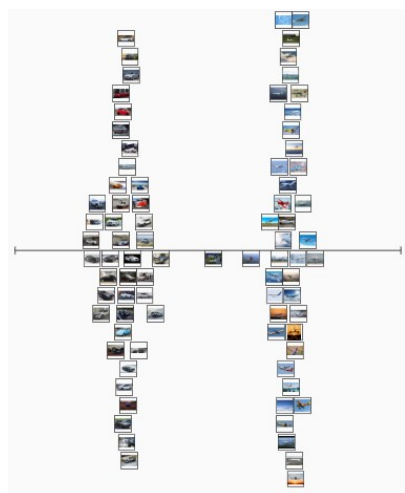
Uppdelningen av klassificeringarna har blivit ännu tydligare, men den högsta säkerhets procenten ökar inte. Vi har även fortfarande ett fel, både med 60 och 80 märkningar, och det är i båda fallen samma bil som med 52% säkerhet blir

klassificerad som flyg. Bilen i fråga kan ses i figur 32, vilket enligt människoögat kan självklart klassificeras som bil, men av någon orsak har algoritmen problem med denna bild.

Visserligen finns det en chans att det är fråga om en ”bug” i applikationen, men det är svårt att säga eftersom bilden inte är klassificerad alltid samma.



Figur 32 Felklassificerad bil



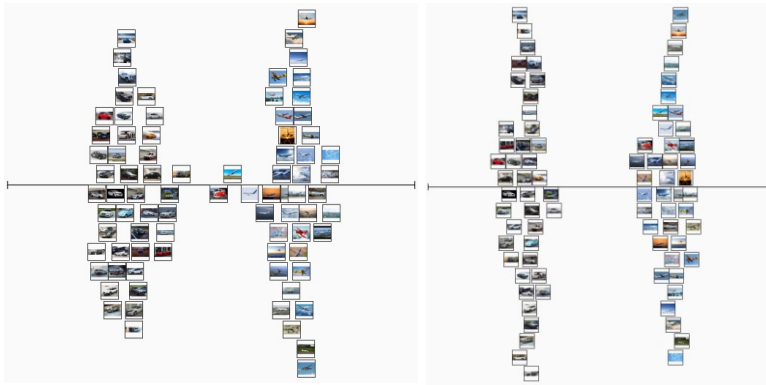
Resultat	Bil	Flyg
Märkta	40	40
Rätt	48	49
Fel	1	0

Figur 33 Test med 80 märkningar

Jag fortsätter med att öka mängden märkningar som är slumpmässigt valda, det vill säga att jag inte kollar på någon speciell bild med en speciell profil. Resultatet fortsätter likadant som tidigare, 1 bil blir felklassificerad, men denna gång är det en annan bil. Nu är problemet en volkswagen som vi kan se i figur 34. Förutom den så förbättras skillnaden mellan klassificeringarna hela tiden, som vi kan se i figur 35.



