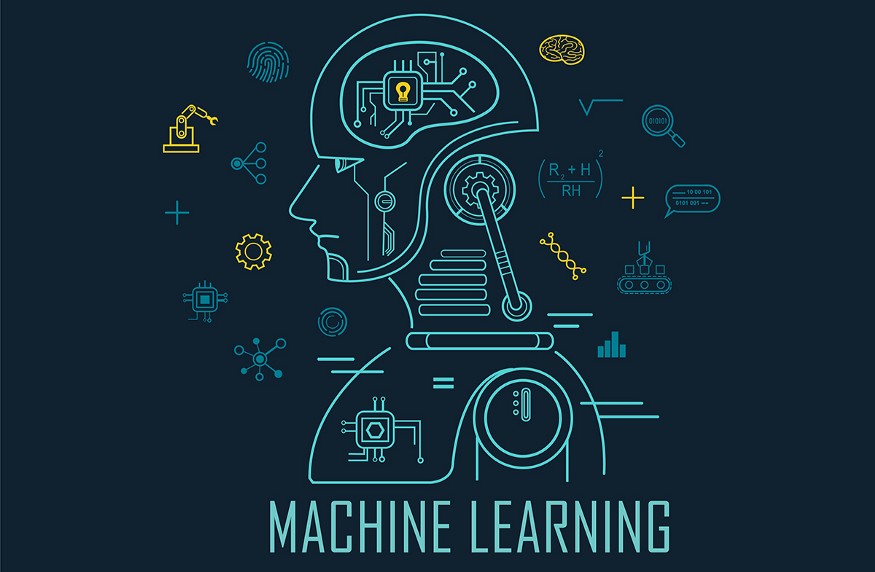
Machine Learning og bioinformatik (matematik)



https://becominghuman.ai/an-introduction-to-machine-learning-33a1b5d3a560

*Materialet er hentet fra Dataekspeditioner.dk (*[*https://dataekspeditioner.dk/matematik-og-machinelearning/*](https://dataekspeditioner.dk/matematik-og-machinelearning/)*) og er udviklet af Mette Machholm og Morten Flø Jørgensen. Det er herefter tilrettet af Kasper Weirum Risgaard*

Indhold

[1. Hvad skal vi? 3](#_Toc126141592)

[1.1. k-Nearest-Neighbors (kNN) 4](#_Toc126141593)

[1.2. Introduktion til Machine Learning 7](#_Toc126141594)

[1.3. Machine Learning i programmet Orange 8](#_Toc126141595)

[1.3.1. Installer programmet Orange 8](#_Toc126141596)

[1.3.2. Import data 11](#_Toc126141597)

[1.3.3. Vælg den kategori du vil forudse 13](#_Toc126141598)

[1.3.4. Vælg ML-model 14](#_Toc126141599)

[1.3.5. Test din ML-model 16](#_Toc126141600)

[1.3.6. Optimer din ML-model 18](#_Toc126141601)

[1.3.7. Test med nye data 20](#_Toc126141602)

[1.3.8. Bias i træningsdata 21](#_Toc126141603)

[2. Forudse øjenfarve ud fra DNA 22](#_Toc126141604)

[2.1. Machine Learning i programmet Orange 22](#_Toc126141606)

[2.2. Baggrundsviden: Hvorfor ser vi øjne som brune eller blå? 23](#_Toc126141607)

[2.3. Forskellen på statistik og Machine Learning 24](#_Toc126141608)

[2.4. Øjenfarve og mutationer 26](#_Toc126141609)

[2.5. Machine Learning – Decision tree i Orange 29](#_Toc126141610)

[2.5.1. Træn din Machine Learning model 29](#_Toc126141611)

[2.5.2. Test din Machine Learning-model 31](#_Toc126141612)

[2.5.3. Evaluering med Confusion matrix 32](#_Toc126141613)

[2.5.4. Hvilken øjenfarve har gerningsmændene? 33](#_Toc126141614)

[2.6. Forstå betydningen af mutationerne 35](#_Toc126141615)

[2.7. Opsplitning af datasæt i trænings- og evalueringsdata 38](#_Toc126141616)

# Hvad skal vi?

I dette modul skal du lave en Machine Learning model, som kan forudse en persons køn. Du lærer hvordan it-systemer i dag kan trænes til at forudse forskellige ting ud fra nogle data.

Du skal

* undersøge Machine Learning modellen kNN sammen med dine klassekammerater (Aktivitet 1)
* se en video om Machine Learning
* installere programmet Orange
* se en video om programmering i Orange
* lave dit første Machine Learning program i Orange (Aktivitet 2)
* læse om bias i træningsdata og lave en øvelse om bias

## k-Nearest-Neighbors (kNN)

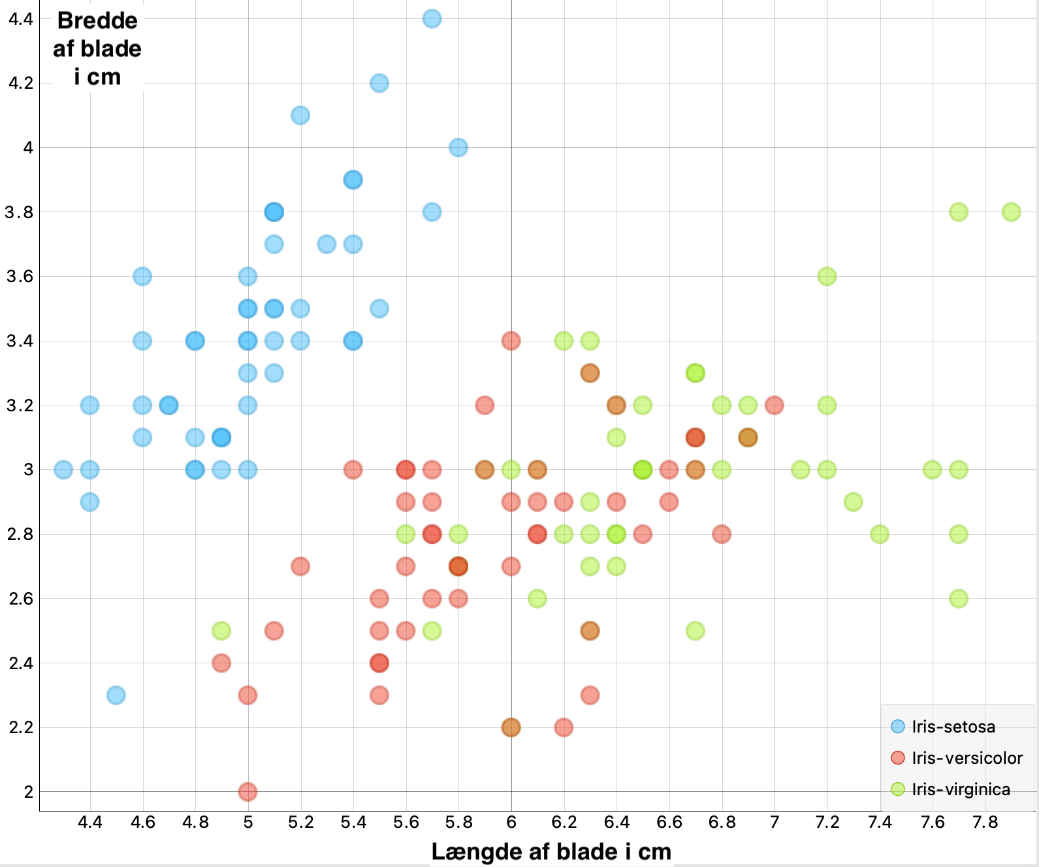
#### Visualisering af k-Nearest-Neighbors-modellen

Nedenfor ses data fra en gymnasieklasse hvor elevernes skostørrelser og højder er plottet i et koordinatsystem. Samtidig er elevernes køn vist ved farverne på datapunkterne. Ved hjælp af modellen k-Nearest-Neighbors (kNN) kan kønnet bestemmes ud fra kønnet på de *k* nærmeste naboer.

Sættes *k=3* for en person med højden 180 cm og skostørrelse 43, vil denne person blive bestemt som mand ud fra modellen, da der er to mænd blandt de nærmeste datapunkter og kun én kvinde, som vist på figuren nedenfor.

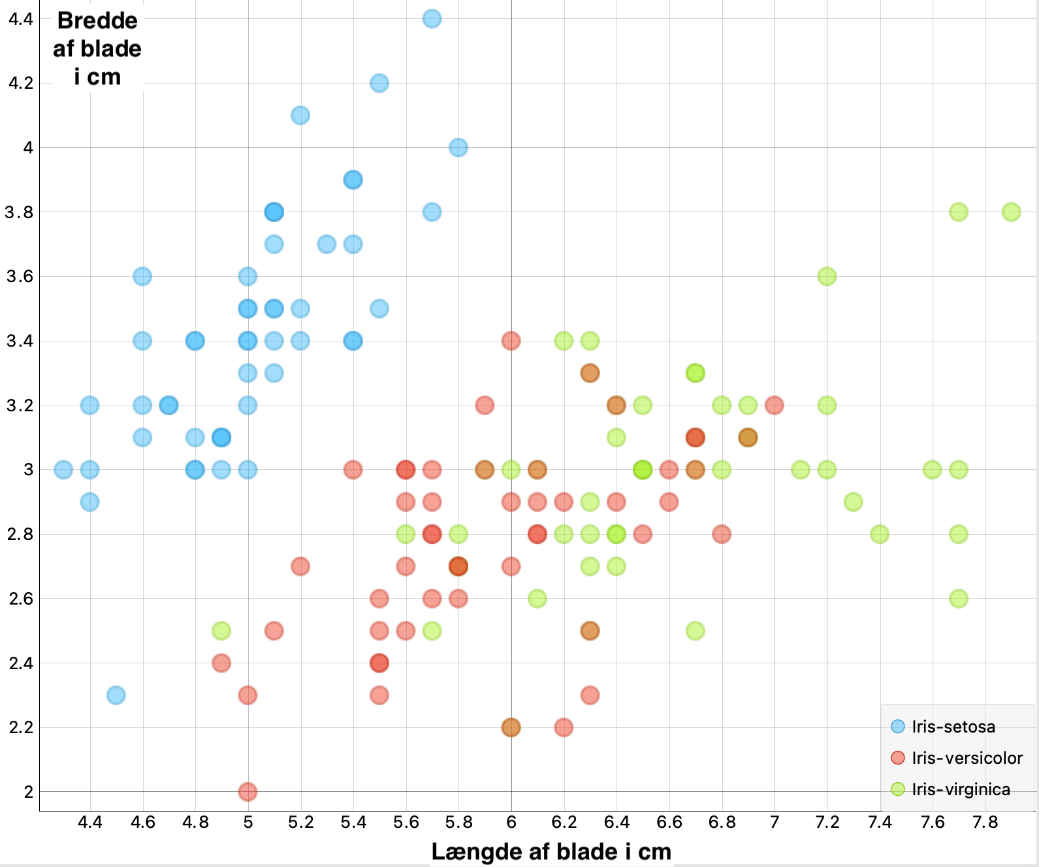
Svar efterfølgende på disse spørgsmål:

* 1. Billedet illustrerer Machine Learning modellen k-Nearest-Neighbors (kNN).  
     Et blad af en iris-blomst har længden 4.8 cm og bredden 3.2 cm. Kan man forudse hvilken Iris-art det er vha. modellen?



*Svarmuligheder:*

* Ja. Det er en Iris-setosa
* Ja. Det er en Iris-versicolor
* Ja. Det er en Iris-virginica
* Nej. Det kan man ikke ud fra data og billedet
  1. Billedet illustrerer Machine Learning modellen k-Nearest-Neighbors (kNN).  
     Et blad af en iris-blomst har længden 6.8 cm og bredden 3.2 cm. Kan man forudse hvilken Iris-art det er vha. modellen?



*Svarmuligheder:*

* Ja. Det er en Iris-setosa
* Ja. Det er en Iris-versicolor
* Ja. Det er en Iris-virginica
* Nej. Det kan man ikke ud fra data og billedet
  1. Hvad står k for i k-Nearest-Neighbors?

*Svarmuligheder:*

* Antallet af datapunkter i koordinatsystemet
* Antallet af punkter man undersøger, når man skal forudse et nyt punkts kategori?
* Antallet af informationer som bruges i modellen. På billederne i de forrige spørgsmål svarer til det k=2

## Introduktion til Machine Learning

Forstå lighederne mellem lineær regression og en af Netflix's algoritmer

Se videoen på <https://youtu.be/kLpQeQv8iv4>.

I videoen får du forklaret nogle vigtige begreber om Machine Learning. Du kender det måske som AI eller kunstig intelligens. Læg mærke til:

1. Machine Learning (*ML*) - og forskellen mellem almindelige computerprogrammer og *ML*
2. Træningsdata
3. Forudsigelse

## Machine Learning i programmet Orange

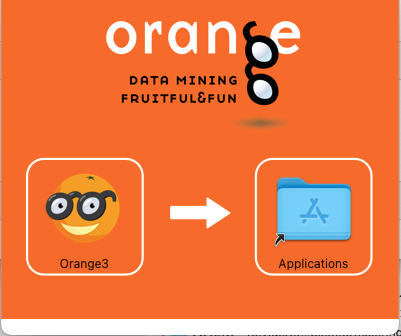
Der findes flere forskellige programmer, som kan bruges til at lave ML. Vi har valgt at bruge programmet Orange, som er baseret på Python. Programmet er valgt fordi det er baseret på blokprogrammering, hvilket gør det relativt let at gå til.

