PROJEKTARBEIT KURS „BIG DATA ANALYSIS“

DANIEL, OSAMA, SINA

PROJEKT ONLINE-MARKETING

INHALT

1. Ausgangslage
2. Vorgehensweise
3. Datenanalyse
4. Ergebnisse und Lessons Learnt
5. Empfehlungen

AUSGANGSLAGE

- Schalten von Bannerwerbung auf Websites
- Einschätzung dieser Werbemaßnahmen in Hinblick auf Erfolg
- Erfolg = Websitenutzer hat auf Ad geklickt
- Analyse der Nutzereigenschaften sowie Untersuchung der relevanten Nutzereigenschaften auf den Erfolg
- Handlungsempfehlung zur Erfolgssteigerung

VORGEHENSWEISE

1. Pre-Processing der Daten
2. Diskussion in der Gruppe hinsichtlich möglicher Probleme und Arbeitsteilung
3. Datenexploration
4. Vorbereitung der Daten für Regressionsanalyse und Durchführung
5. Datasplit nach Test und Trainingsdaten
6. Klassifikation und Cluster-Analysen
7. Diskussion und Verdichtung der Ergebnisse
8. Ableiten von Handlungsempfehlung

ÜBERBLICK DER DATEN

```
pd.read_csv("./Advertising.csv")
```

```
pd.read_csv("./Advertising.csv").info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Daily Time Spent on Site  1000 non-null   float64
 1   Age                       1000 non-null   int64  
 2   Area Income               1000 non-null   float64
 3   Daily Internet Usage      1000 non-null   float64
 4   Ad Topic Line             1000 non-null   object 
 5   City                      1000 non-null   object 
 6   Male                      1000 non-null   int64  
 7   Country                   1000 non-null   object 
 8   Timestamp                 1000 non-null   object 
 9   Clicked on Ad             1000 non-null   int64  
dtypes: float64(3), int64(3), object(4)
memory usage: 62.6+ KB
```

```
pd.read_csv("./Advertising.csv").isna().sum()
```

```
Daily Time Spent on Site    0
Age                         0
Area Income                 0
Daily Internet Usage        0
Ad Topic Line               0
City                        0
Male                        0
Country                     0
Timestamp                   0
Clicked on Ad               0
dtype: int64
```

- 1.000 Datensätze, 10 Variablen
- Keine Missing Values
- „Male“ und „Click on Ad“ quasi metrisch (0/1)
- „Timestamp“ mit Datum und Uhrzeit als Object
- „Daily Time Spent on Site“, „Age“, „Area Income“, „Daily Internet Usage“ metrische Daten
- „City“, „Country“, „Ad Topic Line“ ist der Datatype Object
- Geschlecht 50/50%
- Alter nicht gleich verteilt, zu viele junge Nutzer

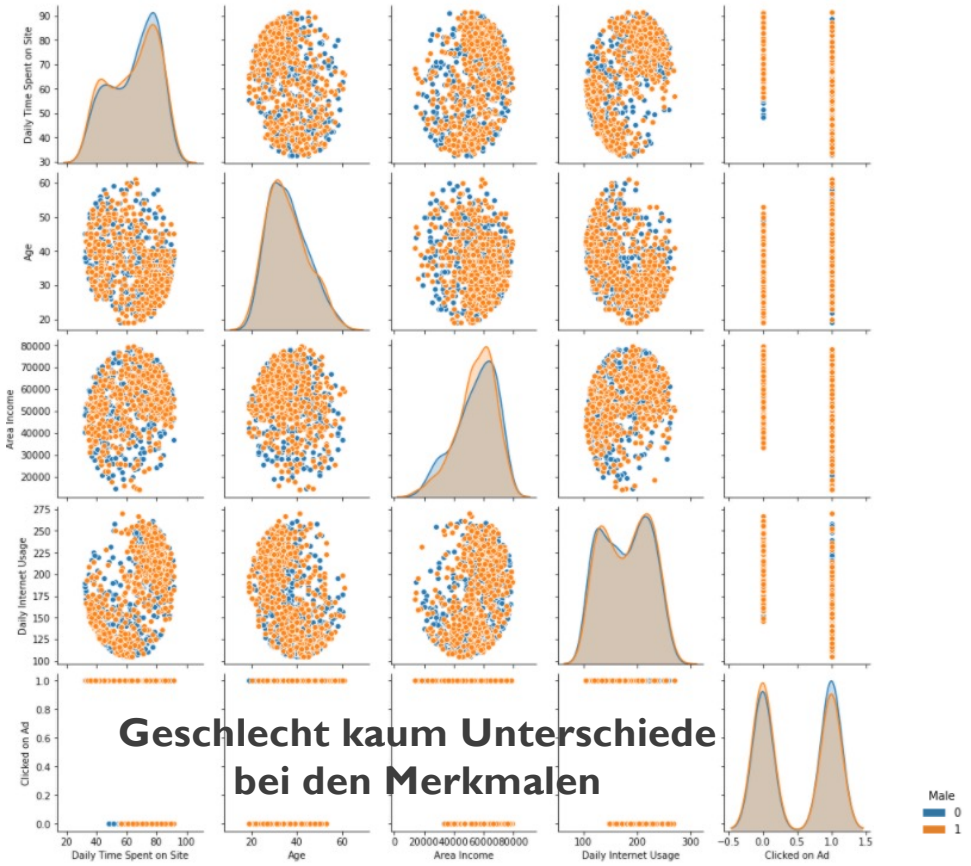
DESKRIPTIVE DATEN ANALYSE

```
df.describe(include='all',datetime_is_numeric=True)
```

