Intro data science

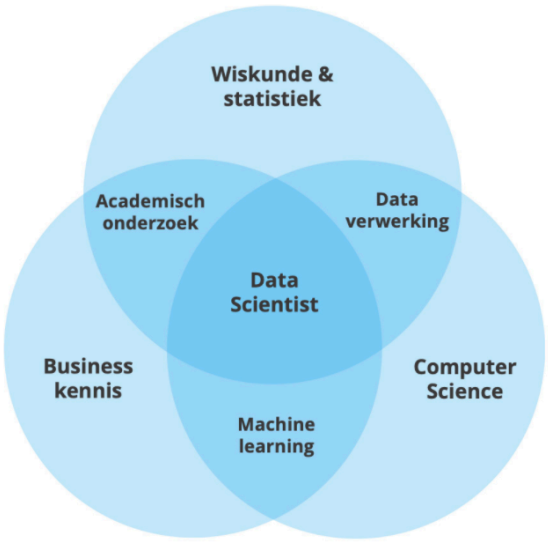
Portfolio assignment 101 & 102

Stefan Jaspers – 2152854
Docenten: Alexander van den Bulck & Dion Koeze
Deadline: 10 maart 2021

Portfolio assignment 101

Data science is een vakgebied waarbinnen het draait om het verkrijgen van inzichten uit data, welke gebruikt kunnen worden voor bijvoorbeeld een betere besluitvorming binnen een bedrijf of organisatie. Data science is interdisciplinair, wat betekent dat het een breed vakgebied is waar verschillende disciplines bij (kunnen) komen kijken.

Onderstaande afbeelding geeft aan dat data science bijvoorbeeld kan overlappen met computer science en machine learning, maar ook wiskunde en statistiek. Dit wordt allemaal in meer of mindere mate toegepast in de portfolio assignments.



Zo maken we in de portfolio assignments gebruik van Python en Jupyter Notebooks om gegevens te importeren en analyses op uit te voeren. Hierbij komen ook verschillende libraries bij kijken, zoals pandas voor data analysis en seaborn voor het visualiseren van deze data. Ten slotte komen ook SciPy en scikit-learn aan bod; respectievelijk bedoeld voor het uitvoeren van wiskundige berekeningen en machine learning.

Portfolio assignment 102

Business Understanding
Dit onderdeel komt bij ieder portfolio assignment terug. Wanneer je analyses gaat uitvoeren voor een bedrijf is het belangrijk dat je het businessmodel goed begrijpt en weet met welke doeleinden je het diepe in gaat. Als je niet goed weet hoe het bedrijf in elkaar zit is het mogelijk om goede analyses te doen.

Data Acquisition & Understanding
Ook dit onderdeel is in ieder portfolio in meer of mindere mate terug te zien. Data wordt opgehaald uit een bron. In het geval van onze assignments was dit bijvoorbeeld Seaborn (penguins dataset) of Kaggle, waar ik mijn eigen datasets vandaan heb gehaald.

Lang niet iedere dataset is meteen klaargestoomd om geanalyseerd te worden. Zo zul je soms gebruik maken van algoritmen die bijvoorbeeld niet met lege (NaN) waarden om kunnen gaan. In dat geval moet je deze eerst wegwerken, door deze bijvoorbeeld te vervangen met de gemiddelde of meest voorkomende waarde in de desbetreffende kolom(men). Dit was nodig bij portfolio assignment 15. Hier zijn alle numerical values vervangen door de mean van de kolom, en de categorical value is vervangen door de meest voorkomende value (male).

Filling the blanks

First, we will fill the missing values of each column with the mean of that column.

In [14]:
```
penguins['bill_length_mm'] = penguins['bill_length_mm'].fillna(value=penguins['bill_length_mm'].mean())
penguins['bill_depth_mm'] = penguins['bill_depth_mm'].fillna(value=penguins['bill_depth_mm'].mean())
penguins['flipper_length_mm'] = penguins['flipper_length_mm'].fillna(value=penguins['flipper_length_mm'].mean())
penguins['body_mass_g'] = penguins['body_mass_g'].fillna(value=penguins['body_mass_g'].mean())
```

In [15]:
```
penguins.head()
```

Out[15]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.10000	18.70000	181.000000	3750.000000	Male
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	Female
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	Female
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	NaN
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	Female

For sex, we will look at the most popular value and use that value to fill the missing values.

