Big Data

Gaffar Sir Elkhatim

Einleitung :

In diesem Projekt werden auf verschiedene Big Data Methoden eingehen die meisten Methoden haben teilweise viel mit Künstliche Intelligenz und Stochastik zu tun die Themen und Methoden Sind folgende:

1- Multiple Linear Regression
2- Support Vector Machines
3- Entscheidungsbäume Algorithmen
4- Keras Bibliothek für Neuronale Netze
5- Random Forest
6- Blockchain
7- logistische Regression

und noch viel deswegen werden wir jetzt direkt mit der Beschreibung jede einzelne Teil anfangen

Multiple Regression:

In der Statistik ist die multiple lineare Regression, auch als multiple lineare Regression (MLR) oder lineare multiple Regression bekannt, eine Regressionsanalysemethode und ein Sonderfall der linearen Regression. Die multiple lineare Regression ist eine statistische Technik, mit der versucht wird, die beobachtete abhängige Variable anhand mehrerer unabhängiger Variablen zu erklären. Das dafür verwendete Modell hat lineare Parameter, und die abhängige Variable ist eine Funktion der unabhängigen Variablen. Diese Beziehung wird durch zusätzliche Störgrößen abgedeckt. Die multiple

lineare Regression repräsentiert die Verallgemeinerung der einfachen linearen Regression auf die Regressionszahl.

Daten die Wir gewählt haben sind Autos daten um zu vorher sage über den Preis eines Auto treffen zu können.

Die Daten sehen so aus

```
]: df = pandas.read_excel("car.xlsx")
   df.head()
```

]:

	Car	Model	kilometer	Alter	preis
0	Toyota	Aygo	1000	10	12000
1	Mitsubishi	Space Star	0	0	30000
2	Skoda	Citigo	195000	15	1000
3	Fiat	500	20000	5	7000
4	Mini	Cooper	0	1	100000

um dann mit Hilfe von Multiple Regression konnten wir vorhersagen treffen werden wie auf das Bild zu sehen ist

```
print("Das Auto kostet ",int(predictedCO2) ,"Euro")
```

```
Das Auto kostet  51753 Euro
```

Und die Predicted variable ist hier

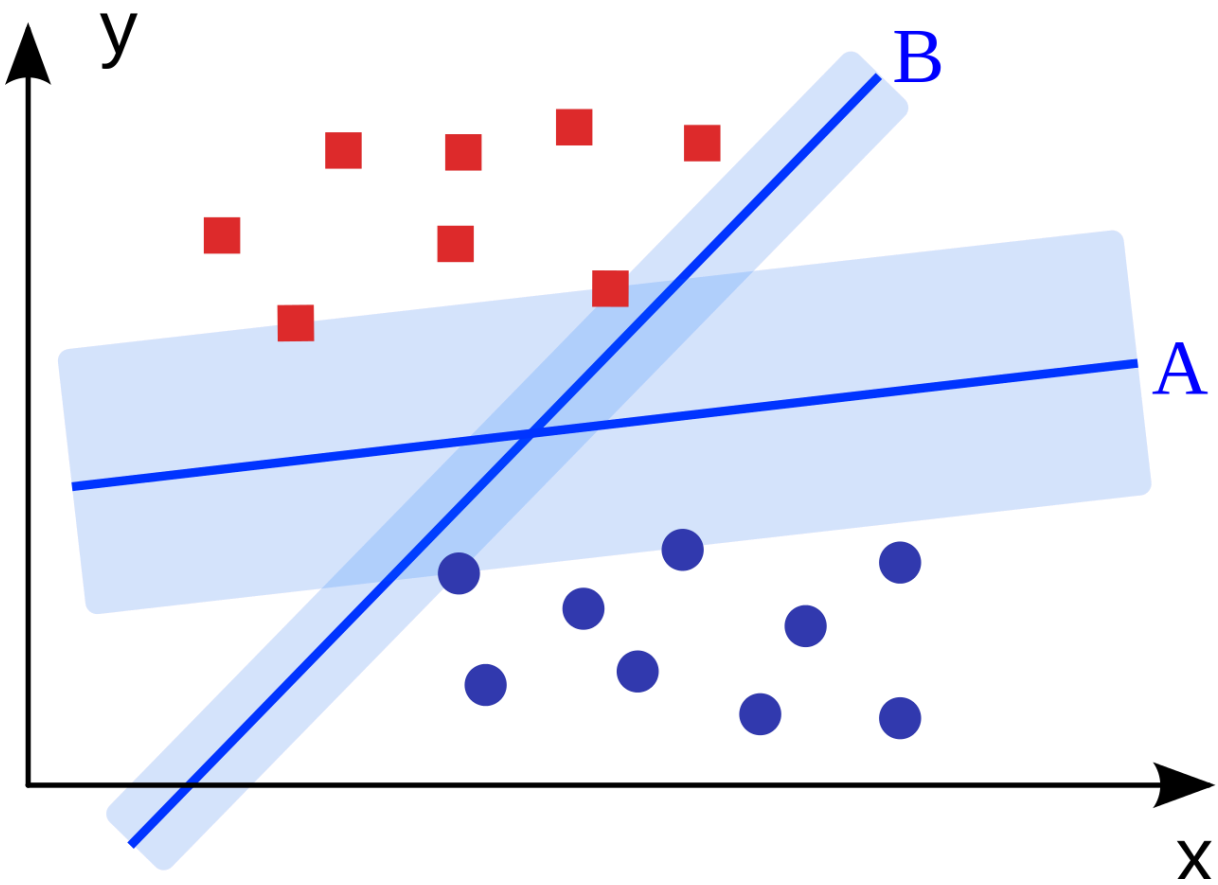
```
predictedCO2 = regr.predict([[2300, 1]])
```