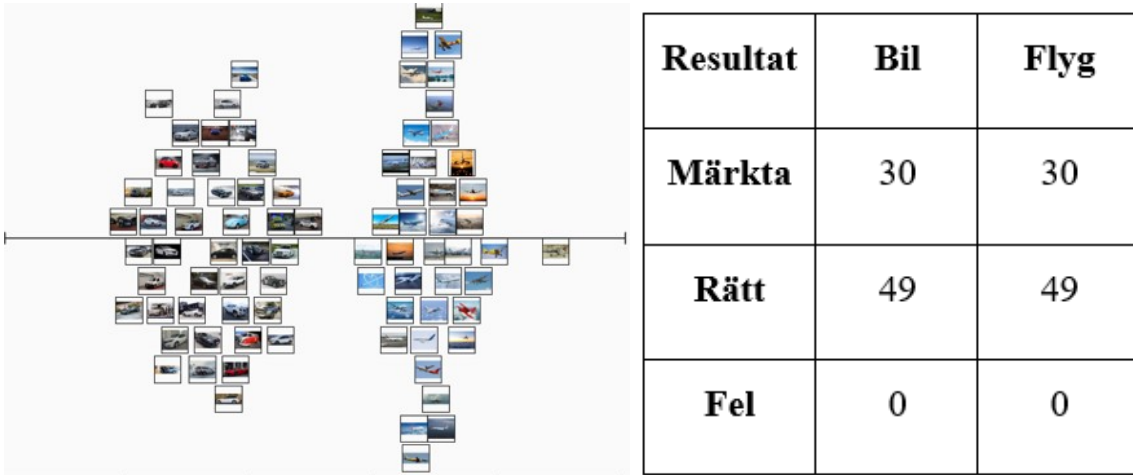
Figur 34 Volkswagen



Resultat	Bil	Flyg	Resultat	Bil	Flyg
Märkta	50	50	Märkta	60	60
Rätt	48	49	Rätt	48	49
Fel	1	0	Fel	1	0

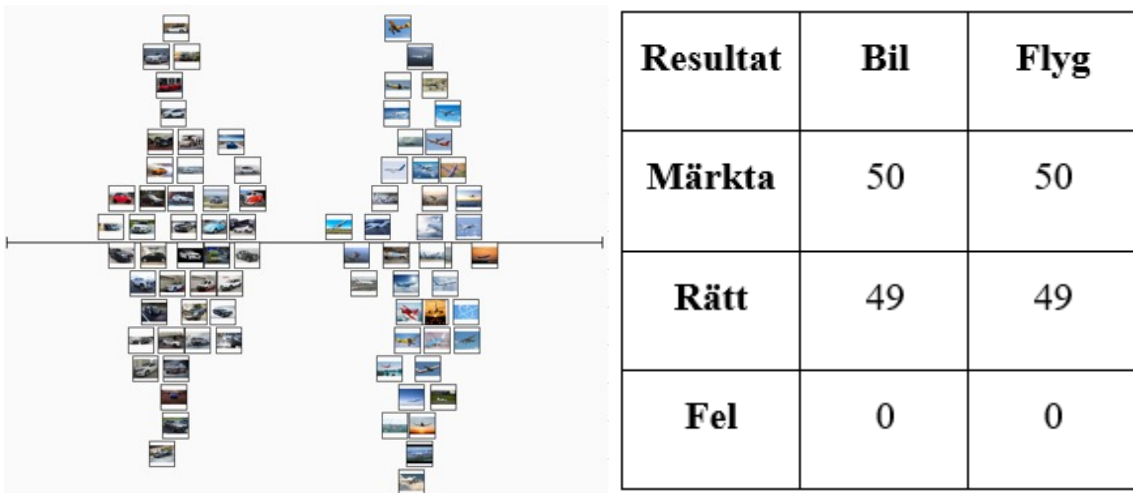
Figur 35 Test med 100 och 120 märkningar

Efter detta börjar jag märka endast bilder som påminner om bilen som konstant är felmärkt, vilket resulterade i att applikationen började ge felfria resultat, det vill säga alla bilder var rätt klassificerade. För att kolla ifall det var de märkta bilderna som var problemet så började jag minska på mängden märkta bilder, och koncentrerade på att märka bilder som liknade de tidigare ”problematiska” bilderna som klassificerats fel. Det visade sig att detta hade en stor skillnad på resultatet och jag kunde gå ända ner till 60 märkningar utan att en ända bild felklassificerades. I figur 36 kan vi se hur resultatet med 30 märkningar av vardera klass såg ut då jag koncentrerat mig på endast bilar som påminde om de tidigare missklassificerade bilarna. Då jag gick under 60 märkningar så började resultaten påminna om de tidigare resultaten, så före denna mängd verkar inte märkningen ha en så stor roll. Resultaten över 60 märkningar påminde om de tidigare resultaten, men istället för att ha en felklassificerad bild så var alla rättklassificerade.