### Installer programmet Orange

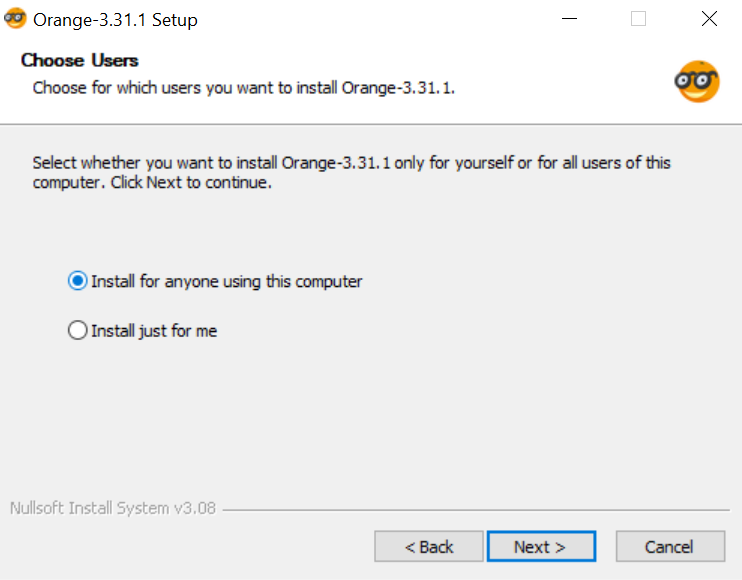
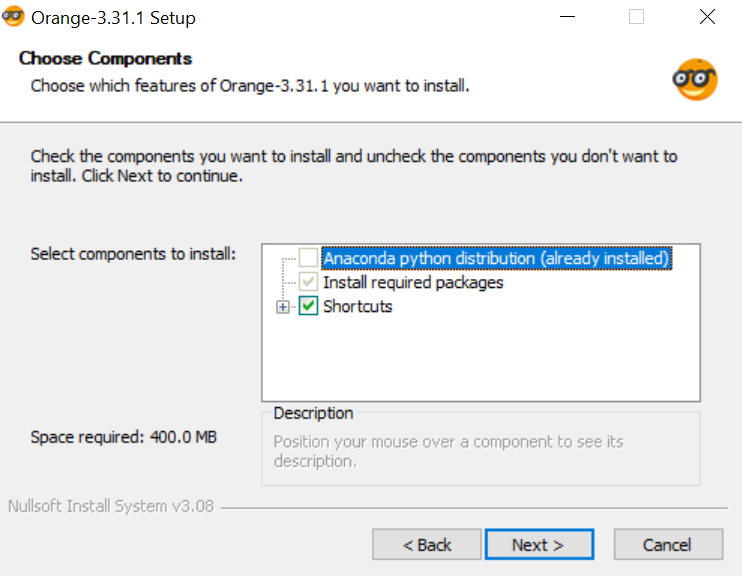
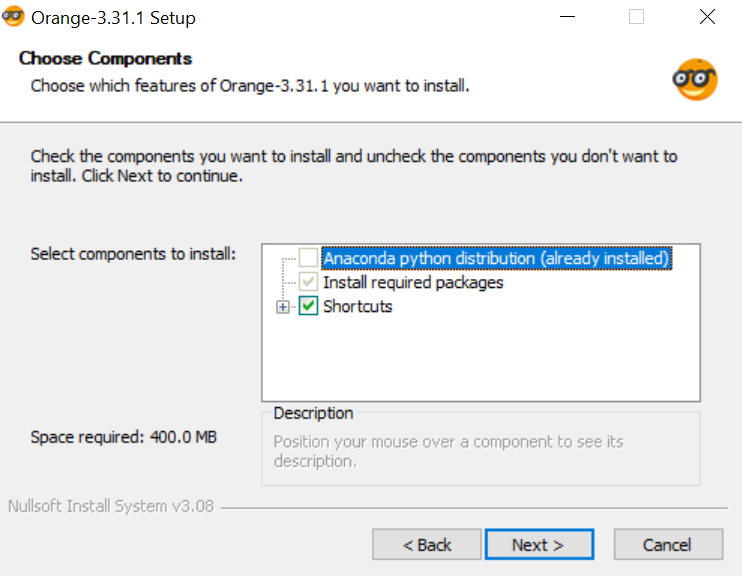
Programmet [Orange hentes her](https://orangedatamining.com/download/)

For at installere programmet skal du gøre følgende:

#### På mac

1. Download programmet
2. Åben dmg-filen i din overførselsmappe
3. Træk Orange ikonet til din Applications mappe  
   

#### På Windows

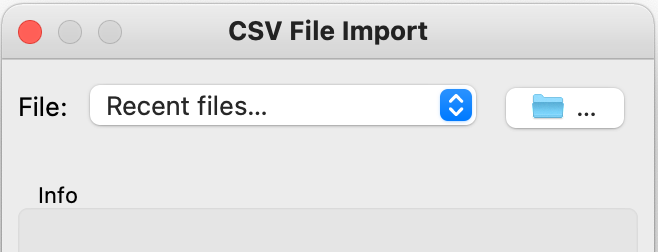
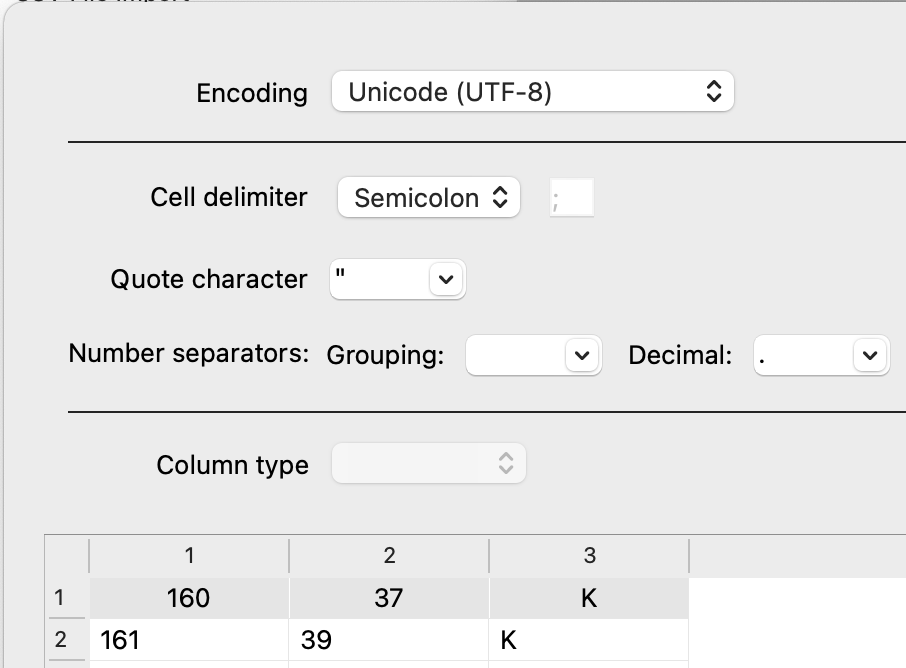
* Download programmet
* Åbn Orange Setup-programmet og følg instruktionerne - check på billederne nedenfor, hvad du skal vælge
* Vælg: “Install for anyone using this computer”  
  
* Accepter “installation af required packages” - også Anaconda  
  
* Tryk “Next” ved de næste step af installationen
* Afslut Setup-programmet  
  

Efter installation af Orange skal du se denne video med en introduktion til programmet: <https://youtu.be/6lskSQZNgDQ>.

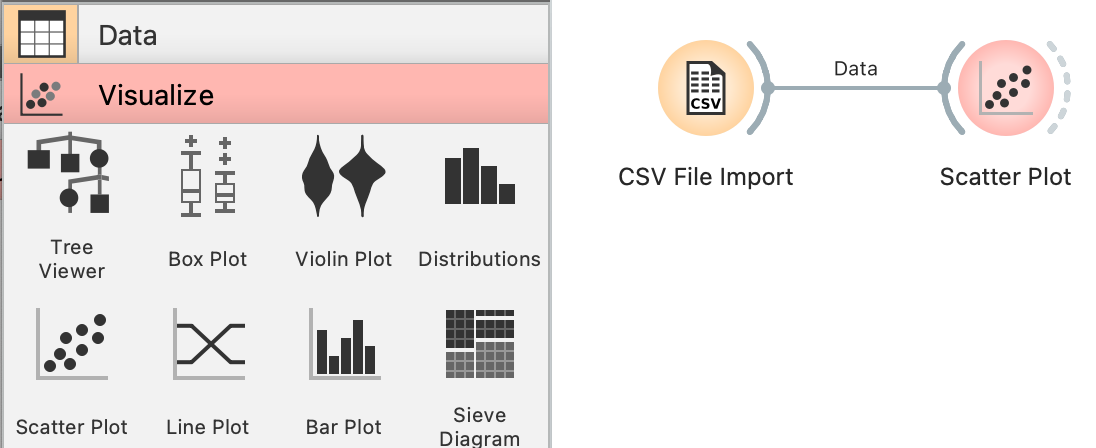
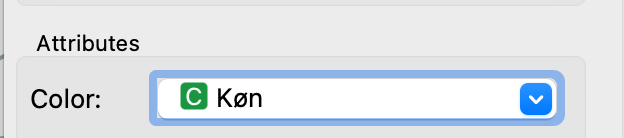
### Import data

Åben programmet Orange - der ligger en video om “import af data” i slutningen af dette afsnit.

#### Opgave: Hent data ind i Orange

1. I Lectio under ”Dokumenter🡪Hold🡪f21htxbde\_2bde\_STO🡪Machine learning” finder du filen "**HoejdeSko.csv**". Hent filen ned på din computer.
2. Tilføj “**CSV File Import**” til dit workflow - du finder ikonet til venstre i paletten “**Data**”
3. Dobbelt klik på “**CSV File Import**” - klik derefter på **mappen med de tre prikker**
4. Find filen “**HoejdeSko.csv**” på din computer
5. Vælg “**Semicolon**” ud for “**Cell delimiter**” - tallene placeres nu i 3 kolonner
6. Klik "Ok"

#### Opgave: Vis de importerede data

1. Tilføj “**Scatter Plot**” fra paletten “**Visualize**”
2. Forbind “**CSV File Import**” til “**Scatter Plot**”  
    
3. Dobbelt klik på “**Scatter Plot**” - nu kan du se data i et plot og vælge, hvordan du vil undersøge data. Du kan vælge hvilke egenskaber som vises som x- og y-værdier. Prøv at sætte “**Color**” eller "**Label**" til Køn.  
   
4. Undersøg de importerede data og besvar spørgsmålene nedenfor

Hvis du har brug for yderligere information kan du se videoen på <https://youtu.be/qB0MZbeHYsc>. I videoen vises det hvordan data importeres og hvordan dataene derefter vises.

* 1. Hvilken skostørrelse har den mand, som har den største skostørrelse?
     + 40 eller mindre
     + 41
     + 42
     + 43
     + 44
     + 45
     + 46
     + 47
     + 48 eller større
  2. Hvilken højde har den højeste kvinde?
     + 170-174 cm
     + 175-179 cm
     + 180-184 cm
     + 185-189 cm
     + 190 cm eller højere