Support Vector Machines

Support-Vektor-Maschinen sind keine Maschinen im herkömmlichen Sinne, dh sie enthalten keine materiellen Komponenten. Es ist eine rein mathematische Mustererkennungsmethode, die in einem Computerprogramm

implementiert ist. Daher geben die Namen einiger Maschinen keine Maschinen an, sondern unterstützen Vektormaschinen, den Ursprungsbereich des maschinellen Lernens.

Ein Beispiel zeigt wann SVM nützlich ist



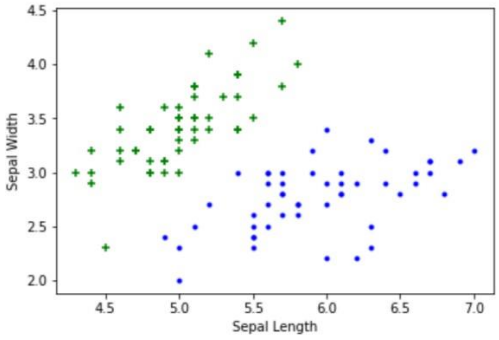
In diesem Beispiel hat zwei trenngerad Lienen um zu wissen welche passt besser für unseres Model können wir die SVM benutzen.

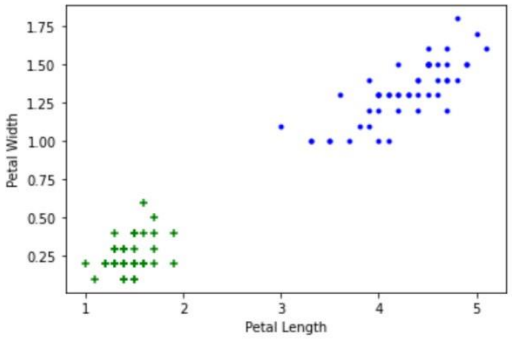
Wir haben in der Aufgabe die Iris Dataset von sklearn ,die Dataset gibt uns über verschiedene Arten von Blumen

Die Datenset sieht so aus

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

Wir haben die Daten Visualisiert





Und danach haben unsere Model Trainiert mit fit() und dann die Vorhersagen getroffen und war immer richtig

```
model.predict([[4.8,3.0,1.5,0.3]])
```

```
array([0])
```