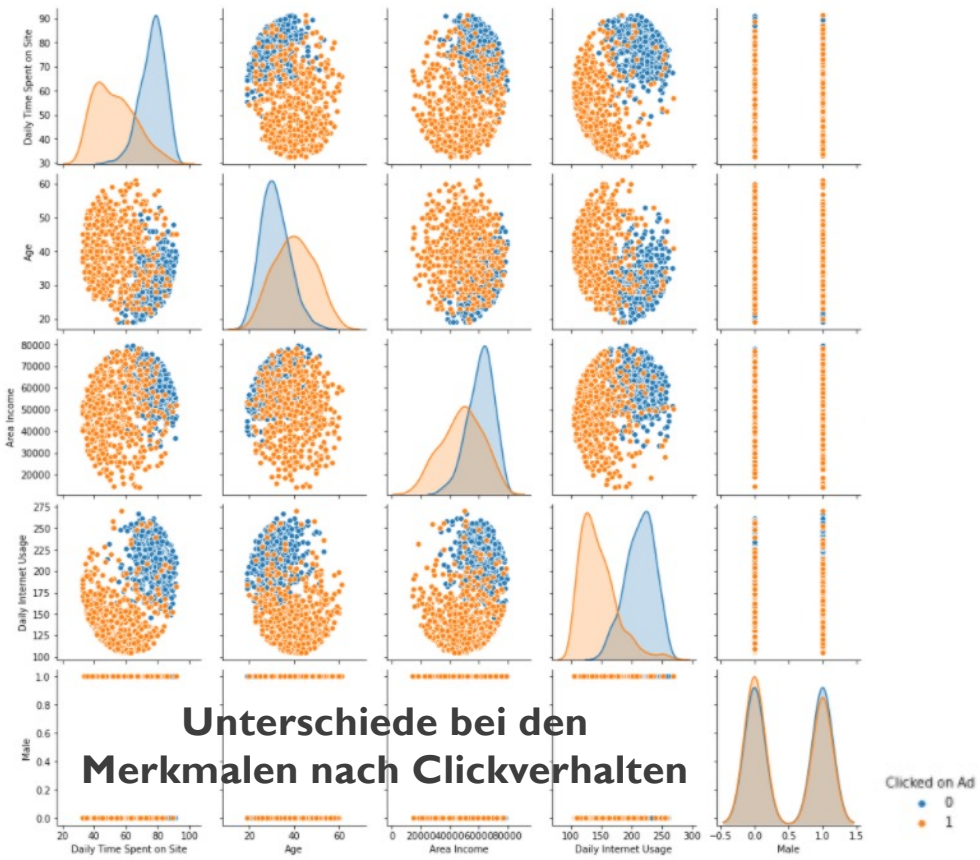
	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad	Weekday	SessionTime	Continent
count	1000.000000	1000.000000	1000.000000	1000.000000	1000	1000	1000.000000	1000	1000	1000.00000	1000	1000	1000
unique	NaN	NaN	NaN	NaN	1000	969	NaN	237	NaN	NaN	7	4	8
top	NaN	NaN	NaN	NaN	Customizable systematic service-desk	Lisamouth	NaN	Czech Republic	NaN	NaN	Sunday	Morning	Asia
freq	NaN	NaN	NaN	NaN	1	3	NaN	9	NaN	NaN	159	294	211
mean	65.000200	36.009000	55000.000080	180.000100	NaN	NaN	0.481000	NaN	2016-04-10 10:34:06.636000	0.50000	NaN	NaN	NaN
min	32.600000	19.000000	13996.500000	104.780000	NaN	NaN	0.000000	NaN	2016-01-01 02:52:10	0.00000	NaN	NaN	NaN
25%	51.360000	29.000000	47031.802500	138.830000	NaN	NaN	0.000000	NaN	2016-02-18 02:55:42	0.00000	NaN	NaN	NaN
50%	68.215000	35.000000	57012.300000	183.130000	NaN	NaN	0.000000	NaN	2016-04-07 17:27:29.500000	0.50000	NaN	NaN	NaN
75%	78.547500	42.000000	65470.635000	218.792500	NaN	NaN	1.000000	NaN	2016-05-31 03:18:14	1.00000	NaN	NaN	NaN
max	91.430000	61.000000	79484.800000	269.960000	NaN	NaN	1.000000	NaN	2016-07-24 00:22:16	1.00000	NaN	NaN	NaN
std	15.853615	8.785562	13414.634022	43.902339	NaN	NaN	0.499889	NaN	NaN	0.50025	NaN	NaN	NaN

PAIRPLOT NACH GESCHLECHT UND CLICKRATE

```
sns.pairplot(df,hue='Clicked on Ad');
```



Geschlecht kaum Unterschiede bei den Merkmalen



Unterschiede bei den Merkmalen nach Clickverhalten

KORRELATIONSANALYSE

```
df_corr = df.drop(['Weekday','Continent','Timestamp','Country','City','Ad Topic Line'],axis=1)
```