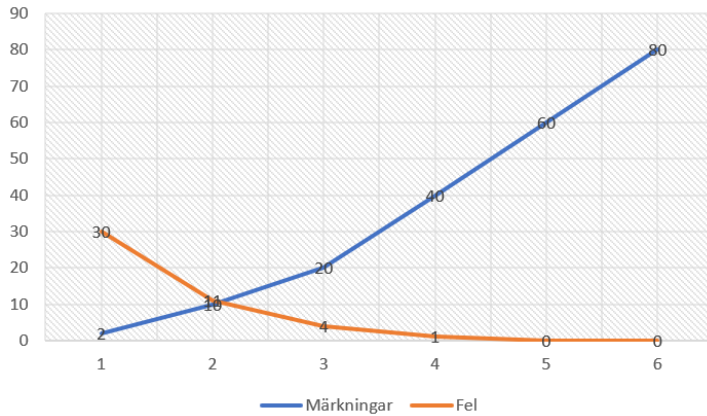
Figur 36 Koncentrerad märkning, 60 bilder

Resultaten över 60 märkningar påminde om de tidigare resultaten, men istället för att ha en felklassificerad bild så var alla rättklassificerade. I figur 37 ser vi resultatet av 100 märkningar, 50 bilar och 50 flyg, där alla bilder klassificerats rätt och med en god säkerhet.



Figur 37 Koncentrerad märkning, 100 bilder

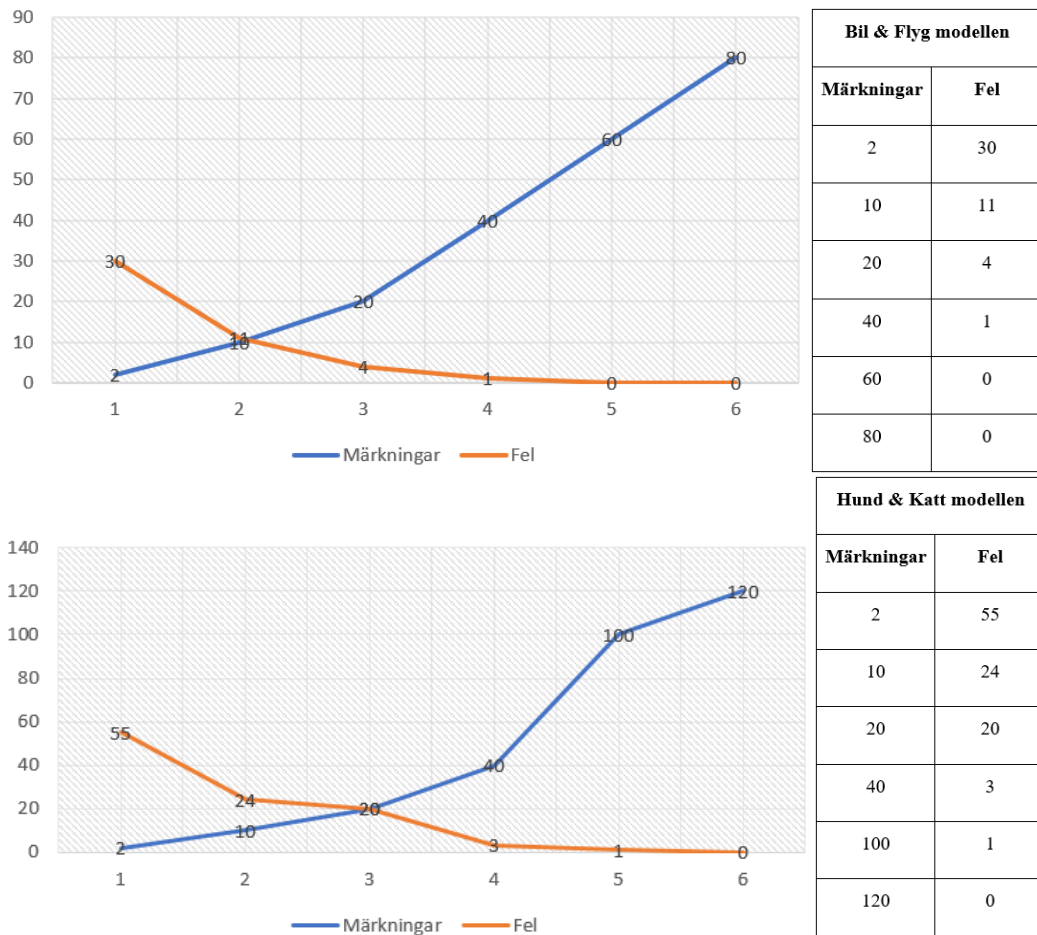
Sammanfattningen av resultaten då jag gjorde en koncentrerad märkning, vilket i detta fall kan även kallas en optimerad märkning, kan ses i diagrammet i figur 38 nedanför. Vi kan konstatera att resultatet är betydligt bättre än i hund och katt versionen, eftersom det endast krävdes totalt 60 märkningar för att algoritmen skall kunna klassificera bilderna felfritt.