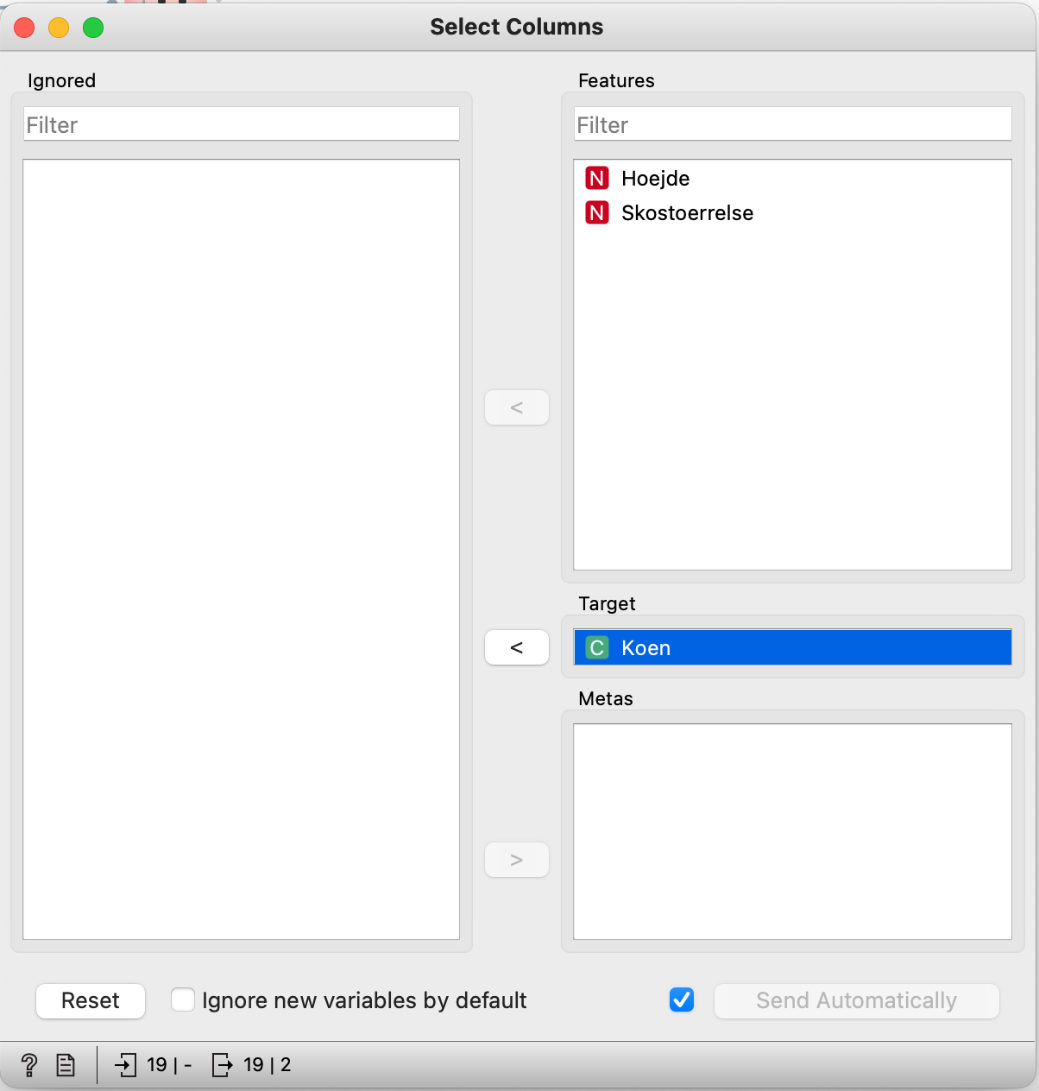
### Vælg den kategori du vil forudse

#### Forudse kategorier

Ofte træner man Machine Learning modeller til at forudse om en person eller en ting passer i en bestemt kategori. I denne øvelse træner vi modellen med data om højde og skostørrelse og vil derudfra forudse køn. Højde og skostørrelse er egenskaber (Engelsk: **features**).

Kønnet er den kategori, vi bruger som mål (**target**) i træningen.

#### Opgave: vælg features og mål

1. Tilføj "**Select Columns**" fra paletten “**Transform**”  
   
2. Forbind “**CSV File Import**” til “**Select Columns**”
3. Dobbelt klik på “**Select Columns**”
4. Check, at de data, du vil bruge som input til forudsigelserne, står under features.
5. Flyt “**Køn**” til “**Target**” - da det er køn du vil forudse med modellen  
   
6. Luk vinduet igen

### Vælg ML-model

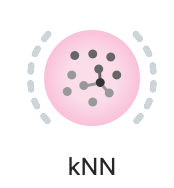
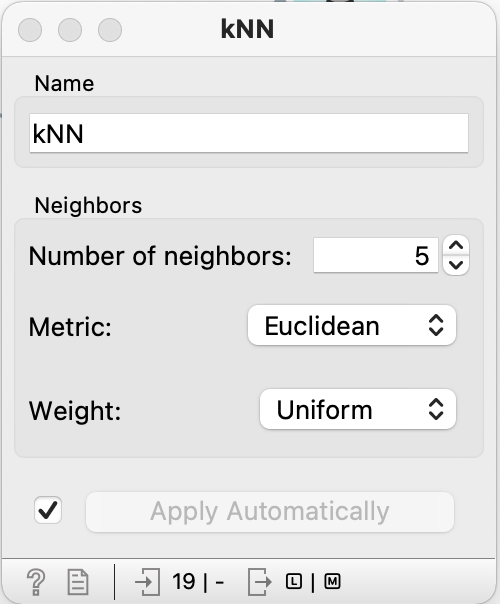
#### Machine Learning modeller

Udtrykket kunstig intelligens eller Machine Learning dækker over mange forskellige modeller. Nogle modeller er gode til at genkende ting på billeder, andre egner sig til at forudse, om vi vil vælge en film, klikke på en nyhed, eller forudse vores køn.

Modellen k-Nearest-Neighbors (*kNN*) er ofte god til at klassificere ting, når man kender nogle numeriske egenskaber for tingene. I dette eksempel prøver vi at forudse kønnet ud fra højde og skostørrelse med en *kNN*-model.

Support Vector Machine (*SVM*) og Decision Tree (*Tree*) er andre modeller, som også kan bruges til at klassificere ting. Neurale Netværk er en *ML*-model, som ofte bruges til at genkende billeder eller tolke håndskrift.

#### Opgave: Vælg ML-model

1. Tilføj “**kNN**” fra paletten “**Model**” - det er modellen k-Nearest-Neighbors  
   
2. Forbind “**Select Columns**” til “**kNN**”
3. Dobbelt klik på “**kNN**”  
    
4. Nu skal du vælge antallet af nærmeste naboer, som du vil bruge til at forudse, kønnet for en ny elev med nye værdier for højde og skostørrelse - normalt vælger vi 3 eller 5 naboer
5. Vi skal også vælge “**Metric**” dvs. den måde vi måler afstande - normalt afstandsmål er **Euclidean** - det svarer til det du kender fra matematik
6. “**Weight**” vælges til "**Uniform**" - så vægter det valgte antal naboer (fx 5) lige højt, når kønnet for en ny elev skal bestemmes. Alternativet er “Distance” - hvor de naboer der er tættest på vægtes højere i bestemmelse af kønnet for en ny elev.

### Test din ML-model

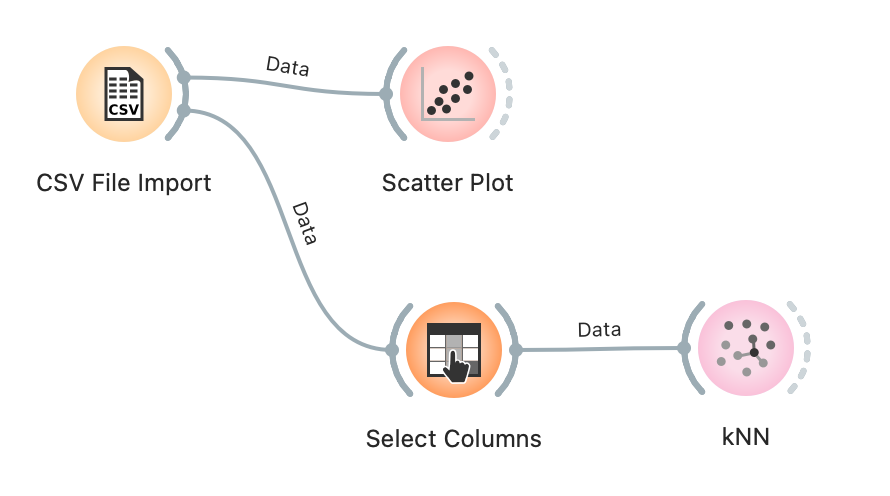
#### Først træner man, derefter tester man modellen

I afsnittet “1.3.4 Vælg ML-model” har du trænet din *kNN*-model med et træningsdatasæt. Det gør Orange automatisk, når du forbinder træningsdata til en *ML*-model.

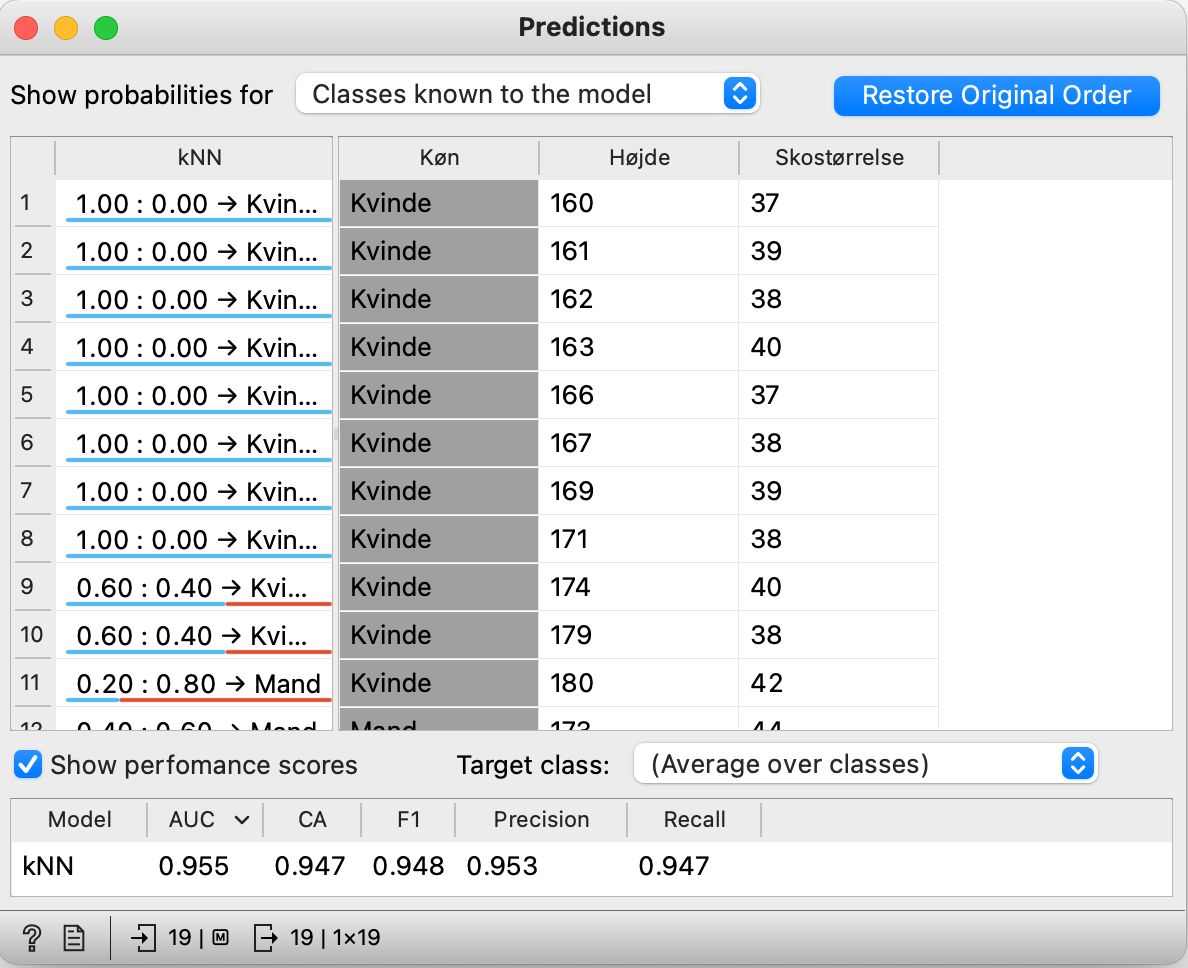
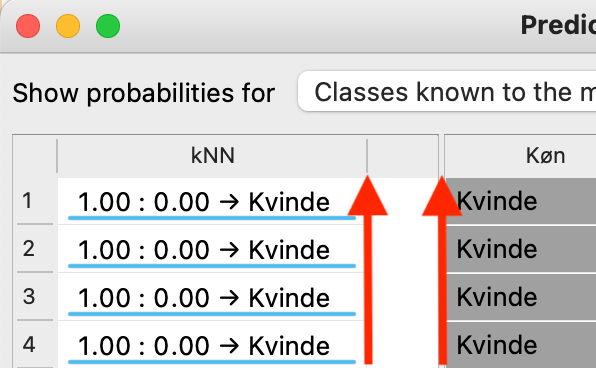
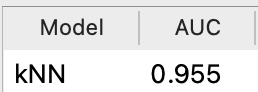
Nu skal du teste, hvor god modellen er til at forudse køn. Man måler hvor mange gange modellen laver den rigtige forudsigelse i forhold til antallet af forudsigelser . På engelsk kaldes det accuracy (AUC) og angives som et decimaltal eller i %.

Modellen vil altid være bedst, når man tester den med nogle data, som ligner træningsdatasættet. Vi starter derfor med at teste med de samme data, som vi brugte til træning af *kNN*-modellen.

#### Sådan skal workflowet se ud, inden du går videre til næste opgave



#### Opgave: Test din *kNN*-model

1. Tilføj "**Predictions**" fra paletten “**Evaluate**”
2. Forbind “**kNN**” til “**Predictions**”
3. Forbind “**Select Columns**” til “**Predictions**”
4. Dobbelt klik på “**Predictions**”   
   
5. Vælg “**Classes known to the model**” ud for “**Show probabilities for**”
6. **Kolonnen kNN** viser hvor sikker modellen er på forudsigelserne af køn.  
   For at se alle detaljer, skal du gøre første kolonne bredere. Træk der hvor pilene peger  
   
7. I linje 9 kan man se, at modellen med 60%-sikkerhed forudser, at en person på 174 cm med skostørrelse 40 vil være en kvinde. Modellen angiver samtidigt 40%-sikkerhed for, at det er en mand. Derfor forudser modellen, at denne person er en kvinde.
8. Hvor mange personers køn forudses forkert i denne model?
9. Nederst i vinduet er der nogle flere mål for modellens evne til at lave rigtige forudsigelser. AUC beregner, hvor god modellen er til at lave rigtige forudsigelser. Når det er bedst, er AUC = 1. Gætter den altid forkert er AUC = 0.  
   