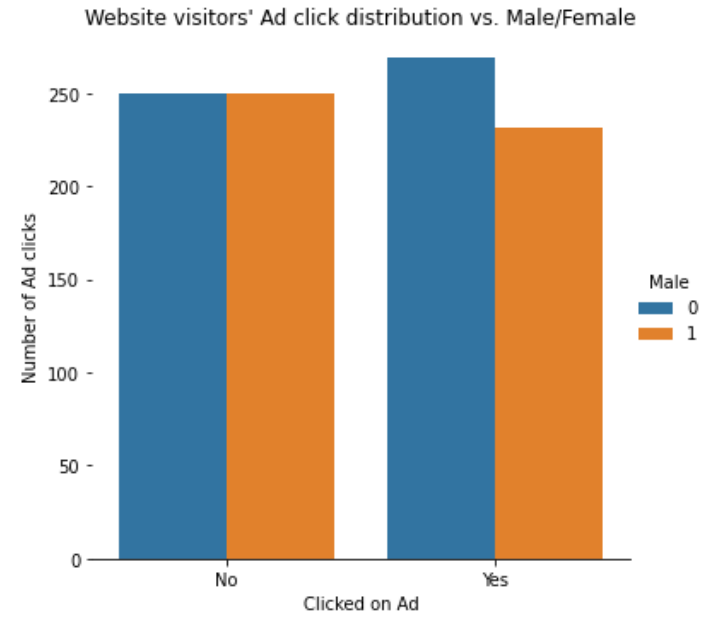
```
colormap = plt.cm.viridis
plt.figure(figsize=(12,12))
sns.heatmap(df_corr.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True, cmap=colormap, linecolor='white', annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xc031598>
```

- Starke Korrelationen zwischen
  - „Daily Time Spent on Site“ & „Click on Ad“ (negativ)
  - „Daily Internet Usage“ & „Click on Ad“ (negativ)
  - „Daily Internet Usage“ & „Daily Time Spent on Site“ (positiv)
- Mittlere Korrelationen
  - „Area Income“ & „Click on Ad“ (negativ)
  - „Age“ & „Click on Ad“ (positiv)
  - „Age“ & „ Area Income“ (positiv)

EXPLORATIVE DATENANALYSE – NACH GESCHLECHT

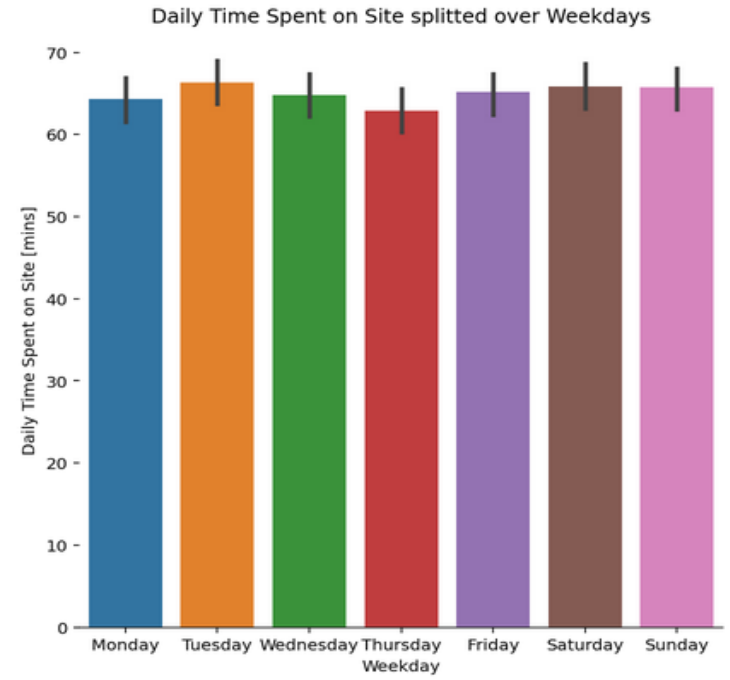
```
g = sns.catplot(x="Clicked on Ad", hue="Male",data=df, kind="count");

(g.set_axis_labels("Clicked on Ad", "Number of Ad clicks")
  .set_xticklabels(["No", "Yes"])
  .set(title="Website visitors' Ad click distribution vs. Male/Female")
  .despine(left=True));
```



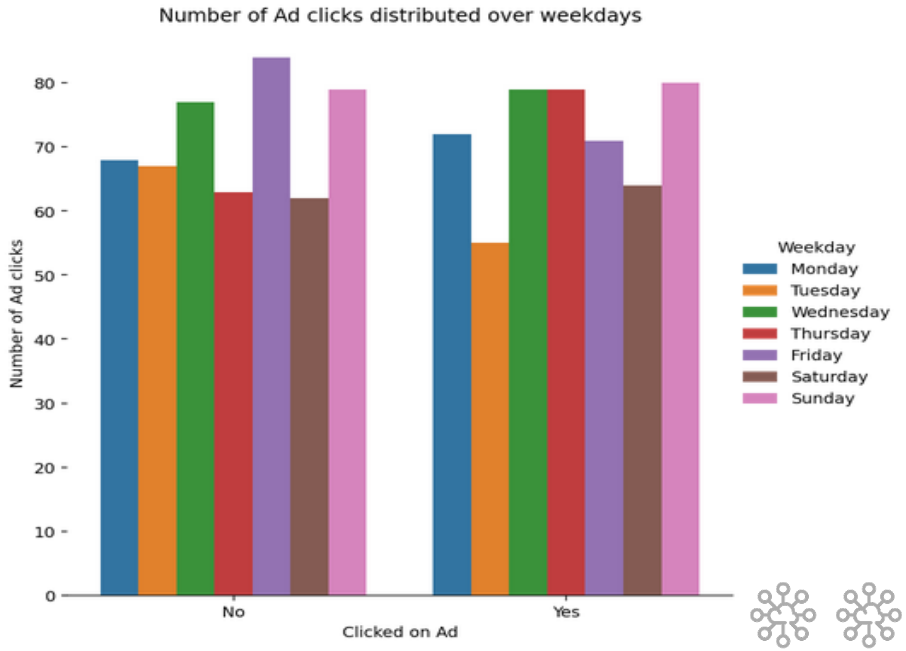
- Frauen haben tendenziell etwas mehr Interesse an Werbebannern