Figur 38 Resultat för Bil/Flyg modellen

5.4 Sammanfattning av resultaten

I figur 39 kan vi se sammanfattningen av resultaten ur båda modellerna. Det märks klart att algoritmen hade svårare att klassificera hund och katt bilderna jämfört med bil och flyg modellen, eftersom det endast krävdes 60 märkningar för algoritmen att klara av klassificeringen utan fel för bil och flyg modellen, medan det tog hund och katt modellen dubbelt flera märkningar. Det som jag kan konstatera är att det absolut har en skillnad i resultatet beroende på bilderna som märkts. Detta kom tydligt fram i bil och flyg modellen där algoritmen hade svårt att klassificera en bil, men efter att jag började märka bilar som hade samma form som den problematiska bilen så klarade algoritmen av att klassificera rätt vid ett tidigt skede.



Figur 39 Sammanfattning av resultaten

6. Diskussion och slutsatser

Denna avhandling fick sin början redan i min kandidatavhandling våren 2017, då jag skrev om datavisualisering för silo.ai, dåvarande Infolytika. Då var min fokus på att hitta optimala lösningar för visuella aspekter, som kunde användas i dåvarande samt framtida applikationer. I denna avhandling har huvudfokus varit på att undersöka hur stor inverkan användaren har i en bildigenkänningsapplikation som är byggd på en klassificeringsalgoritm, men för att möjliggöra detta så har vi tillsammans med 2 personer från silo.ai planerat på designen av applikationen för att göra den användarvänlig först.

Eftersom det ännu finns så lite litteratur samt information publicerat angående bildigenkänningsapplikationer, så ville jag i min empiriska del vara med och utveckla en applikation tillsammans med ett professionellt företag, i detta fall silo.ai, för att själv kunna påverka testen och prova olika synvinklar samt analysera resultaten. Jag är tacksam för hjälpen jag fått från silo.ai och för tiden som de offrat för att utveckla applikationen. Det gläder mig att jag är nöjd med resultaten jag kommit fram till i denna avhandling, och hoppeligen motiverar den till fortsatta undersökningar inom ämnet.

Applikationen som användes under den empiriska delen av avhandlingen anser jag vara relativt användarvänlig redan. För tillfället kräver den en viss del handledning för att användaren skall kunna ta full nytta av den. Det som kan kräva en del hjälp för användaren nu är att förstå hur själva träningsdelen av applikationen fungerar och vilken data som tas i beaktande för körningen. ”Reset” knappen t.ex. nollställer inte allting, utan den går endast tillbaka ett steg. För att göra sig av med data som märkts måste man manuellt ta bort 1 bild i taget. De huvudsakliga funktionerna är dock lättanvändbara som t.ex. märkning av data och analyserande av resultatet, eftersom de kommer tydligt fram åt användaren. Eftersom märkning av data och resultatanalys är de mest kritiska användarfunktionerna i denna applikation, så anser jag att vi lyckats bra med skapandet av användargränssnittet och användarupplevelsen.