### Optimer din ML-model

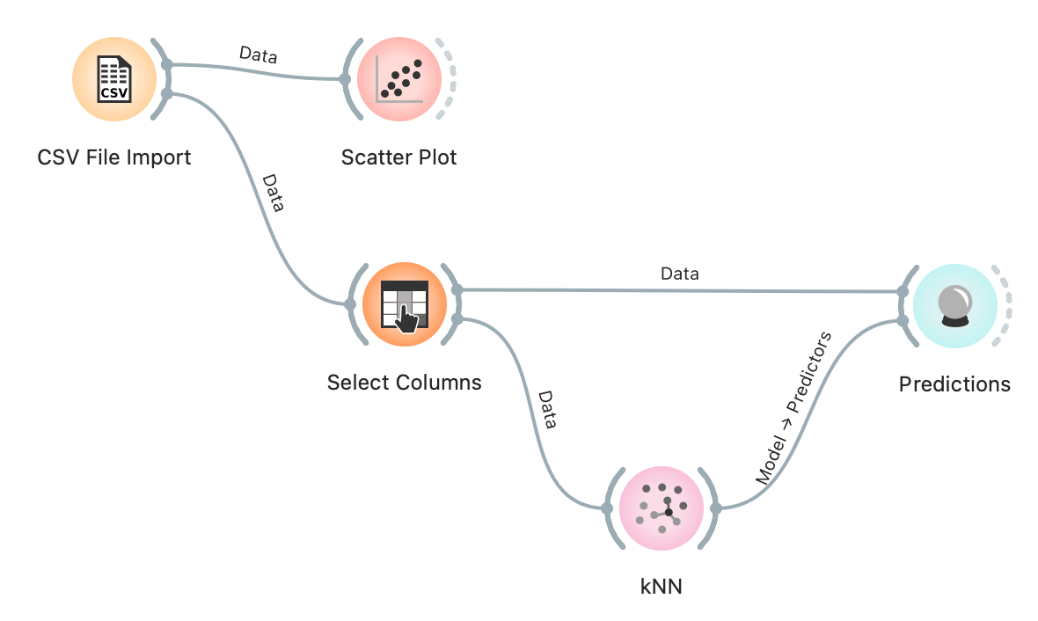
#### Optimering af Machine Learning modeller

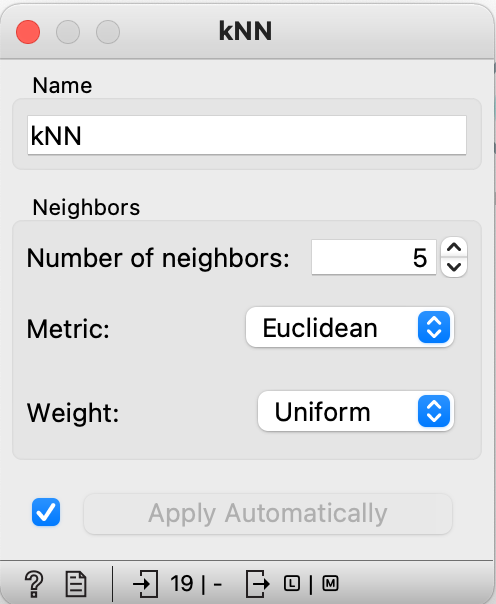
Machine Learning modeller skal optimeres, så de virker bedst muligt. Det gør man ved at justere på modellens parametre.

 I *kNN*-modeller er det bl.a. antallet af naboer, man bruger. Bruger man 5 naboer, bestemmer flertallet af de 5 naboer, hvilken kategori det nye punkt tilhører. Hvis 3 naboer er kvinder og 2 er mænd, vil modellen forudse, at det nye punkt er en kvinde.

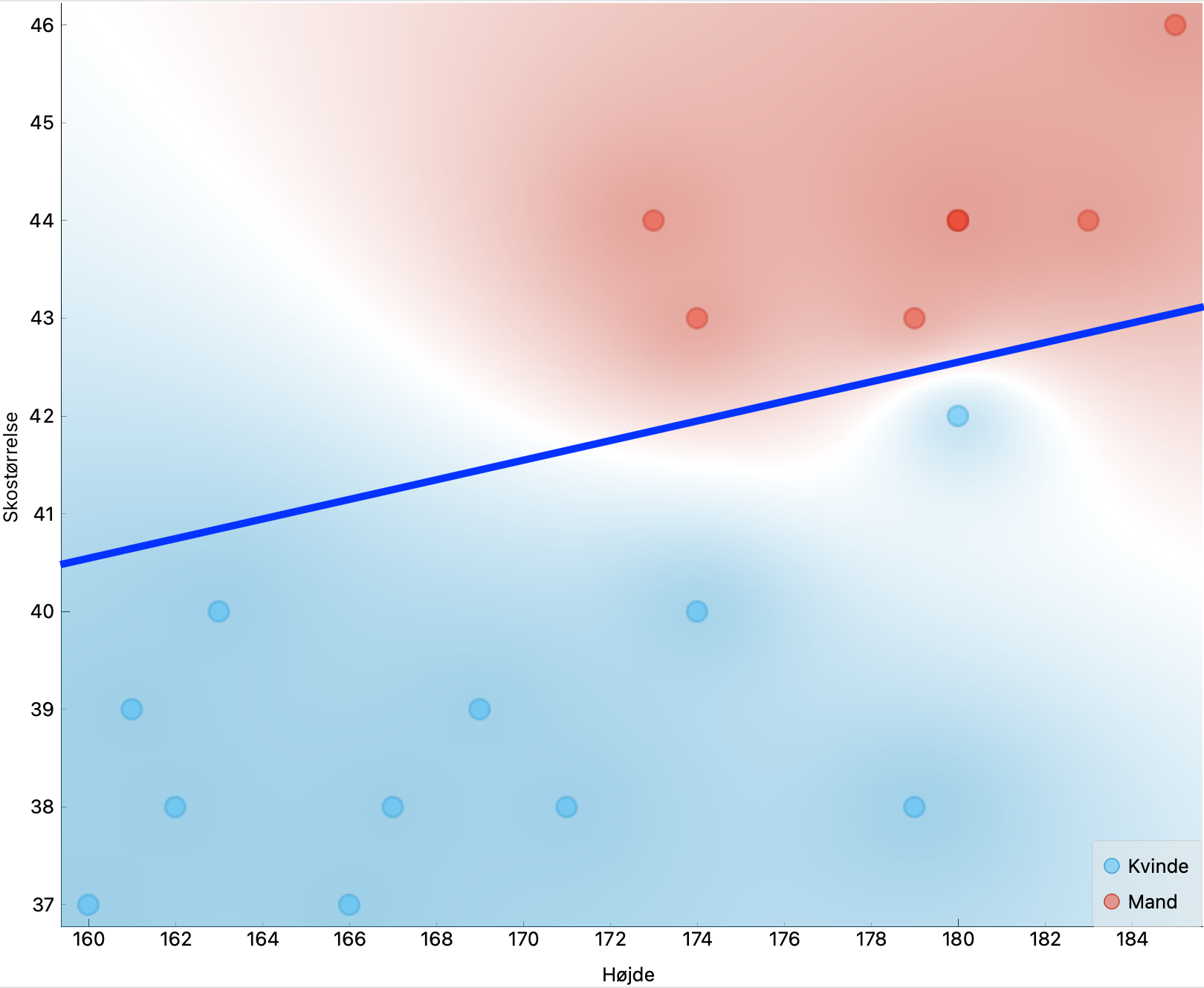
 Man kan også justere måden man måler afstand (Metric) og om de nærmeste punkter vægter mere end punkterne længere væk (Weight).

#### Opgave: hvor mange naboer skal vi bruge i kNN?



1. Dobbelt klik på “**kNN**” - her ændrer du parameter til modellen  
   Derefter kan du se resultatet i "**Predictions**"  
   
2. Undersøg om modellen bliver dårligere eller bedre til at forudse køn ud fra højde og skostørrelse, når modellen bruger **1, 2, 3, 4 eller 5 naboer** til at forudse kønnet.
3. Undersøg om modellen bliver bedre ved at sætte “**Weight**” til "**Distance**", når du bruger 5 naboer. “Distance” betyder, at naboer tæt på, vægter højere end naboer længere vægt, når modellen skal forudse kønnet for et nyt datapunkt.

#### Opgave: Support Vector Machine



I denne opgave skal du undersøge, om *SVM*-modellen fungerer bedre end *kNN*. *kNN*-modellen bruger de *k* nærmeste naboer i forudsigelser med modellen.

*SVM*-modellen deler koordinatsystemet op i områder.  Træningen af modellen svarer til at bestemme områderne, dvs. hvor skal den blå linje placeres.

Når modellen bruges til forudsigelser, undersøger den, på hvilken side af den blå streg det nye punkt ligger.

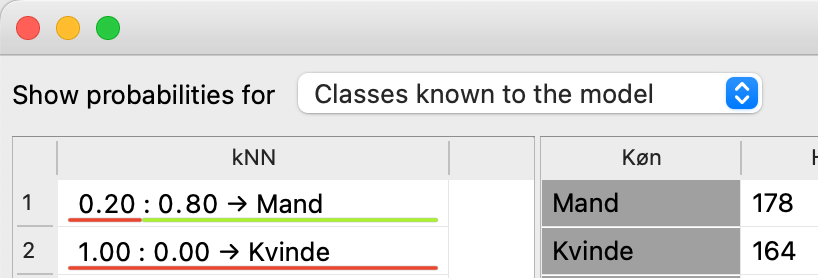
1. Tilføj “**SVM**” fra paletten “**Model**”
2. Forbind “**Select Columns**” til “**SVM**”
3. Forbind “**SVM**” til “**Predictions**”
4. Nu kan du sammenligne “**kNN**” med “**SVM**” i "**Predictions**" - dobbelt klik på "**Predictions**" og sammenlign AUC for kNN- og SVM-modellerne.

### Test med nye data

#### Testdata

Når modellen er optimeret, skal den testes igen med et helt nyt datasæt. Du finder datasættet i Lectio under ”Dokumenter🡪Hold🡪f21htxbde\_2bde\_STO🡪Machine learning”. Vi tester for at vide, hvordan modellen vil virke, hvis vi vil bruge den rigtigt.

#### Opgave: test modellen med HoejdeSko2-datasættet

1. Tilføj en ny “**CSV File Import**” fra paletten “**Data**”
2. Indlæs den nye fil **”HoejdeSko2.csv”**, som også ligger i Lectio under ”Dokumenter🡪Hold🡪f21htxbde\_2bde\_STO🡪Machine learning”.
3. Tilføj en ny "**Select Columns**" fra paletten “**Transform**”
4. Forbind den nye “**CSV File Import**” til den nye “**Select Columns**”
5. Vælg “**Køn**” til “**Target**” i “**Select Columns**”
6. Tilføj en ny “**Predictions**”
7. Forbind “**kNN**” til den nye "**Predictions**"
8. Forbind den nye “**Select Columns**” til den nye "**Predictions**"
9. Dobbelt klik på “**Predictions**” og undersøg, hvor godt modellen virker.  
   Vælg “**Classes known to the model**” og check om AUC er bedre eller dårligere med det nye datasæt.  
   

### Bias i træningsdata

En ML-model bliver aldrig bedre end de data, man har brugt til at træne modellen med.

#### Bias i træningsdata

**Bias** henviser til menneskers fordomme og stereotype opfattelser om andre. I Machine Learning henviser bias til systematiske fejl, som f.eks. kan opstå fordi træningsdatasættet ikke repræsentere omverdenen præcist.

En gruppe af “midaldrende hvide mænd” beskriver kun en begrænset del af verdens befolkning. *ML*-modeller trænet med træningsdata, som kun repræsenterer en begrænset gruppe eller inkluderer egenskaber som favorisere en bestemt gruppe, vil give et meget skævt billede af den verden som modellen skal beskrive.

# Forudse øjenfarve ud fra DNA



## Machine Learning i programmet Orange

#### Hjælpe politiet med at forudse 3 gerningsmænds øjenfarve

Forestil dig, at politiets efterforskere sender dig DNA-materiale fra 3 forskellige forbrydelser. Du har derefter behandlet DNA-sporene og undersøgt dem for mutationer, som bestemmer øjenfarven. Nu skal du lave en ML-model, der kan forudse gerningsmændenes øjenfarve.