In [16]:
```
penguins['sex'].value_counts()
```

Out[16]:
```
Male      168
Female    165
Name: sex, dtype: int64
```

Looks about equal, so I will just fill the missing values with Male.

In [17]:
```
penguins['sex'] = penguins['sex'].fillna(value='Male')
```

In [18]:
```
penguins.head()
```

Out[18]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.10000	18.70000	181.000000	3750.000000	Male
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	Female
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	Female
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	Male
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	Female

Het kan ook zijn dat kolomnamen te onduidelijk of abstract zijn en je ze wilt hernoemen, of dat juist de waarden in deze kolommen niet aan je wensen voldoen. Zo heb ik in assignment 12 gebruik gemaakt van een dataset over hartziekten. Deze dataset bevat een kolom “sex” waarin de waarden 0 en 1 staan (respectievelijk man en vrouw). Om meer duidelijkheid te creëren heb ik deze waarden veranderd naar Male en Female.

```
In [62]: heart = pd.read_csv('datasets/heart.csv')

         heart.head()
```

Out[62]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

```
In [64]: heart = heart.replace({'sex': {0: 'Male', 1: 'Female'}})
```

```
In [65]: heart.head()
```

Out[65]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	Female	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	Female	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	Female	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	Female	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	Male	0	138	294	1	1	106	0	1.9	1	3	2	0

Ten slotte kan het nog zo zijn dat je data moet opdelen in groepen zodat je per groep analyses kan uitvoeren. In assignment 12 heb ik de hartpatiënten opgedeeld in leeftijdsgroepen, van 20-30, 30-40 enzovoort.

```
In [69]: age_groups = pd.cut(heart['age'], bins=[20, 30, 40, 50, 60, 70, 80],
                             labels=['20-30', '30-40', '40-50', '50-60', '60-70', '70-80'])
```

```
In [70]: age_groups
```

Out[70]:

```
0       50-60
1       50-60
2       60-70
3       60-70
4       60-70
        ...
1020    50-60
1021    50-60
1022    40-50
1023    40-50
1024    50-60
Name: age, Length: 1025, dtype: category
Categories (6, object): ['20-30' < '30-40' < '40-50' < '50-60' < '60-70' < '70-80']
```

Now, we can insert the age groups into the original dataframe.

```
In [71]: heart.insert(1, 'age group', age_groups)
```

```
In [72]: heart.head()
```

Out[72]:

	age	age group	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	50-60	Female	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	50-60	Female	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	60-70	Female	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	60-70	Female	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	60-70	Male	0	138	294	1	1	106	0	1.9	1	3	2	0

```
In [74]: heart.groupby('age group').mean()
```

Out[74]:

age group	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
20-30	29.000000	1.000000	130.000000	204.000000	0.000000	0.000000	202.000000	0.000000	0.000000	2.000000	0.000000	2.000000	1.000000
30-40	37.390625	1.328125	127.093750	211.031250	0.000000	0.796875	166.531250	0.281250	0.887500	1.593750	0.500000	2.406250	0.640625
40-50	45.028340	0.991903	124.080972	236.105263	0.080972	0.582996	157.748988	0.222672	0.652632	1.489879	0.384615	2.210526	0.672065
50-60	55.767123	0.892694	133.317352	248.815068	0.205479	0.513699	147.591324	0.404110	1.146804	1.363014	0.787671	2.363014	0.465753
60-70	64.587302	0.876984	137.634921	260.543651	0.154762	0.436508	139.142857	0.353175	1.454365	1.238095	1.107143	2.376984	0.373016
70-80	73.100000	1.000000	126.150000	243.600000	0.200000	0.650000	135.150000	0.300000	0.575000	1.650000	1.100000	2.000000	0.850000

Modeling

Dit kwam vooral in de laatste portfolio assignments voor, bij de multivariate analysis. Zo moesten we in portfolio assignment 15 een decision tree trainen om het soort penguin te kunnen voorspellen op basis van hun kenmerken. Data werd opgesplitst in twee sets, namelijk de train- en testset. Op deze manier kan op de trainset geoefend worden, waarna op de testset gekeken kan worden hoe accuraat de voorspellingen zijn.