EXPLORATIVE DATENANALYSE – NACH WOCHENTAGE

```
g = sns.catplot(x="Weekday",y="Daily Time Spent on Site",
                order=['Monday','Tuesday','Wednesday',
                       'Thursday','Friday','Saturday','Sunday'],
                    data=df, kind="bar",height=6.0);
(g.set_axis_labels("Weekday", "Daily Time Spent on Site [mins]")
 .set(title="Daily Time Spent on Site splitted over Weekdays")
 .despine(left=True));
```



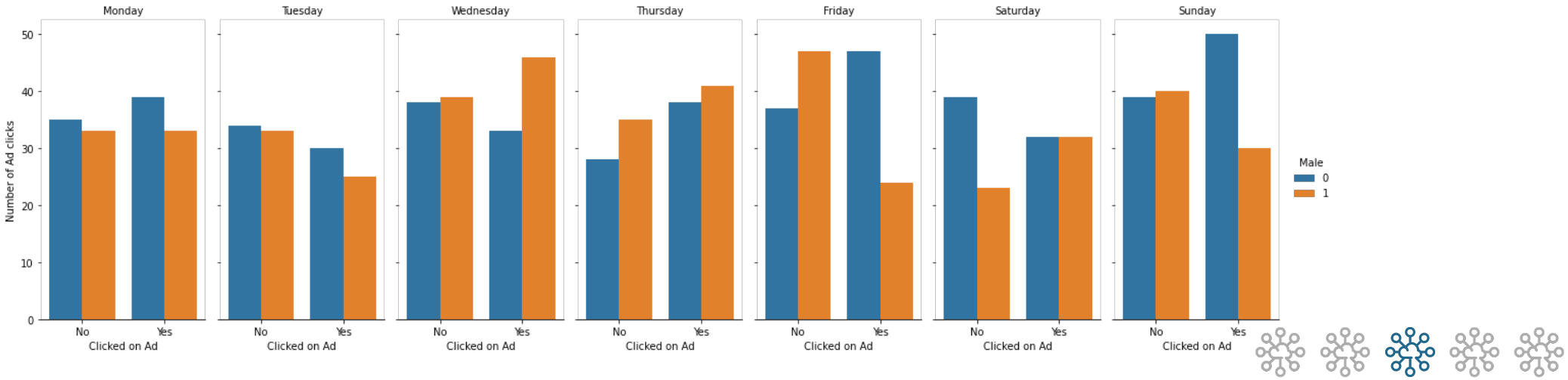

```
g = sns.catplot(x="Clicked on Ad", hue="Weekday",
                hue_order=['Monday','Tuesday','Wednesday',
                           'Thursday','Friday','Saturday','Sunday'],
                    data=df, kind="count",height=6.0);
(g.set_axis_labels("Clicked on Ad", "Number of Ad clicks")
  .set_xticklabels(["No", "Yes"])
  .set(title="Number of Ad clicks distributed over weekdays")
  .despine(left=True));
```



EXPLORATIVE DATENANALYSE – NACH GESCHLECHT UND TAGE

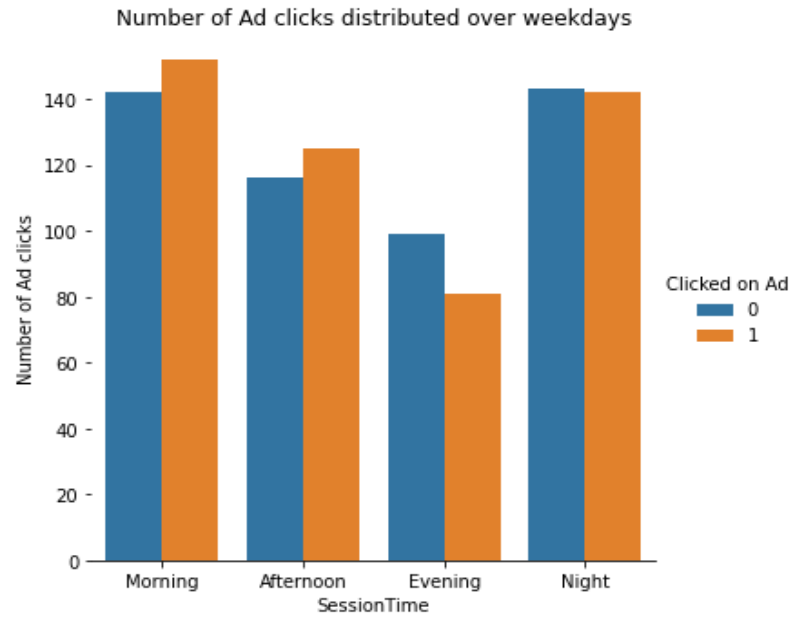
```
g = sns.catplot(x="Clicked on Ad", hue="Male", col="Weekday", col_wrap=7,
            col_order=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday'],
                data=df, kind="count",aspect=0.5);

(g.set_axis_labels("Clicked on Ad", "Number of Ad clicks")
  .set_xticklabels(["No", "Yes"])
  .set_titles("{col_name}")
  .despine(left=True));
```

- Mal klicken Männer, mal Frauen mehr, in Summe ähnliches Verhalten
- Frauen deutlich stärker Interessiert als Männer am Fr. und So.