#### DNA-fænotyping: fra DNA til gerningsmand

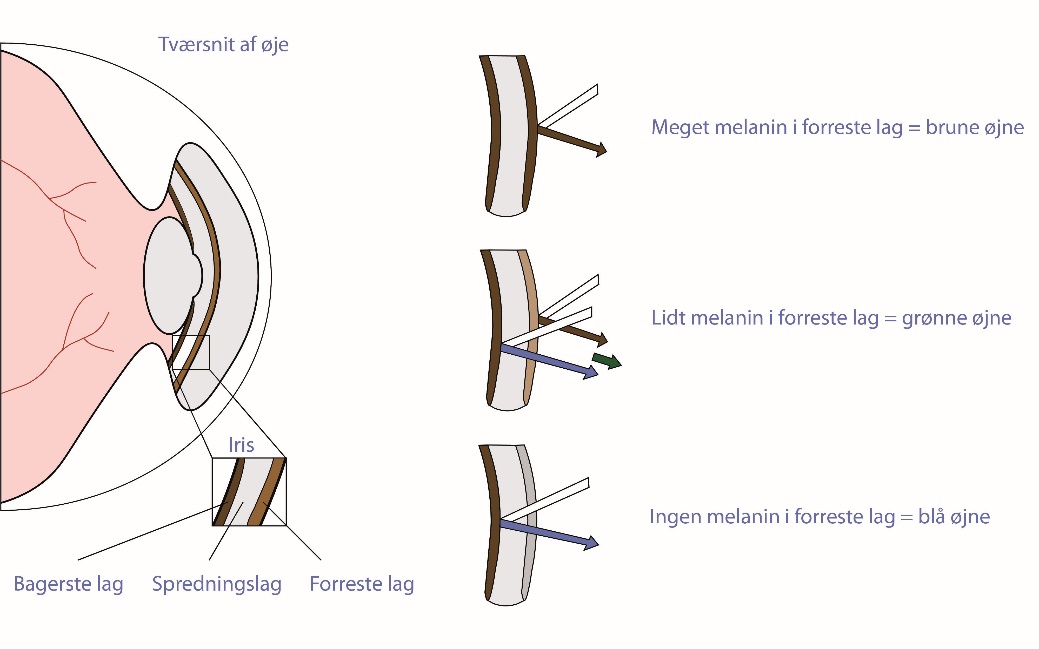
DNA-fænotyping er at bruge DNA til at forudsige, hvordan en person ser ud.  Har politiet fundet et DNA-spor som ikke matcher tidligere gerningsmænd i deres database, kan de bruge DNA-fænotyping til at få information om gerningsmandens udseende. Det svarer til politiets fantomtegninger, men her laves tegningen ud fra DNA sporets information.

#### I dette modul skal du

* læse om hvorfor øjne har forskellige farver
* læse om forskellen på statistisk og Machine Learning
* lave et program i Orange
* træne din ML-model
* teste din ML-model
* forudse øjenfarven for de 3 DNA-spor, du har modtaget
* undersøge hvilke mutationer ML-modellen bruger til forudsigelser

## Baggrundsviden: Hvorfor ser vi øjne som brune eller blå?

#### Melanin i øjnene bestemmer øjenfarven



Figur : Reflekteres lyset i det forreste lag i iris, ses en brun øjenfarve. Hvis lyset når ind i spredningslaget, sker den samme spredning af det blå lys, som vi kender fra Jordens atmosfære og giver den blå himmel. Det blå lys spredes lettere en andre farver og derfor ser himlen og øjnene ud som om de er blå, da det kun er dette lys som kastes tilbage.

Dit DNA bestemmer, hvor meget af farvestoffet melanin, der dannes i de forreste lag i din øjne. Der er flere gener, som koder for produktionen af melanin.

Nogle mennesker har mutationer i de gener, som koder for melanin. Derfor har de nedsat eller ingen melanin i øjnene, og øjnene er grønne eller blå.

I dette modul skal vi lave en Machine Learning model, som kan forudse øjenfarve ud fra mutationer i de gener, som styrer produktionen af melanin.

## Forskellen på statistik og Machine Learning

I matematik har du lært at lave statistik. Du kan beregne gennemsnittet af kvinders højde eller lave et boksplot, som viser fordelingen af højder. Du kender også sandsynligheds regning, som du kan bruge til at forudse sandsynligheden for, at en tilfældig person er 172 cm høj eller har blå øjne.

Machine learning går skridtet videre. Her kan du ud fra nogle oplysninger om personen, forudse andre egenskaber. Computeren undersøger data om mange personer og laver en model ud fra data. Når modellen derefter fordres med data om en ny person, kan den f.eks. forudse personens øjenfarve.

#### Statistik

I statistik beskriver man data med frekvens, middelværdi, spredning og andre statistiske deskriptorer. F.eks. kan øjenfarven for befolkningen i Danmark beskrives i en tabel med frekvensen for hver øjenfarve.

|  |  |
| --- | --- |
| **Øjenfarve** | **Frekvens** |
| Blå | 70% |
| Grøn | 14% |
| Brun | 16% |

#### Sandsynlighedsregning

Her kan vi besvare spørgsmål af typen:

1. Hvor stor er sandsynligheden for, at en tilfældigt valgt person i Danmark har blå øjne?
2. Hvor stor er sandsynligheden for, at en blåøjet mor og en brunøjet far får et blåøjet barn?

Svaret på det første spørgsmål er “sandsynligheden for at en tilfældigt valgt person har blå øjne er 70%”. Sandsynlighedsregning bygger ofte videre på statistikken. Man skal først observere mange personers øjenfarve = statistik, før man har en sandsynlighed for, at en persons øjenfarve er blå.

#### Korrelationer i statistik

Ofte undersøger man sammenhængen mellem 2 hændelser, f.eks. sammenhængen mellem mutationer i vores DNA og øjenfarve. Hvis alle eller næsten alle med blå øjne også har den samme mutation i DNA, siger man, at der er en sammenhæng. Sammenhængen kalder man en korrelation.

Korrelation forklarer ikke sammenhængen mellem hændelser. Kun at de to hændelser optræder hos de samme mennesker. Hvis mutation A giver blå øjne og mutation B giver lys hud, vil vi ofte se, at personer med blå øjne også har mutation B, selv om mutation B ikke påvirker øjenfarven.

#### Genome-wide association study (GWAS)

Søger man efter den genestiske årsag til en sygdom eller f.eks. øjenfarve, skal man først finde de DNA-mutationer, som optræder sammen med sygdommen. Man finder altså korrelationer (associations) mellem bestemte mutationer i DNA'et og den hændelse, at personen har en bestemt egenskab eller sygdom.

Korrelationerne hjælper med at finde de mutationer, som kan være årsag til sygdommen. Man skal derfor undersøge færre mutationer for at finde årsagen til sygdommen, da man kun behøver at undersøge betydningen af de mutationer, der optræder sammen med sygdommen.

Med resultaterne fra GWAS kan man igen svare på sandsynlighedsspørgsmål, f.eks. hvor stor er sandsynlighed for, at en tilfældig person med mutation A, B og C har blå øjne.

#### Machine Learning

I Machine Learning (*ML*) går vi skidtet videre. Med Machine Learning vil vi gerne kunne forudse om en bestemt person har en bestemt sygdom eller en bestemt egenskab f.eks. øjenfarve. I *ML* træner man en *ML*-model på en computer med DNA-data for nogle personer, hvor øjenfarven også er kendt.

Efter træning af *ML*-modellen, kan den forudse en persons øjenfarve, når *ML*-modellen får DNA-data om en person. Finder politiet et DNA-spor fra en gerningsmand, kan man derfor forudse gerningsmandens øjenfarve f.eks. blå øjne og dermed udelukke alle mistænkte med brune øjne.

*ML*-modeller kan bruge mange forskellige metoder. Nogle metoder kan vi forstår og i princippet regne efter i hånden. Det gælder *kNN* og *SVM*, som vi brugte i 1. modul. Andre *ML*-modeller kalder vi black-boks, fordi vi ikke kan forstå præcis, hvordan *ML*-modellen når frem til resultaterne ud fra træningsdata. Det gælder f.eks. Neurale netværk.

|  |
| --- |
| ****Vil du vide mere om Machine Learning og Neurale netværk?**** Elements of AI tilbyder et [gratis online kursus](https://www.elementsofai.dk), hvor du bliver introduceret til Machine Learning.  På [3blue1brown](https://youtu.be/aircAruvnKk) kan du få forklaret, hvordan et neuralt netværk genkender håndskrevne tal. |

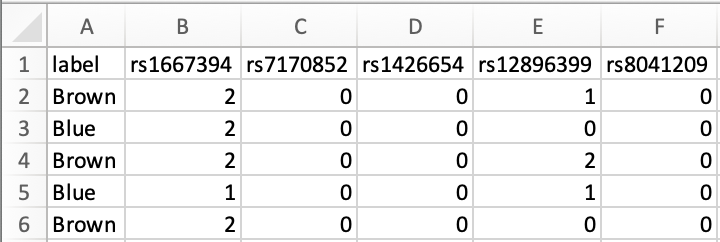
## Øjenfarve og mutationer

#### Genetisk mutationer

Du har to kopier af hvert gen i dit DNA. Den ene kopi af genet har du arvet fra din mor og den anden fra din far. Har du en mutation, kan mutationen findes enten i den ene kopi eller i begge kopier af genet. Har du arvet mutationen fra begge forældre, har du altså den samme mutation i begge kopier af genet.

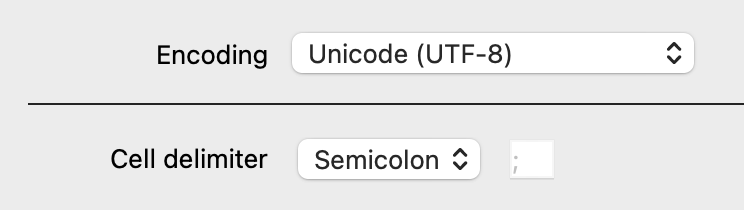
I det datamateriale, du skal arbejde med, er mutationerne navngivet med **rs** efterfulgt af 8 tal. Hver række svarer til en person. Label angiver øjenfarve for personen. For hver person angives om mutationen findes i ingen (0), den ene (1) eller begge kopier (2) af genet.

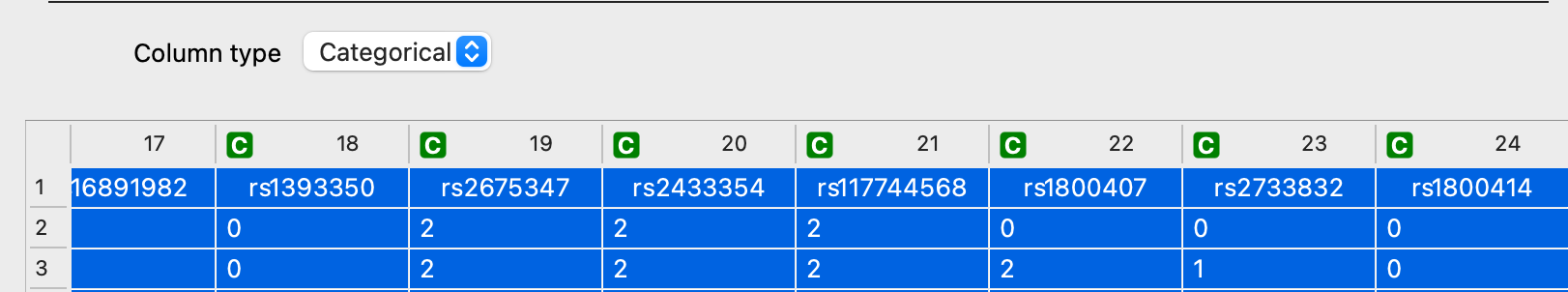
I den første opgave skal du undersøge, hvilke mutationer, som optræder oftere hos personer med blå øjne eller hos personer med brune øjne.