Hetzelfde kan gedaan worden voor het voorspellen van numerical values, maar dan door middel van regression. Dit deden we in portfolio assignment 17, waar vervolgens met behulp van de RMSE (root-mean-square-error) wordt berekend wat de gemiddelde afwijking is tussen de voorspelde waarden en de daadwerkelijke waarden.

Fitting the DecisionTreeRegressor

We will be predicting the penguin's body mass based on the flipper length. In the Pearson's correlation table we saw that these two variables have a strong correlation.

In [50]:
```
from sklearn.tree import DecisionTreeRegressor

features = ['flipper_length_mm']
dt_regression = DecisionTreeRegressor(max_depth=3) # increase max_depth to see effect in the plot
dt_regression.fit(penguins_train[features], penguins_train['body_mass_g'])
```

Out[50]:
```
DecisionTreeRegressor(max_depth=3)
```

In [51]:
```
def calculate_rmse(predictions, actuals):
    if(len(predictions) != len(actuals)):
        raise Exception('The amount of predictions did not equal the amount of actuals.')

    return (((predictions - actuals) ** 2).sum() / len(actuals)) ** (1/2)
```

In [52]:
```
predictionsOnTrainSet = dt_regression.predict(penguins_train[features])
predictionsOnTestSet = dt_regression.predict(penguins_test[features])

rmseTrain = calculate_rmse(predictionsOnTrainSet, penguins_train['body_mass_g'])
rmseTest = calculate_rmse(predictionsOnTestSet, penguins_test['body_mass_g'])

print('RMSE on training set ' + str(rmseTrain))
print('RMSE on test set ' + str(rmseTest))
```

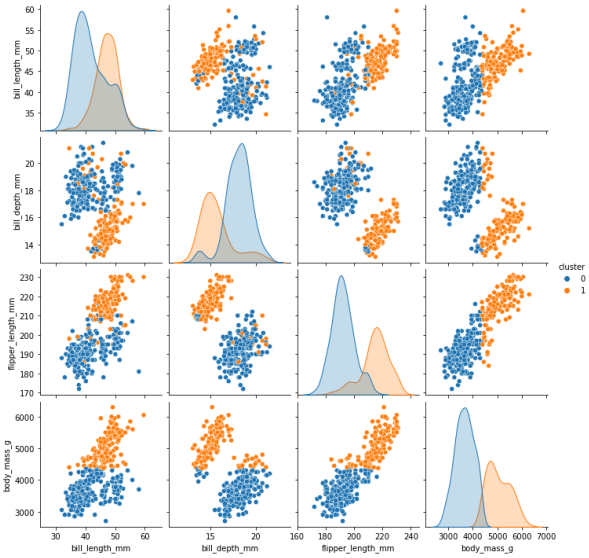
```
RMSE on training set 385.7037576357823
RMSE on test set 340.3536605027887
```

Ten slotte hebben we nog gebruik gemaakt van clustering; een machine learning methode om data op te delen in groepen (clusters) op basis van waarden. In portfolio assignment 19 hebben we dit met de penguins dataset gedaan. Het resultaat hebben we visueel gemaakt door middel van een pair plot met een hue op basis van de cluster. Door te spelen met het aantal clusters bleek dat er eigenlijk maar 2 echte groepen van elkaar te onderscheiden zijn.

Pairplot with hue

In [25]:
```
sns.pairplot(penguins, hue='cluster')
```

Out[25]:
```
<seaborn.axisgrid.PairGrid at 0x7fc164c60070>
```



Door gebruik te maken van de Silhouette Coefficient konden we een schatting maken van hoe consistent de clustervoorspelling was. Bij een aantal clusters van 2 was deze het hoogste, namelijk 0.62. Naarmate het aantal clusters hoger werd daalde de coefficient.

Silhouette Coefficient

In [33]:
```
from sklearn import metrics
from sklearn.metrics import pairwise_distances
```

In [36]:
```
# You can change the features and n_clusters in the k-means above and check the impact on the Silhouette Coefficient
metrics.silhouette_score(penguins[features], km.labels_, metric='euclidean')
```

Out[36]: 0.6270788983213472