EXPLORATIVE DATENANALYSE – NACH TAGESZEIT

```
g = sns.catplot(x="SessionTime", hue="Clicked on Ad",
            order=['Morning','Afternoon','Evening','Night'],
                data=df, kind="count");

(g.set_axis_labels("SessionTime", "Number of Ad clicks")
  .set(title="Number of Ad clicks distributed over weekdays")
  .despine(left=True));
```



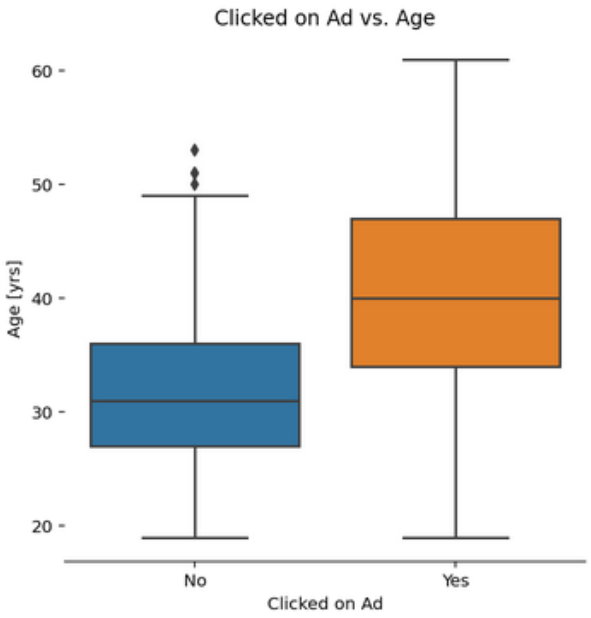
- Meisten Besucher auf Website werden Morgens oder in der Nacht verzeichnet
- Klickrate ist morgens und nachmittags etwas höher in Relation zu den Besuchern

```
def get_part_of_day(hour):
    return (
        "morning" if 6 <= hour <= 12
        else
        "afternoon" if 12 < hour <= 18
        else
        "evening" if 18 < hour <= 22
        else
        "night"
    )
```

```
df['SessionTime'] = df['hour'].apply(get_part_of_day).astype('category')
```

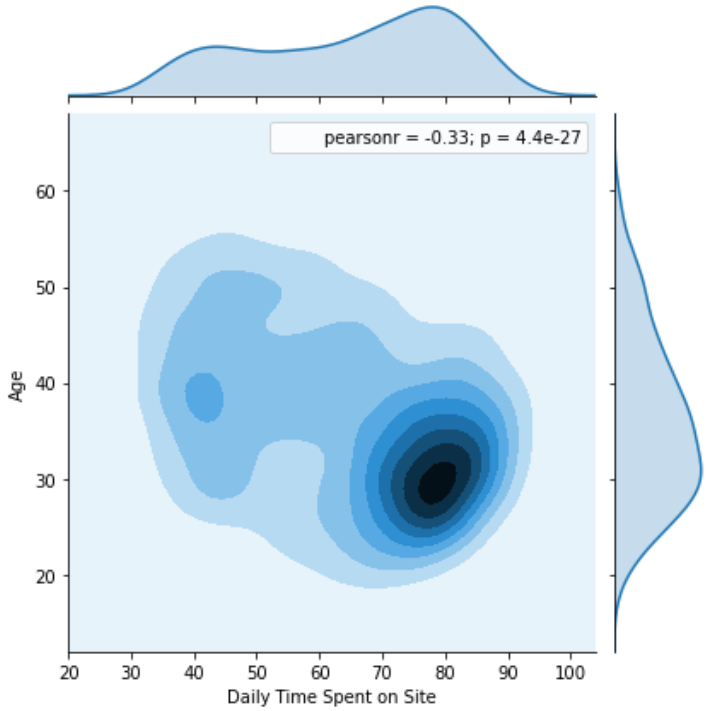
EXPLORATIVE DATENANALYSE – NACH ALTER

```
g = sns.catplot(x="Clicked on Ad",y="Age",
                data=df, kind="box");
(g.set_axis_labels("Clicked on Ad", "Age [yrs]")
  .set_xticklabels(["No", "Yes"])
  .set(title="Clicked on Ad vs. Age")
  .despine(left=True));
```



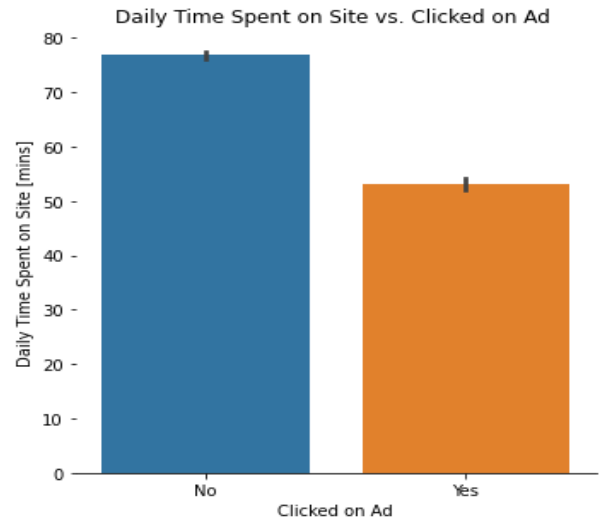
- Jüngere Nutzer verbringen mehr Zeit auf der Website
- Jüngere klicken seltener auf die Werbebanner als ältere Nutzer

```
sns.jointplot(x="Daily Time Spent on Site",y="Age",
              data=df,kind="kde", stat_func= stats.pearsonr);
```