#### Opgave 1: indlæs DNA-data om mutationer, samt øjenfarve

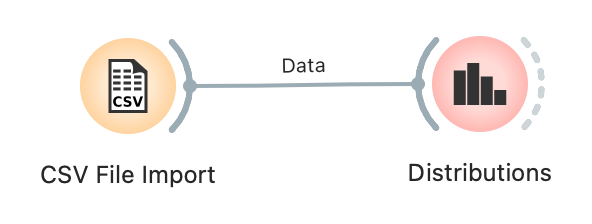
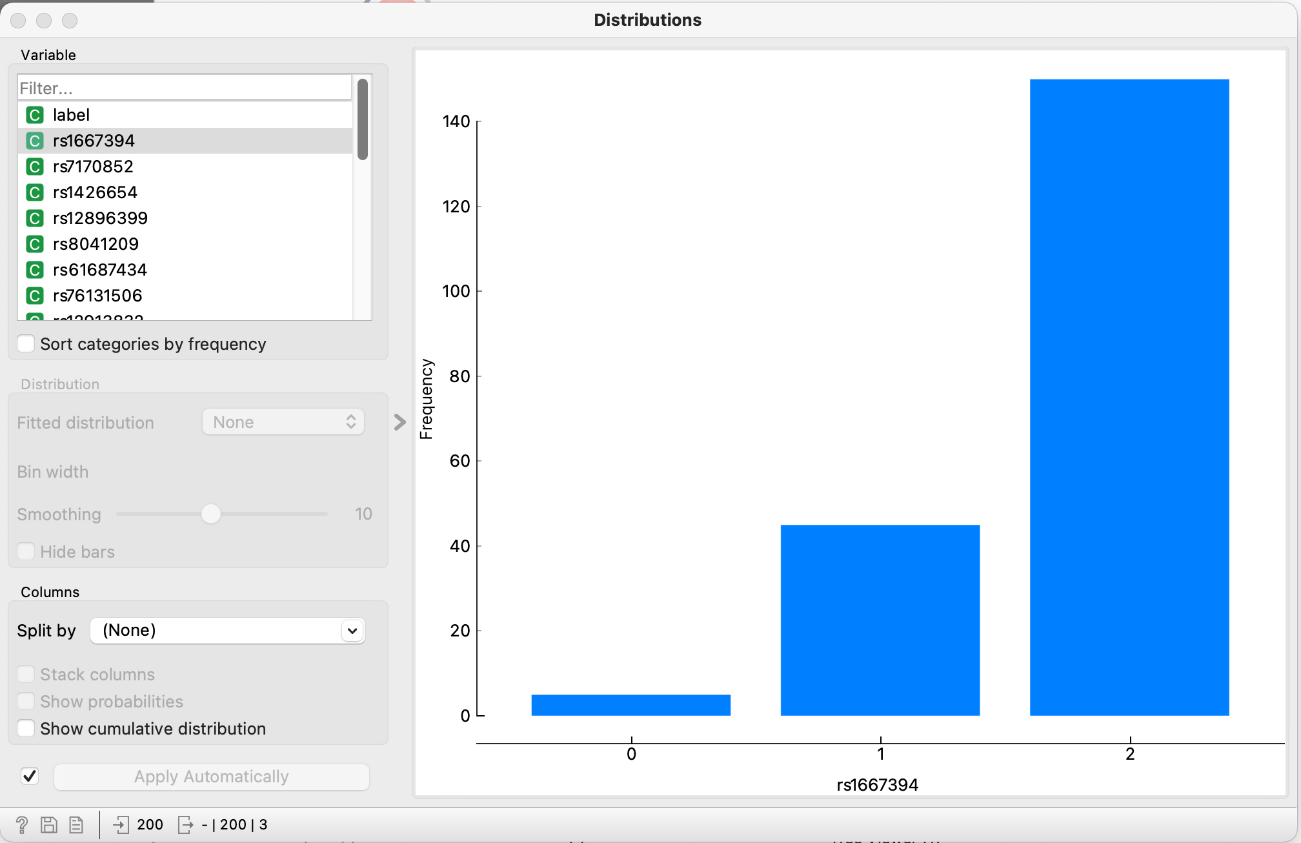
Åben et nyt workflow i programmet Orange.

1. Tilføj “**CSV File Import**” fra paletten “**Data**”
2. Dobbelt klik “**CSV File Import**” og vælg filen “**Træningsdata\_øjenfarve.csv**” – filen ligger i Lectio under ”Dokumenter🡪Hold🡪f21htxbde\_2bde\_STO🡪Machine learning”
3. Vælg "**Semicolon**" ud for "**Cell delimiter"**  
   
4. Marker **alle** kolonner og vælg "**Categorical**" ud for “**Column type”**

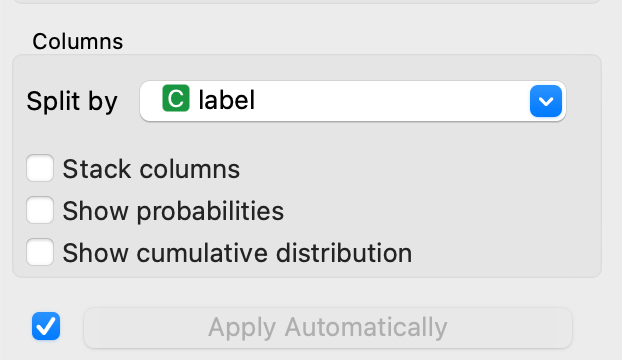
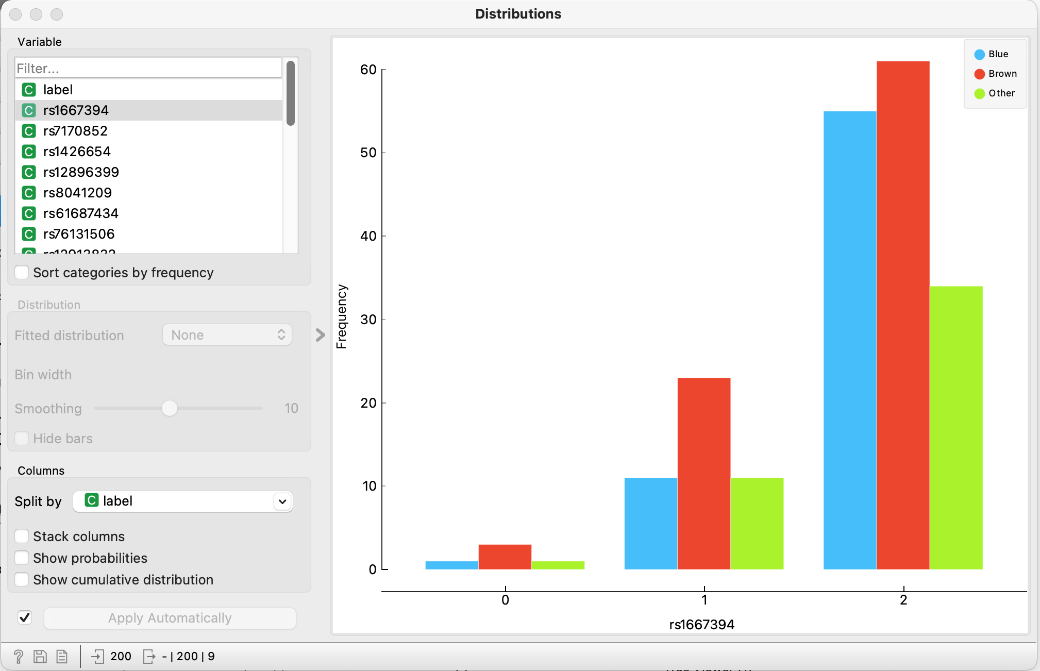


1. Klik ok
2. Du kan altid komme tilbage til dette vindue ved at klikke på “I**mport Options…**” i "**CVS File Import**"-vinduet.
3. Huske at gemme dit workflow, så du ikke mister hvad du har lavet

#### Opgave 2: Statistik over mutationer

1. Tilføj “**Distributions**” fra paletten “**Visualize**”
2. Forbind “**CSV File Import**” og “**Distributions**”  
   
3. Dobbelt klik på “**Distributions**”
4. I Distributions-vinduet kan du undersøge fordelingen for antallet af mutationer for de forskellige gener. Her kan du se fordelingen for mutationen rs12913832. Der er lidt mere end 100 personer som har mutationen på begge kopier af genet, og lidt færre end 20 personer som ingen mutationer har på dette gen.  
   
5. Undersøg de forskellige mutationer på listen

#### Opgave 3: Sammenhængen mellem mutationer og øjenfarve

1. Nederst i Distributions-vinduet sættes “**Split by**”  til “**label**”  
   
2. Nu skal du undersøge alle mutationerne og se om mutationerne optræder forskelligt for de forskellige øjenfarver. Vælg én mutation, 1 efter 1, og noter de mutationer, hvor der er stor forskel på de 3 øjenfarver  
   
3. Noter de 3 mutationer du synes er mest betydningsfulde for bestemmelsen af øjenfarve og sammenlign med sidemandens.

## Machine Learning – Decision tree i Orange

### Træn din Machine Learning model

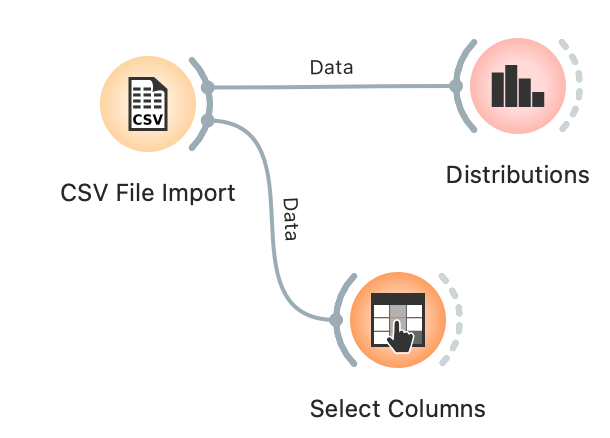
#### Forudsigelse af øjenfarve vha. DNA-spor

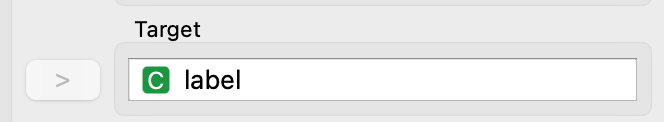
Først skal du træne en Machine Learning model, som du kan bruge til at forudse øjenfarve ud fra nye DNA-spor.

#### Opgave: Træning af modellen

Åben dit workflow fra aktivitet "2.4 - Øjenfarve og mutationer"

1. Tilføj “**Select Columns**” fra paletten “**Transform**”
2. Forbind “**CSV File Import**” og “**Select Columns**”



1. Dobbelt klik på “**Select Columns**”
2. I “**Select Columns**”-vinduet flyttes “**label**” til Target-feltet - dermed har du valgt at modellen skal trænes til at forudse øjenfarve - luk vinduet  
   
3. Tilføj “**Tree**” fra paletten "**Model**"
4. Forbind “**Select Columns**” og “**Tree**”

#### Hvordan virker en træ-model

I 1. modul arbejdede du med *ML*-modellerne *kNN* og *SVM*. I dette modul bruger du *Decision Tree* (beslutningstræ). Det passer bedre til de data vi har her, da alle features har værdierne 0, 1 eller 2.

Træ-modeller deler træningsdatasættet i mindre og mindre grupper, indtil alle eller næsten alle i gruppen har sammen label. Her deler vi ud fra mutationerne, indtil alle i en gruppe har samme øjenfarve.

For hver mutation udregner træ-modellen, hvor stor procentdel der har brun, blå og anden øjenfarve. For rs12913832 kan du se i tabellen at både 0 og 1 mutation stort set kun forekommer for brun og anden øjenfarven. Godt halvdelen med 2 mutationer har blå øjne.

Træ-modellen deler derfor personerne i 2 grupper. Dem med 0 og 1 mutation (mest brune og andet) og dem med 2 mutationer (fleste med blå). Derefter undersøges de 2 grupper hver for sig. Og grupperne deles i mindre grupper vha. mutationer for andre gener.

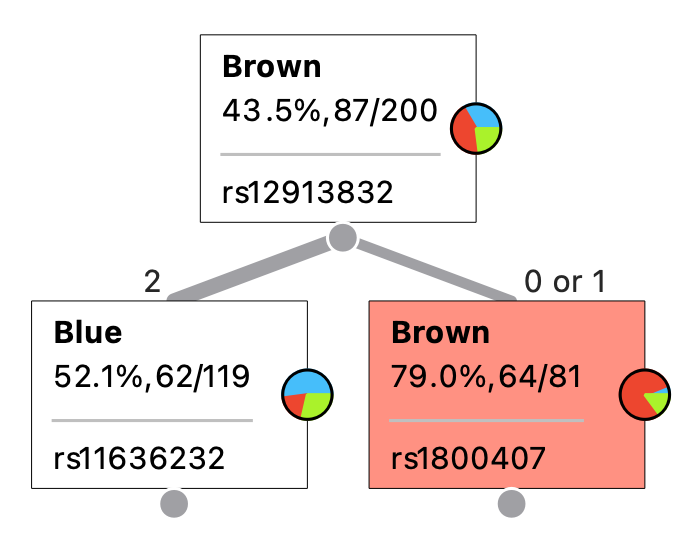
|  | **Øjenfarve** | | |
| --- | --- | --- | --- |
| **Antal mutationer for rs12913832** | **Blå** | **Brune** | **Andet** |
| **0** | 0% | 100% | 0% |
| **1** | 7,8% | 73,4% | 18,8 |
| **2** | 52,1% | 19,3% | 28,6% |

#### 

#### Illustration af Decision Tree

Træ-modellen vises som et omvendt træ, hvor man begynder øverst. I den øverst boks vises at 43,5% af de 200 personer i datasættet har brune øjne. Derefter deles datasættet ud fra mutationen rs12913832. Den ene gruppe består af 81 personer med 0 eller 1 mutation på rs12913832. Af dem har 79% brune øjne. Af de 119 med 2 mutationer har 52% blå øjne.

Når man bruger modellen til forudsigelser, følger man træet fra top til bund. Hvis vi begynder med en person med 1 mutation på rs12913832, går vi til højre i træet. Da der står “Brown” på højre boks vil vi forudse at personen med 79% sikkerhed har brune øjne. I afsnit 2.6 kommer vi mere ind på træ-modellen og her skal du også lave selve træet.