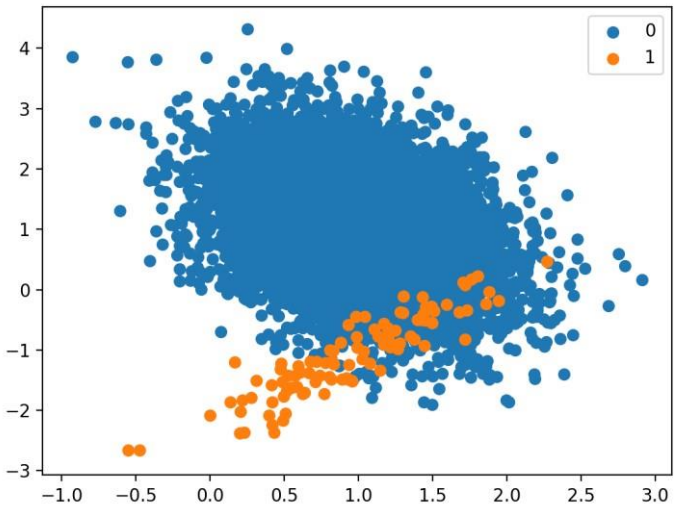
Die werte die eingegeben haben passen zum ersten Blumen in der Tabelle

Entscheidungsbäume Algorithmen:

Sind geordneter gerichteter Baum, der zur Darstellung von Entscheidungsregeln verwendet wird. Die grafische

Darstellung als Baumdiagramm veranschaulicht die hierarchische kontinuierliche Entscheidungsfindung. Sie sind in vielen Bereichen wichtig, in denen eine automatische

Klassifizierung erfolgt oder formale Regeln aus empirischem Wissen abgeleitet oder vorgeschlagen werden.



	Unterhnehmn	job	gerad	gehalt
0	SAP	Manger	bachelor	1
1	SAP	Markiting	master	1
2	SAP	Informatiker	master	1
3	SAP	Manger	master	1
4	SAP	Markiting	bachelor	0
5	SAP	Informatiker	bachelor	0

Hier in dieser Daten geht es Mitarbeitern die Bei verschiedenen Konzernen arbeiten und entweder mehr als 100.000 im Jahr verdienen oder weniger und wir haben die alles mit zahlen ersetzt damit wir das Model trainieren können

Auf das Foto ist der Neuer Format dargestellt

Unterhnehmn	job	gerad	company_n	job_n	degree_n
SAP	Manger	bachelor	1	1	0
SAP	Markiting	master	1	2	1
SAP	Informatiker	master	1	0	1
SAP	Manger	master	1	1	1
SAP	Markiting	bachelor	1	2	0
SAP	Informatiker	bachelor	1	0	0
Telekom	Informatiker	bachelor	2	0	0
Telekom	Manger	bachelor	2	1	0
Telekom	Markiting	bachelor	2	2	0
Telekom	Informatiker	master	2	0	1
Telekom	Markiting	master	2	2	1

Hier ist unser vorhersage zu sehen

```
#SAP manger beachelor

model.predict([[1,1,0]])
```

Mit dem ergebenes 1

Keras Bibliothek für Neuronale Netze:

Keras stellt schnelle Implementierungen neuronaler Netze für tiefes Lernen zur Verfügung. Dies ist eine in Python geschriebene Open Source-Bibliothek, die in Verbindung mit Frameworks wie TensorFlow oder Theano verwendet werden kann.

hier haben wir die Daten von tensorflow und keras die mnist Dataset

hier sind unsere Layers die wir für das Model benutzt

```
model.add(tf.keras.layers.Flatten())
```

```
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(10, activation=tf.nn.softmax))
```

Und so haben wir es kompiliert mit dem Optimizer adam

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

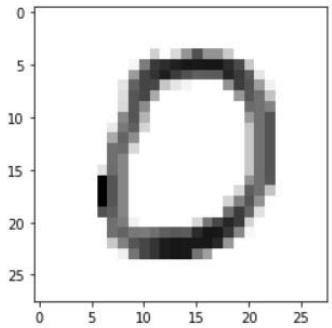
Und am Ende haben wir getestet ob wir das erkennen können

Und so war das ergebenes

```
import numpy as np

print(np.argmax(predictions[10]))
```

```
0
```

```
import matplotlib.pyplot as plt

plt.imshow(x_test[10],cmap=plt.cm.binary)
plt.show()
```



logistische Regression:

Die logistische Regression oder das Logit-Modell sollte als Regressionsanalyse verstanden werden, die die diskrete Verteilung der abhängigen Variablen modelliert (meistens Modellierung). Wenn die logistische Regression nicht weiter als polynomielle oder geordnete logistische Regression charakterisiert wird, bedeutet dies normalerweise eine binomiale logistische Regression der dichotomen (binären) abhängigen Variablen. Unabhängige Variablen können eine

beliebige Skalierungsstufe haben, und diskrete Variablen mit mehr als zwei Merkmalen können in eine Reihe von binären Pseudovariablen zerlegt werden.