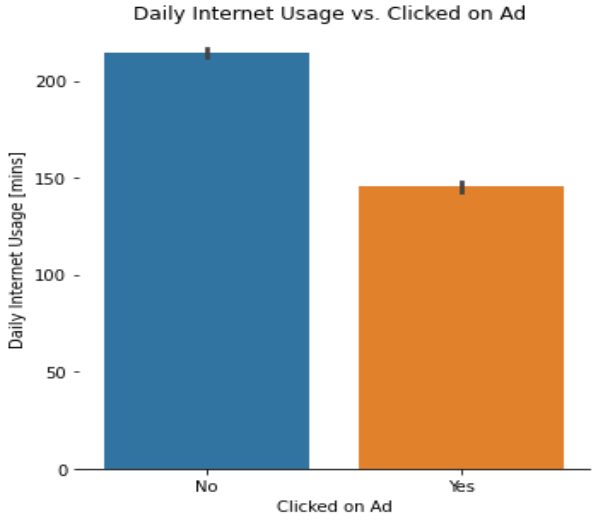


EXPLORATIVE DATENANALYSE – NACH INTERNETNUTZUNG

```
g = sns.catplot(x="Clicked on Ad", y="Daily Time Spent on Site",
                data=df, kind="bar");

(g.set_axis_labels("Clicked on Ad", "Daily Time Spent on Site [mins]")
  .set_xticklabels(["No", "Yes"])
  .set(title="Daily Time Spent on Site vs. Clicked on Ad ")
  .despine(left=True));
```



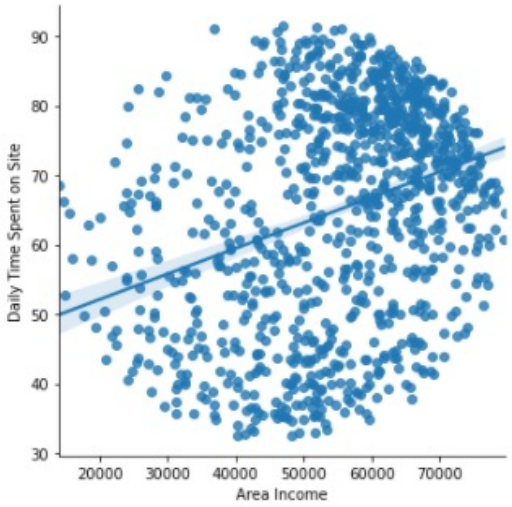

```
g = sns.catplot(x="Clicked on Ad", y="Daily Internet Usage",
                data=df, kind="bar");

(g.set_axis_labels("Clicked on Ad", "Daily Internet Usage [mins]")
  .set_xticklabels(["No", "Yes"])
  .set(title="Daily Internet Usage vs. Clicked on Ad ")
  .despine(left=True));
```



- Stärkere Internetnutzung führt nicht zu einer höheren Klickrate

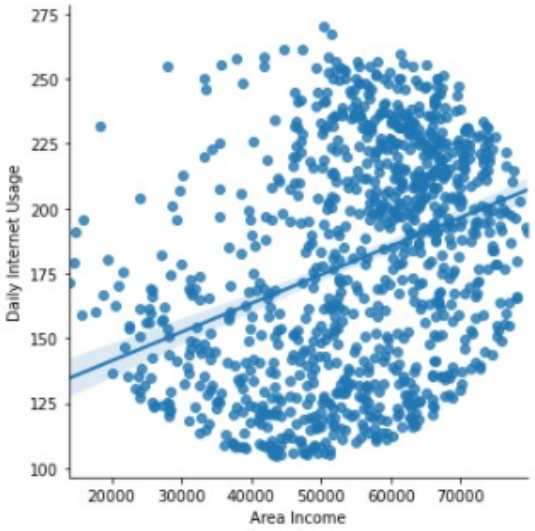
EXPLORATIVE DATENANALYSE – NACH EINKOMMEN

```
sns.lmplot("Area Income","Daily Time Spent on Site", data=df);
```

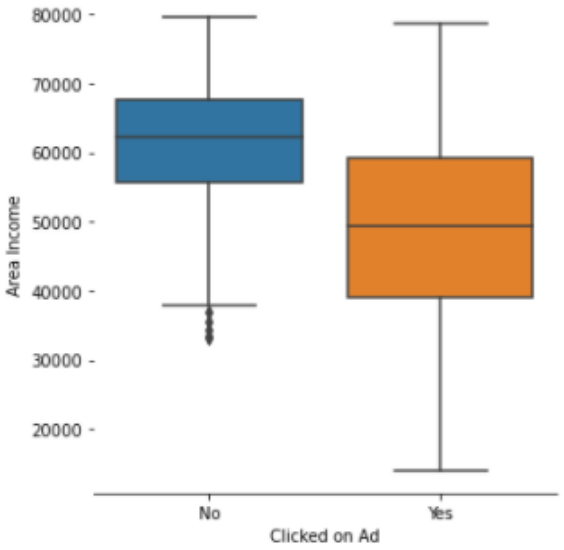


- Leichte Tendenz, dass Personen mit mehr Einkommen mehr Zeit im Internet sowie auf der relevanten Website verbringen
- (r = 0,34 / 0,31)

```
sns.lmplot("Area Income","Daily Internet Usage", data=df);
```




```
g = sns.catplot(x="Clicked on Ad",y="Area Income",data=df, kind="box");
(g.set_axis_labels("Clicked on Ad", "Area Income")
  .set_xticklabels(["No", "Yes"])
  .despine(left=True));
```