Efter att ha gjort testen med bildigenkänningsapplikationen kan jag konstatera att jag är nöjd med resultaten jag fått, eftersom resultaten visar en jämn och tydlig utveckling i resultaten, vilket jag anser vara ett tecken på att algoritmen i applikationen fungerar. Jeremy Howard (2014) går igenom i sin presentation om hur datorer lär sig bland annat hur en

bildigenkänningsapplikation fungerar. Orsaken till varför jag inte fick en klar uppdelning efter bara några märkta bilder är för att algoritmen koncentrerar sig inte på djuret i bilden, så som människan gör, utan algoritmen tar i beaktande varje pixel i bilden. Det vill säga att bakgrunden och allting runtomkring i bilden påverkar resultatet som ges. Howard (2014) påpekar i sin presentation att på grund av att algoritmen tar allting i beaktande, så leder det till att datorn i ett skede går förbi människans kunskap att känna igen föremålet på bilden eftersom bilden inte behöver vara tydlig för algoritmen. Med hjälp av att märka mer och mer data lär sig algoritmen hur den skall skilja på klassificeringen, vilket vi även kan se i denna forskning. Även bilder där det är svårt för människoögat att avgöra ifall det är hund eller katt, som vi kan se i figur 40. Denna bild kan vara svår för människoögat att avgöra vilket djur det är fråga om, men applikationen fick den rätt.



Figur 40 Otydlig katt

Det som överraskade mig var den lilla mängden data som måste märkas, tills algoritmen klarade av att självständigt klassificera bilderna. I hund och katt applikationen krävdes det endast 120 bilder för att applikationen inte skulle göra fel, medan i bil och flyg versionen krävdes det endast 60 märkningar. Dessa resultat tyder på det att algoritmen har lättare att klassificera bilder som även för människoögat har stora skillnader sinsemellan. En hund och katt är definitivt svårare att skilja på jämfört med skillnaden på en bil och ett flyg.

Jag är även nöjd med det att på basis av denna undersökning så kan jag konstatera att användaren har en klar påverkan i resultatet som ges av en bildigenkänningsapplikation, eftersom jag kunde i min bil och flyg version klart se skillnad på resultaten beroende på vilka bilder som märktes i upplärningen av algoritmen.

Ifall jag haft mera tid hade jag varit intresserad av att öka mängden bilder till 1000 stycken, för att se ifall det skulle ändra på resultatet. Tyvärr var det inte möjligt till denna avhandling, men det är en rekommendation till nästa undersökning.

Källor

Assam, Evergreen (2013) *Data Visualizaion, Part 1. New Directions for Evaluation. Number 139, Fall 2013*. Jossey-Bass and the American Evaluation Association.

Bo Eneroth (2011) *Grunbok I kvalitativ metod: "Hur mäter man "vackert"*, Natur & Kultur. Hämtad den 3.5.2019 från <https://aboakademi.elib.se/Books/Details/1000482>

Bryan R. Burnham (2015) *Fundamental Statistics for the Behavioral Sciences (v. 2.1)*, University of Scranton, US.

Burr Settles (2009) *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Dzemyda, G., Kurasova, O. & Medvedev, V. (2007) *Emerging Artificial Intelligence Applications in Computer Engineering: Dimension Reduction and Data Visualization Using Neural Networks*. IOS Press, Amsterdam.

Fayyad, U., Grinstein, G. & Wierse, A. (2002) *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers.

Few, Stephen (2008). *Practical Rules for Using Color in Charts. Visual Business Intelligence Newsletter, February 2008, pp. 2-7*. Perceptual Edge. Hämtat den 9.3.2018 http://www.perceptualedge.com/articles/visual_business_intelligence/rules_for_using_color.pdf

Hellberg, Peter (2017). *Användning av färger och grafer i datavisualisering*. Kandidatavhandling, Åbo Akademi.

Jeremy Howard (2014) *The wonderful and terrifying implications of computers that can learn*. Tedx Talks, Youtube. Hämtad den 2.5.2019 från <https://www.youtube.com/watch?v=xx310zM3tLs&feature=youtu.be&t=13m4s>

Keinonen, T., Mattelmäki, T., Soosalu, M. & Säde, S. (1998) *User interface design tools*. University of Art and Design Helsinki UIAH, Helsinki.

Konyushkova, K., Sznitman, R. & Pascal, F. (2017) *Learning Active Learning from Data*. Neural Information Processing Systems (NIPS), Long Beach. Hämtat den 22.8.2018 <http://papers.nips.cc/paper/7010-learning-active-learning-from-data.pdf>

Kotsiantis, S.B. (2007) *Emerging Artificial Intelligence Applications in Computer Engineering: Supervised Machine Learning: A Review of Classification Techniques*. IOS Press, Amsterdam.