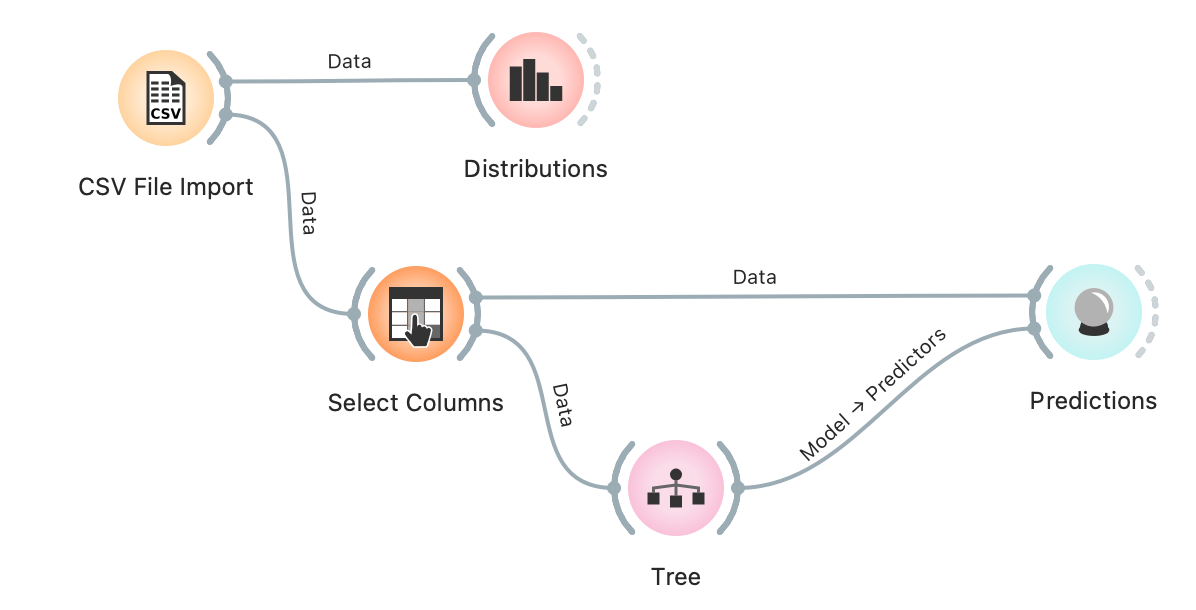


### Test din Machine Learning-model

#### Forudsigelse af øjenfarve vha. DNA-spor

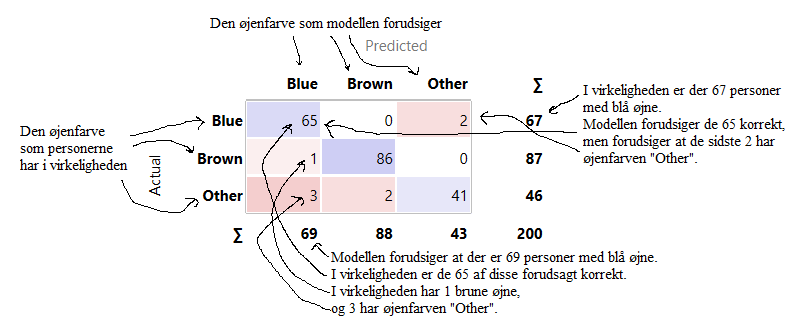
Nu skal du teste din Machine Learning model for at se, hvor god den er til at forudse øjenfarve ud fra DNA-spor.

#### Opgave: test din Machine Learning model

1. Tilføj “**Predictions**” fra paletten “**Evaluate**”
2. Forbind “**Select Columns**” til “**Predictions**”
3. Forbind “**Tree**” til “**Predictions**”  
   
4. Dobbelt klik på “**Predictions**” og undersøg, hvor god modellen er til at forud se øjenfarve  
   - gør det samme som i aktivitet “1.3.5 Test din ML-model” i 1. modul.
5. Svar på spørgsmålene nedenfor ud fra “**Predictions**”-vinduet
   1. Hvilken øjenfarve forudser modellen for person nr. 1?
      * Brown
      * Blue
      * Other
   2. Hvilken sikkerhed giver modellen for denne forudsigelse af øjenfarven for person nr. 1?
   3. Hvilken øjenfarve forudser modellen for person nr. 15?
      * Brown
      * Blue
      * Other
   4. Hvilken sikkerhed giver modellen for denne forudsigelse af øjenfarven for person nr. 15?
   5. Er forudsigelsen for person nr. 15 korrekt?
      * Ja
      * Nej

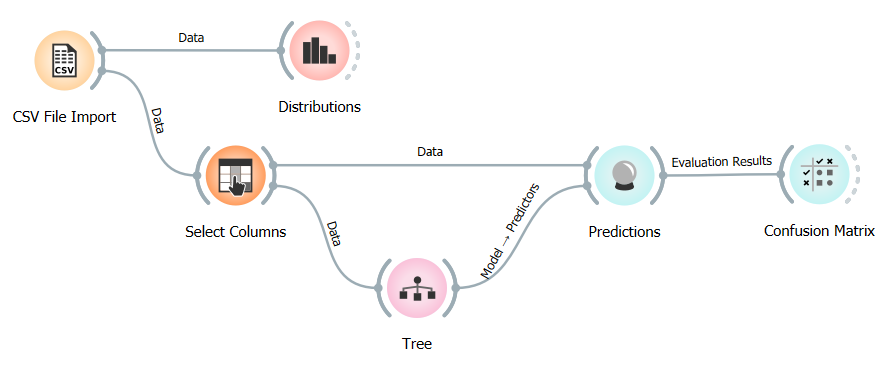
### Evaluering med Confusion matrix

I stedet for at undersøge hver enkelt person i datasættet og finde ud af om deres øjenfarve er forudsagt korrekt kan man tilføje en såkaldt confusion matrix, som vist på billedet nedenfor. I denne matrix er modellens forudsigelser (”Predicted”) sammenlignet med den øjenfarve, som personerne har i virkeligheden (”Actual”). Tolkningen af tallene i matrixen er også angivet i figuren.



#### Opgave: Tilføj en confusion matrix til din model

1. Tilføj “**Confusion Matrix**” fra paletten “**Evaluate**”
2. Forbind “**Predictions**” til “**Confusion Matrix**” som vist her:



1. Dobbelt klik på “**Confusion Matrix**” for at få matrixen frem
2. Vurder om du mener at modellen er tilstrækkelig præcis til at kunne bruges i dette tilfælde.

### Hvilken øjenfarve har gerningsmændene?

#### Nu er du klar til at hjælpe politiet

Vi har 3 forskellige DNA-spor fra 3 forskellige forbrydelser. Du skal forudse øjenfarve i alle 3 sager og angive hvor sikker modellen er i forudsigelserne.

[Retsmedicinsk institut](https://retsmedicin.ku.dk/forskning/vi-forsker-i-retsgenetik/) laver analyser for politiet. De kan f.eks. hjælpe politiet med at bestemme alder, øjen-, hår- og hudfarve, samt etnisk oprindelse. Når politiet afleverer et DNA-spor fra en gerningsmand til analyse, kan de få disse oplysninger om gerningsmandens udseende.

Det er også vigtigt at beregne, hvor sikker modellen er i forudsigelsen af f.eks. øjenfarve. Hvis modellen er meget usikker på øjenfarven, kan politiet ikke bruge oplysningen som bevis i en retssag.

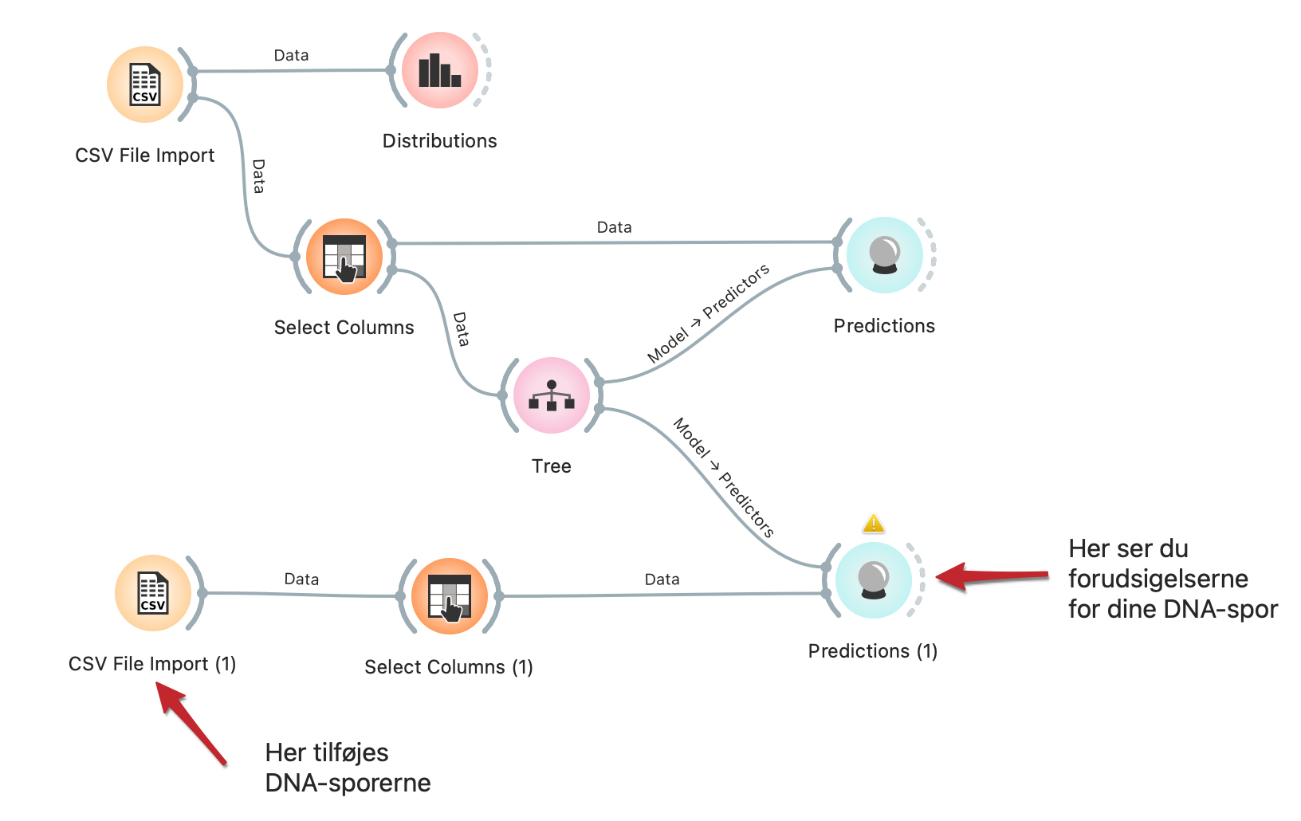
#### Opgave: Bestem gerningsmandens øjenfarve

Vi vil bruge den trænede model til at forudse gerningsmændenes øjenfarver. Du har modtaget DNA-sporene i en csv-fil. Du finder filen nederst på denne side.  
Du skal fortsætte i det workflow du har lavet i 2.5.2.

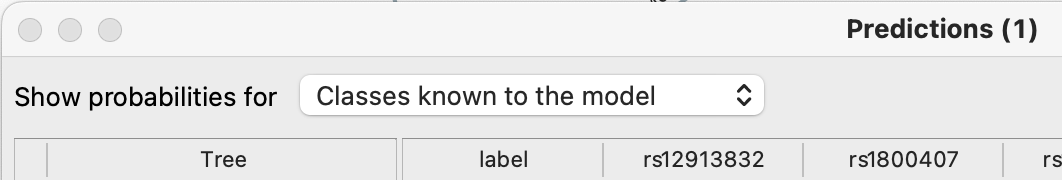
1. Tilføj “**CSV File Import**” fra paletten “**Data**”
2. Dobbelt klik på “**CSV File Import**” og indlæs det DNAsporet i filen “**DNA spor farve.csv**” - husk at vælge “**Semicolon**” i “**Cell delimiter**” og “**Categorial**” i “**Column Type**”
3. Tilføj “**Select Columns**” fra paletten “**Transform**”
4. Forbind “**CSV File Import**” og “**Select Columns**”

Nu er du klar til at bestemme øjenfarven

1. Tilføj “**Predictions**” fra paletten “**Evaluate**”
2. Forbind “**Select Columns**” til “**Predictions**”
3. Forbind “**Tree**” fra den tidligere opgave til den nye “**Predictions**”

Sådan ser dit workflow ud nu  


1. Dobbelt klik på den nye “**Predictions**”
2. I Predictions-vinduet skal du vælge “**Classes known to the model**” og gøre Tree-kolonnen bredere så du kan se sandsynlighederne



1. Hvilken øjenfarve forudser modellen for de 3 DNA-spor?
2. Hvilke sandsynligheder angives for de 3 mulige øjenfarver?

## Forstå betydningen af mutationerne

#### White box og Black box modeller i Machine Learning

Nogle Machine Learning modeller kan vi undersøge og forstå, hvordan de når frem til et bestemt svar. Vi kalder dem White Box modeller. *kNN* og *SVM* modellerne fra 1. modul er White Box modeller, fordi vi kan undersøge placeringen af træningsdata i et koordinatsystem, og hvor de forskellige kategorier er placeret i koordinatsystemet.

Neurale Netværk kan ikke analyseres på samme måde og vi kalder dem derfor for Black Box modeller. De bruges ofte til billedgenkendelse - f.eks. når computeren skal afgøre, om der er et ansigt på et foto.

I dette modul bruger du *Decision Tree* (beslutningstræ). Det passer bedre til de data vi har her, da alle features har værdierne 0, 1 eller 2. Træ-modeller er også White Box. I træ-modellen deles data i grupper ud fra forskellige features, så man hver gang får delt træningsdata bedst muligt i forhold til de kategorier man skal forudse.