- Personen mit höheren Einkommen, klicken seltener auf die Werbebanner

EXPLORATIVE DATENANALYSE – NACH LÄNDERN

```
#installation
!pip install pycountry-convert
```

```
#function to convert to alpah2 country codes and continents
from pycountry_convert import convert_continent_code_to_continent_name, country_name_to_country_alpha2

def get_continent(country_name):
    try:
        country_alpha2 = country_name_to_country_alpha2(country_name)
    except:
        country_alpha2 = 'Unknown'
    try:
        country_continent_code = country_alpha2_to_continent_code(country_alpha2)
    except:
        cn_continent = 'Unknown'
    try:
        country_continent_name = convert_continent_code_to_continent_name(country_continent_code)
    except:
        country_continent_name = 'Unknown'
    return country_continent_name
```

```
df['Continent'] = df['Country'].apply(lambda x: get_continent(x))
```

```
df['Country'][df['Continent']=='Unknown'].count()
```

```
74
```

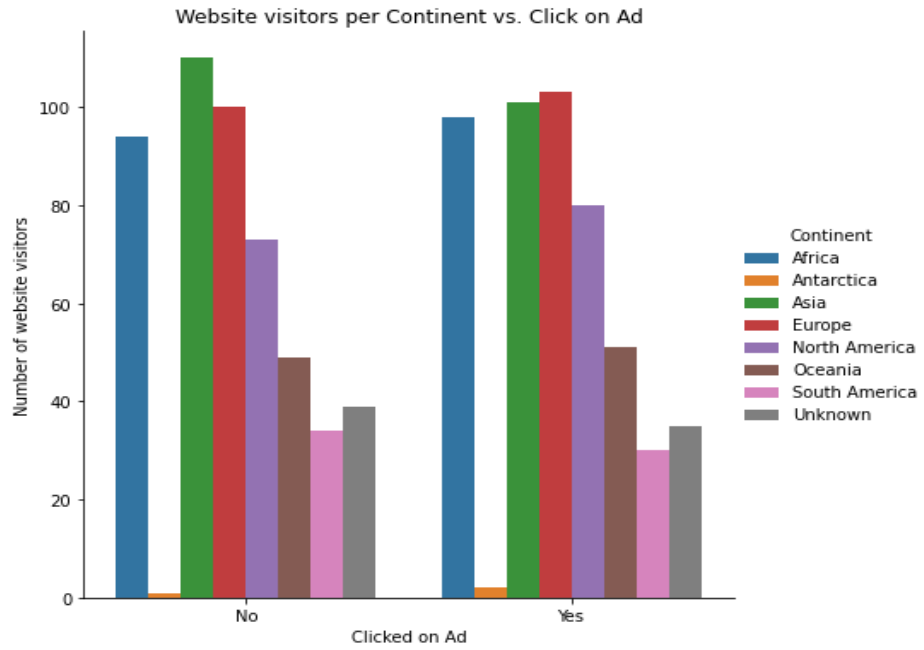
- Länder zu viele und zu geringe Datenmenge
- Clustering in Kontinente
- 74 Einträge konnten nicht zugeordnet werden

```
df.Country.value_counts()
```

```
Czech Republic                                  9
France                                          9
Greece                                          8
Australia                                       8
Senegal                                         8
                                               ..
Mozambique                                      1
Aruba                                           1
British Indian Ocean Territory (Chagos Archipelago)    1
Cape Verde                                      1
Slovenia                                        1
Name: Country, Length: 237, dtype: int64
```

EXPLORATIVE DATENANALYSE – NACH CONTINENT

```
g = sns.catplot(x="Clicked on Ad",hue="Continent",data=df, kind="count", height = 6);
(g.set(title="Website visitors per Continent vs. Click on Ad")
 .set_axis_labels("Clicked on Ad", "Number of website visitors")
 .set_xticklabels(["No", "Yes"])
);
```



- Die Meisten Besucher sind aus Asia und Europe, gefolgt von Africa
- Verhältnis geklickt und nicht-geklickt über alle Kontinente ca. 50/50

REGRESSIONSANALYSE

```
X = df[['Daily Time Spent on Site', 'Age', 'Area Income']].values
y = df['Clicked on Ad'].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)

from sklearn.linear_model import LogisticRegression as logit

logitmodel = logit()

logitmodel.fit(X_train, y_train)

print(logitmodel.score(X_test, y_test))

print(logitmodel.coef_)
```

```
0.884
[[-1.07809397e-01  2.51432010e-01 -3.30540722e-05]]
```

```
X = df[[ 'Age', 'Area Income', 'Daily Internet Usage']].values
y = df['Clicked on Ad'].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)

from sklearn.linear_model import LogisticRegression as logit

logitmodel = logit()

logitmodel.fit(X_train, y_train)

print(logitmodel.score(X_test, y_test))

print(logitmodel.coef_)
print(logitmodel.intercept_)
```

```
0.888
[[ 2.38091096e-01 -2.92624159e-05 -3.79511144e-02]]
[0.00383651]
```

- 'Daily Time Spent on Site' und 'Daily Internet Usage' korreliert stark, daher vergleichen wir zwei Modelle. Der Score unterscheidet sich um 0,004.
- Für weitere Berechnungen wurde die Variable 'Daily Time Spent on Site' verwendet, da diese als relevanter erachtet wird.
- ‚Male' wurde hier nicht verwendet, da es keine Erklärung für die Variable ‚Click on Ad' gibt.