Lantz, Brett (2015) *Machine Learning with R, Second Edition*. Packt Publishing Ltd.

Lewis, D. & Gale, W. (1994) *A sequential algorithm for training text classifiers*. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 3-12. Springer-Verlag.

Lin, Fortuna, Kulkarni, Stone, Heer (2013) *Selecting Semantically-Resonant Colors for Data Visualization*. *Computer Graphics Forum*, vol. 32, no. 3pt4, pp. 401-410. Blackwell Publishing Ltd. Hämtat den 16.5.2017 <http://vis.stanford.edu/files/2013SemanticColor-EuroVis.pdf>

Manning, Q. *Why Your App's UX is More Important than You Think*, Code Magazine, Hämtad den 10.5.2018 <http://www.codemag.com/Article/1401041>

Moreno, Helga (2014) *The Gap Between UI and UX Design – Know the Difference* Hämtad den 5.5.2018 <http://snip.ly/mjj7s#http://www.onextrapixel.com/2014/04/24/the-gap-between-ui-and-ux-design-know-the-difference/>

Nieminen, Tuula (2009) *Visuaalinen markkinointi*. WSOYpro Oy

Rokach, L. & Maimon, O. (2008) *Data Mining with Decision Trees: Theory and Applications*, World Scientific Publishing Co Pte Ltd

Saari, Matti (2011) *Silmätautioppi*. Kandidaattikustannus Oy, Helsinki.

Sacks, O. (1995) *En antropolog på Mars*. Bromberg, Stockholm.

Sarlin, Peter (2013) *Mapping Financial Stability*. Åbo Akademi University, Turku Centre for Computer Science.

Seth Mottaghinejad (2018) *Azure Machine Learning Demo*, Youtube, Hämtad 29.4.2019 från <https://www.youtube.com/watch?v=FKbbuYuojf4>

Shalev-Shwartz, Shai & Ben-David, Shai (2014) *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge.

Sibson, R. (1973) *Slink: An optimally efficient algorithm for the single-link cluster method*, The Computer Journal, Volume 16 Number 1, Hämtad den 27.4.2018 https://grid.cs.gsu.edu/~wkim/index_files/papers/sibson.pdf

Stephen M. Kosslyn (2006) *Graph Design for the Eye and Mind*. Oxford University Press, Inc. New York, New York.

Stone, Maureen (2006) *Choosing Colors for Data Visualization, January 2006, pp. 6-9*. Hämtat den 9.3.2018 http://www.perceptualedge.com/articles/b-eye/choosing_colors.pdf

Tong, Simon & Koller, Daphne (2001) *Support Vector Machine Active Learning with Applications to Text Classification*, *Journal of Machine Learning Research*, pp. 45-66. Hämtat den 21.8.2018 <http://www.jmlr.org/papers/volume2/tong01a/tong01a.pdf>

Tullis, Tom & Albert, Bill (2013) *Measuring the user experience: Collecting, Analyzing, and Presenting Usability Metrics. Second edition*. Elsevier Inc.

Urner, J., Bucher, D., Yang, J. & Jonietz, D. (2018) *Assessing the Influence of Spatio-Temporal Context for Next Place Prediction using Different Machine Learning Approaches*, International Journal of Geo-Information, MDPI AG, Basel, Switzerland. Hämtad den 10.5.2018 <http://www.mdpi.com/2220-9964/7/5/166/htm>

Vural, M. & Gök, M. (2017) *Criminal prediction using Naive Bayes theory*, Neural Computing and Applications, Volume 28, issue 9, pp 2581-2592. Hämtat den 28.4.2018 <https://link-springer-com.ezproxy.vasa.abo.fi/content/pdf/10.1007%2Fs00521-016-2205-z.pdf>

Widerberg, K. (2002) *Kvalitativ forskning i praktiken*. Lund: Studentlitteratur.

Witten, I. & Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier Science & Technology.