Beispiel wo man das nutzen kann:

1- Conversion-Prognose: Kauft ein Kunde ein Produkt?
2- Bonität: Zahlt ein Kreditnehmer einen Kredit vollständig zurück?

3- Markenbekanntheit: Kennt jemand eine Marke?

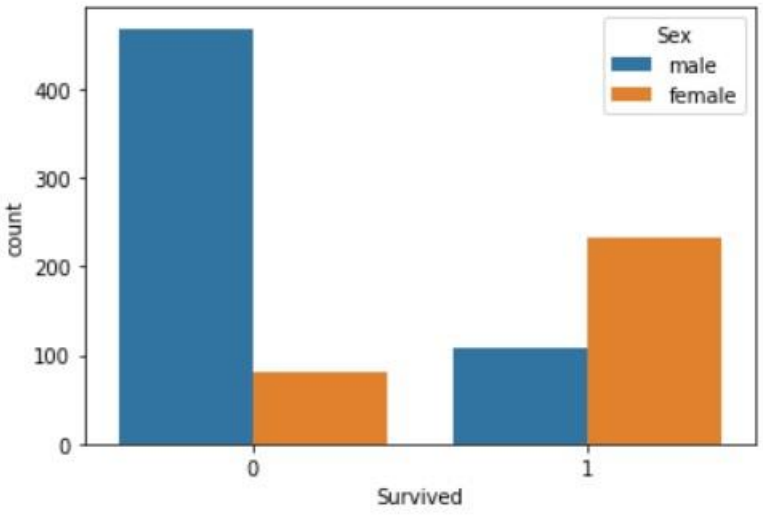
Hier haben wir eine Dataset von den Überlebenden von Titanic

```
train = pd.read_csv("train.csv")
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Das ist die Visualisierung von den Überlebenden nach Geschlecht

```
<AxesSubplot:xlabel= Survived , ylabel= count >
```



wir haben hier den Bericht von der Kalzifizierung

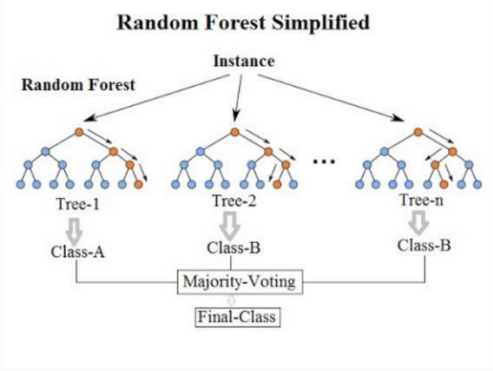
```python
predictions = logmodel.predict(X_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.75      0.55      0.63        22
           1       0.74      0.88      0.81        33

    accuracy                           0.75        55
   macro avg       0.75      0.71      0.72        55
weighted avg       0.75      0.75      0.74        55
```

Random Forest:

Random Forest ist eine Klassifizierungs- und Regressionsmethode, die aus mehreren nicht zusammenhängenden Entscheidungsbäumen besteht. Während des Lernprozesses wachsen alle Entscheidungsbäume unter einer gewissen Randomisierung.

Um das zu veranschaulichen habe das nächste Foto

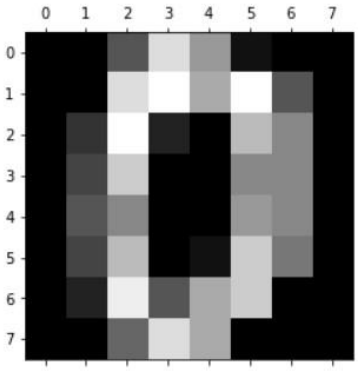


Hier haben wir die Datenset von sklrean die MINST

```
: import pandas as pd
  from sklearn.datasets import load_digits
  digits = load_digits()
```

