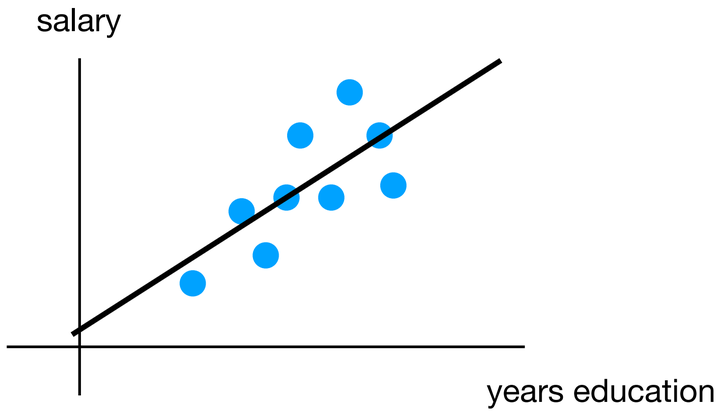
Recap förra veckan - linjär regression och scikit-learn

Years of education	Salary
5	40000
3	35000
2	30000
5	36000
4	33000
3	45000

feature label

bestämma vilka som är features och vilken som är label

Label är det vi vill predicta



Med en feature och en label —> simpel linjär regression

Men det kan utökas till n st dimensioner (antalet features) och då får vi ett n-dimensionellt hyperplan —> predicta vår label

regression i machine learning —> kontinuerlig label

Ex 32516.3214

Scikit-learn steg

0. dela upp i X och y

plocka ut features matris och label vektor

```
X, y = df.drop("sales", axis="columns"), df["sales"]
```

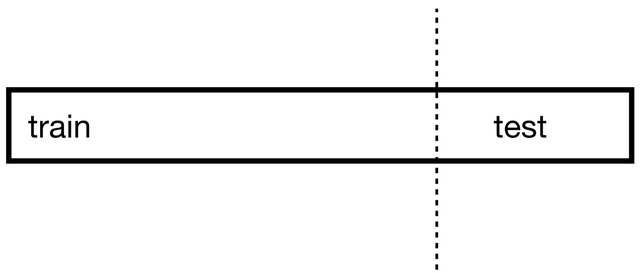
1. train|test split

analogi

train - gamla tentor

test - ny tenta som vi ska evalueras på

shuffle dataset och därefter splitta




```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42
)
```

jag brukar köra help(train_test_split) och därefter kopiera den här kodsnutten för att inte råka skriva fel på ordningen av X_train, X_test, y_train, y_test

test_size mellan 0 och 1 —> andel data till test resten till train

Ex 0.33 -> 33% test och 67% train

2. skala dataset

behåller datasetets distribution, men vi gör skalan närmare

många algoritmer kräver skalat dataset för att fungera bra

Ex. Normalization och feature standardization

```
from sklearn.preprocessing import MinMaxScaler

# instantiate a MinMaxScaler instance
scaler = MinMaxScaler()

# important note: fit on X_train and not X_test -> this avoids data leakage
scaler.fit(X_train) # use training data to fit the scaler

# transforms or scales X_train and X_test
scaled_X_train = scaler.transform(X_train)
scaled_X_test = scaler.transform(X_test)
```

plockar fram parametrar för att skala mha X_train och därefter skalar man X_train och X_test

Viktigt: gör inte fit på X_test —> läcker datan

3. träna modellen

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
# put in training data features and label
model.fit(scaled_X_train, y_train)
```

modellen lär sig parametrarna

4. predict on test data

```
y_pred = model.predict(scaled_X_test)
```

5. evaluerar

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

jämför resultaten mot vad andra modeller performat och plocka bästa