#### Hvordan virker en træ-model?

Træ-modeller deler træningsdatasættet i mindre og mindre grupper, indtil alle eller næsten alle i gruppen tilhører samme kategori. Her deler vi ud fra mutationerne, indtil alle i en gruppe har samme øjenfarve.

For hver mutation udregner træ-modellen, hvor stor procentdel der har brun, blå og anden øjenfarve. For rs12913832 kan du se i tabellen, at både 0 og 1 mutation stort set kun forekommer for brun og anden øjenfarven. Godt halvdelen med 2 mutationer har blå øjne.

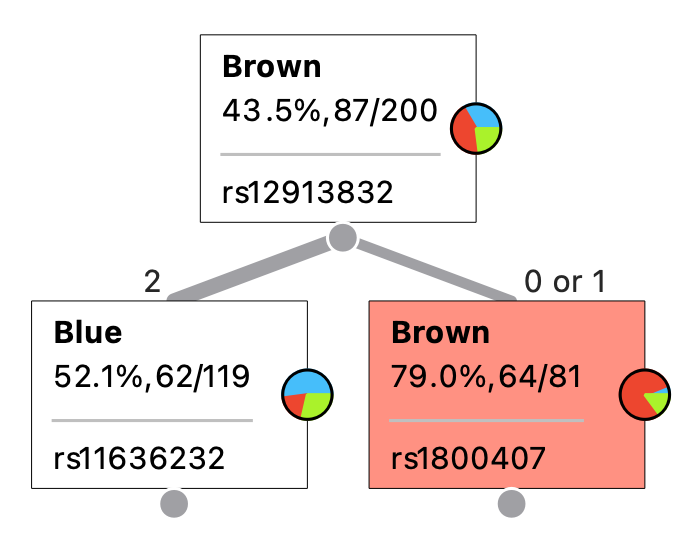
Træ-modellen deler derfor personerne i 2 grupper. gruppen med 0 og 1 mutation (mest brune og andet) , og gruppen med 2 mutationer (fleste med blå). Derefter undersøges de 2 grupper hver for sig. Og grupperne deles i mindre grupper vha. mutationer for andre gener.

|  | **Øjenfarve** | | |
| --- | --- | --- | --- |
| **Antal mutationer for rs12913832** | **Blå** | **Brun** | **Anden** |
| **0** | 0% | 100% | 0% |
| **1** | 7,8% | 73,4% | 18,8 |
| **2** | 52,1% | 19,3% | 28,6% |

#### Illustration af Decision Tree

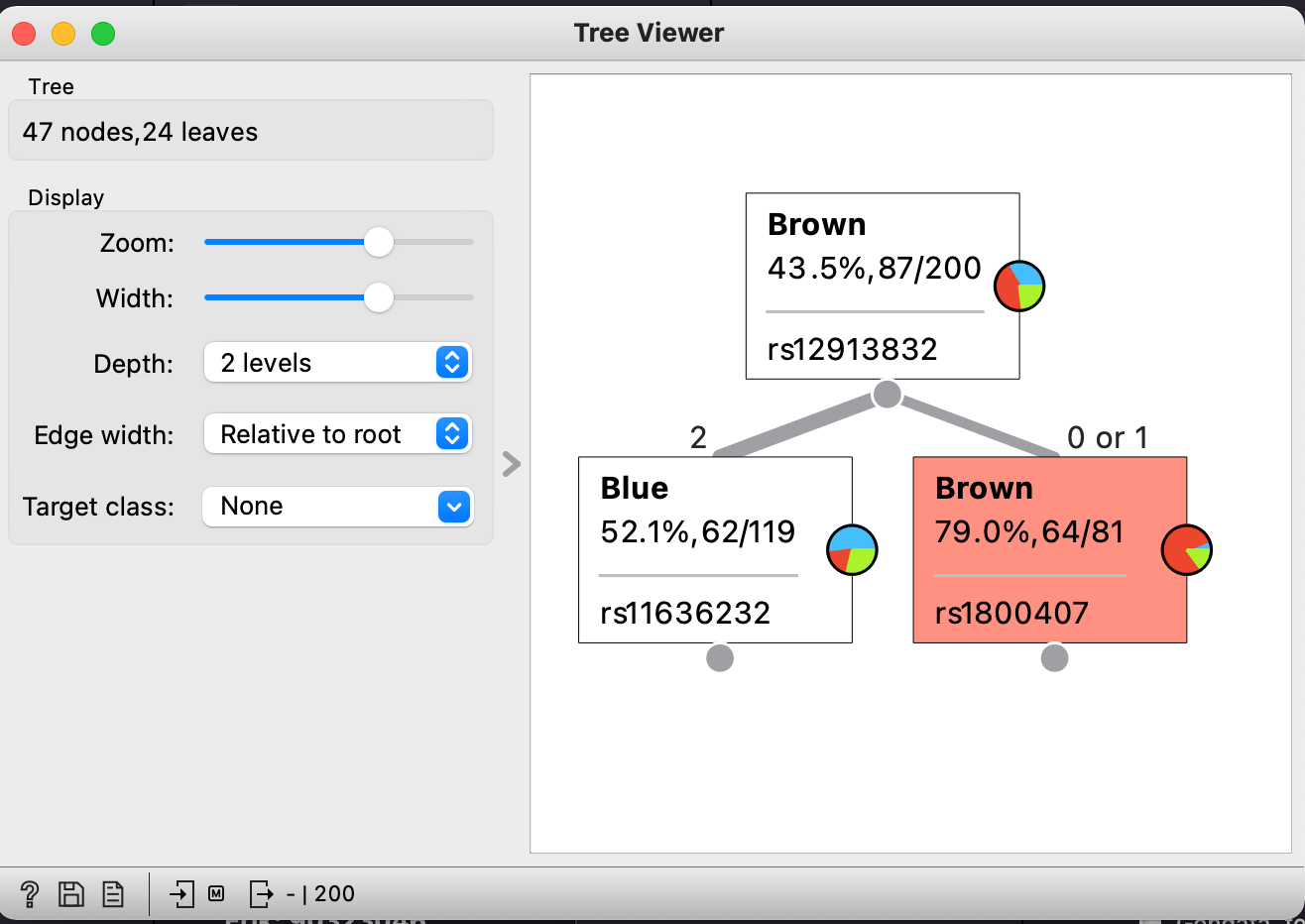
Træ-modellen vises som et omvendt træ, hvor man begynder øverst. I den øverst boks vises at 43,5% af de 200 personer i datasættet har brune øjne. Derefter deles datasættet ud fra mutationen rs12913832. Den ene gruppe består af 81 personer med 0 eller 1 mutation på rs12913832. Af dem har 79% brune øjne. Af de 119 med 2 mutationer har 52% blå øjne.

Når man bruger modellen til forudsigelser, følger man træet fra top til bund. Hvis vi begynder med en person med 1 mutation på rs12913832, går vi til højre i træet. Da der står “Brown” på højre boks vil vi forudse, at personen med 79% sikkerhed har brune øjne.



#### Opgave 1: forstå hvilke gener som bruges i forudsigelsen af øjenfarven

Åben dit workflow fra aktivitet 2.5.4

1. Tilføj “**Tree Viewer**” fra paletten “**Visualize**”
2. Forbind “**Tree**” og “**Tree Viewer**”
3. Dobbelt klik på “**Tree Viewer**”
4. I vinduet “**Tree Viewer**” kan du undersøge hvilke mutationer modellen bruger til at forudse øjenfarve.  
   De små cirkler viser fordelingen af øjenfarver i denne gruppe.  
   
5. Indstil Depth til “**2 levels**”.
6. Hvilken mutationer bruger modellen til at forudse øjenfarven?
7. Indstil Depth til “**3 levels**”.
8. Hvilke ekstra mutationer bruger modellen nu til at forudse øjenfarven?

#### Opgave 2:

Om en person får du følgende information:

* rs12913832 har 1 mutation
* rs11800407 har 2 mutationer

Hvilken øjenfarve forudser modellen?

Hvilken sandsynlighed giver modellen for denne øjenfarve?

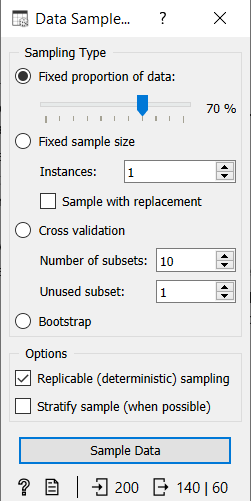
Du får nu den ekstra information, at rs7170852 har 0 mutationer.  
Ændrer denne oplysning på modellens forudsigelse?

## Opsplitning af datasæt i trænings- og evalueringsdata

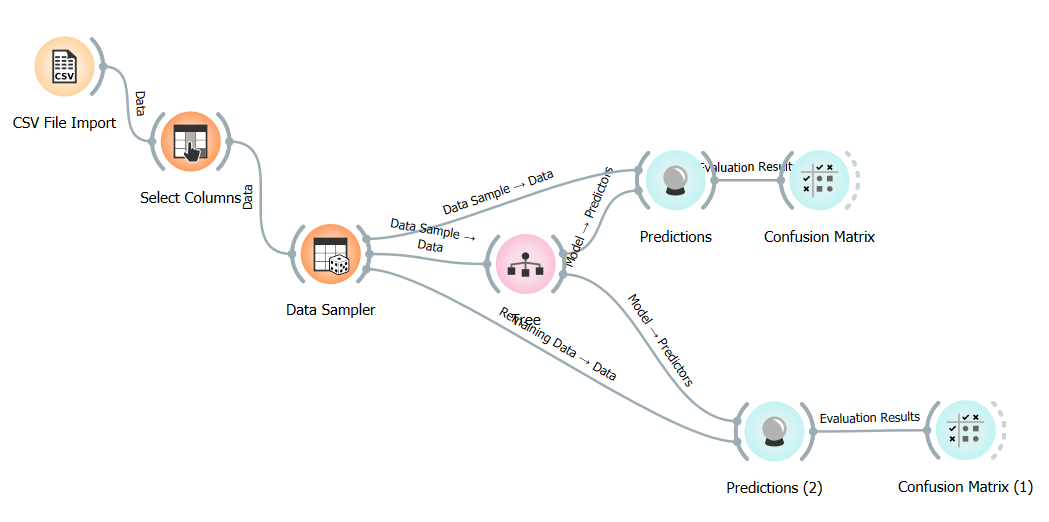
I afsnit 2.5.3 har vi evalueret hvordan beslutningstræet virkede på de data vi brugte til at træne modellen, og i afsnit 2.5.4 har vi anvendt modellen, men ofte har men et trin derimellem, hvor man evaluerer modellen på en del af træningsdatasættet, som man har taget ud inden man trænede modellen. På den måde får man et mere reelt billede af modellens anvendelighed. Nedenfor bruger vi igen datasættet for øjenfarve.

#### Opgave: opsplitning af datasæt

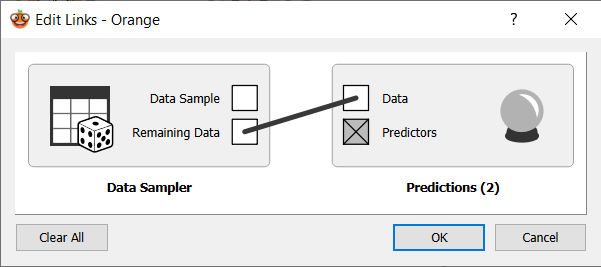
1. Åben en ny fil i Orange, og tilføj “**CSV File Import**” fra paletten “**Data**”
2. Dobbelt klik på “**CSV File Import**” og vælg filen “**Træningsdata\_øjenfarve.csv**” – filen ligger i Lectio under ”Dokumenter🡪Hold🡪f21htxbde\_2bde\_STO🡪Machine learning”, som i afsnit 2.4.
3. Tilføj “**Select Columns**” fra paletten “**Transform**”
4. Forbind “**CSV File Import**” og “**Select Columns**”
5. Dobbelt klik på “**Select Columns**”
6. I “**Select Columns**”-vinduet flyttes “**label**” til Target-feltet - dermed har du valgt at modellen skal trænes til at forudse øjenfarve - luk vinduet
7. Tilføj **”Data Sampler”** fra paletten **”Transform”**, og forbind “**Select Columns**” til **”Data Sampler”.**
8. Dobbeltklik på **”Data Sampler”** for at være sikker på at **”Fixed proportion of data”** står på 70%. Dette betyder at 70% af datasættet bruges som træningsdata og de resterende 30% kan bruges som evalueringsdata.



1. Herefter kan der tilføjes “**Tree**” og “**Predictions**” til både trænings- og evalueringsdata. Derefter skal dit workflow se ud som nedenfor



1. Læg mærke til at der på linjen ned til **”Predictions (2)”** står **”Remaining Data 🡪 Data”**, da det her er evalueringsdataene, der bruges. Det indstilles ved at dobbeltklikke på linjen og forbinde **”Remaining Data”** til **”Data”**, som vist her



1. Hermed kan Træ-modellen evalueres både i forhold til træningsdata (**”Predictions”**) og evalueringsdata (**”Predictions (2)”**).