```
: %matplotlib inline
  import matplotlib.pyplot as plt
```

```
: for i in range(5):
      plt.matshow(digits.images[i])
```



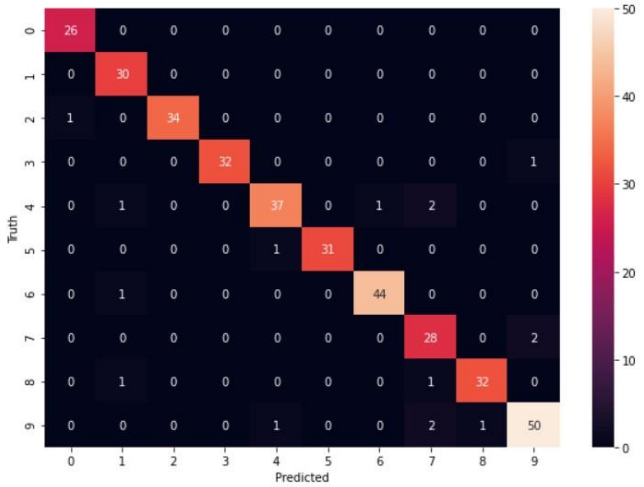
Hier haben wir unsere Classifier für die Zahlen

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
```

Und dann haben wir eine Confussion Matrix um zu schauen wie gut das Modell

```
plt.ylabel('Truth')
```

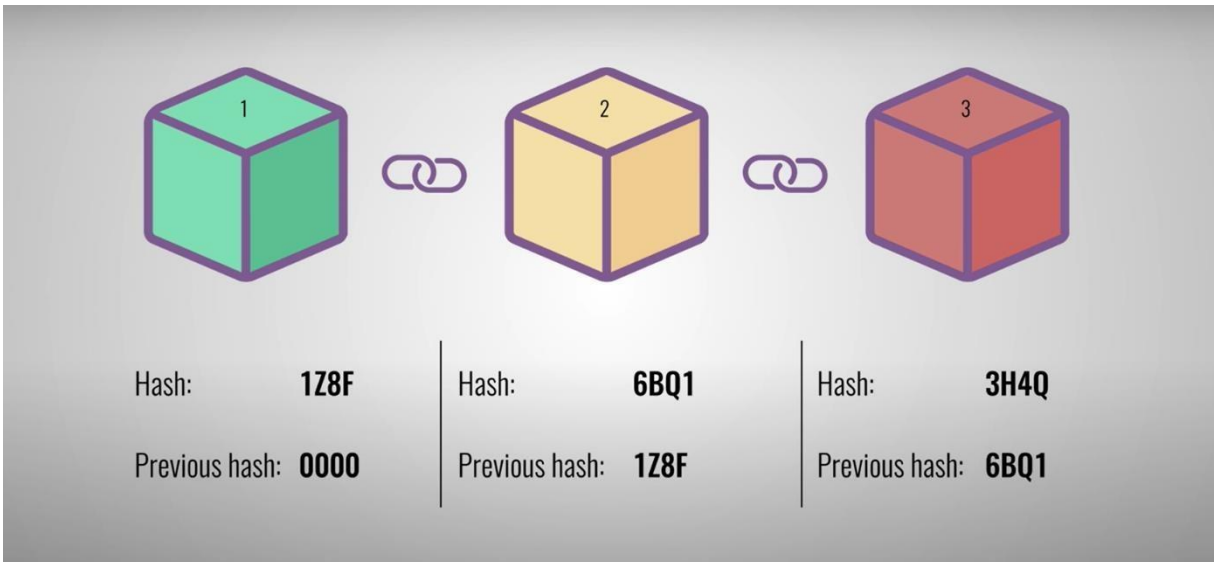
```
Text(69.0, 0.5, 'Truth')
```



Blockchain:

Es handelt sich um eine fortlaufende und erweiterbare Liste von Datensätzen, sogenannten "Blöcken", die mithilfe eines Verschlüsselungsprozesses miteinander verknüpft werden
Jeder Block enthält normalerweise den kryptografisch sicheren Hash des vorherigen Blocks, Zeitstempel und Transaktionen

Auf das Bild ist zu sehen wie ein Block funktioniert



Mit unserem Programm haben wir es geschafft Random Hash zu erstellen und das ganze beim nächsten block zu speichern mit den Hash vom dem neuen Block Außerdem auch die Daten.