TRAININGSMODEL

```
X = df[['Daily Time Spent on Site', 'Age', 'Area Income']].values
y = df['Clicked on Ad'].values
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)
```

```
#### Standarizierun (Pre-Procesing)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train_std = scaler.transform(X_train)
X_test_std = scaler.transform(X_test)
```

```
from sklearn.linear_model import LogisticRegression as logit

logitmodel = logit()
```

```
logitmodel.fit(X_train_std, y_train)
```

```
LogisticRegression()
```

```
logitmodel.score(X_test_std, y_test)
```

```
0.912
```

```
logitmodel.coef_
```

```
array([[-3.1291158 ,  1.35783719, -1.3399501 ]])
```

```
logitmodel.intercept_
```

```
array([0.91426185])
```

- Score mit nicht-standardisierten Daten liegt bei 0,88
- Daher wurden Daten standardisiert

VORHERSAGE – KLASSIFIKATIONS-REPORT

```
from sklearn.metrics import classification_report

print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.89	0.93	0.91	117
1	0.94	0.89	0.92	133
accuracy			0.91	250
macro avg	0.91	0.91	0.91	250
weighted avg	0.91	0.91	0.91	250

Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+TP)
Precision	= TP/(FP+TP)
Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)

- Im Durchschnitt können wir 91 Prozent der Beobachtungen korrekt vorhersagen (sensitivity/recall/TPR).
- Die Präzision unseres Modells beträgt 94 Prozent für die korrekte Vorhersage eines Klicks auf die Werbeanzeige (precision/PPV)
- Das Modell hat eine Genauigkeit von 91% bei der Vorhersage des korrekten Status von Website-Besuchern (clicked/not_clicked)

KLASSIFIKATIONSANALYSE

'Age' und 'Area Income'

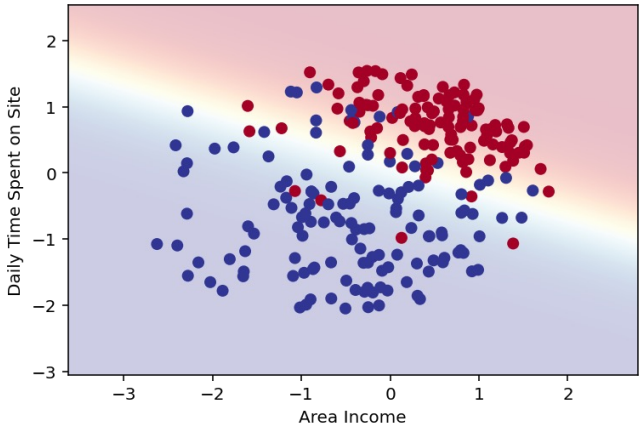
```
X = df[['Age', 'Area Income']].values
y = df['Clicked on Ad'].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)

from sklearn.linear_model import LogisticRegression as logit

logitmodel1 = logit()

# Preprocessing
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train_std = scaler.transform(X_train)
X_test_std = scaler.transform(X_test)

logitmodel1.fit(X_train_std, y_train)

print(logitmodel1.score(X_test_std, y_test))

print(logitmodel1.coef_)
```

```
import statsmodels.api as sm
from statsmodels.formula.api import logit

formula = ('Click')
```

```
plot_classifier(logitmodel1, X_test_std, y_test, proba = True, xlabel = "Age", ylabel = "Area Income")
```

KLASSIFIKATION

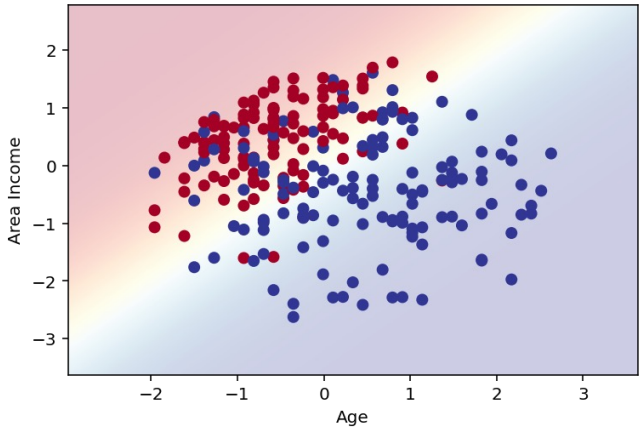
0.888 [[-1.20646073 -3.02661727]]



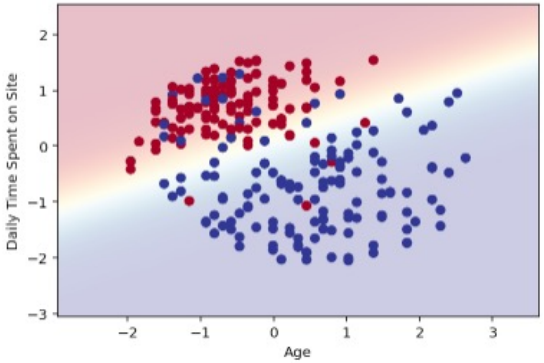
- Mehrverdiener und intensive Internetnutzer klicken wahrscheinlich eher nicht auf die Werbebanner

- Ältere und mittleres bis geringeres Einkommen zeigen tendenziell mehr Interesse

0.828 [[1.34613217 -1.32776685]]



0.9 [[1.26773576 -2.99601844]]



CUSTOMER SEGMENTATION - CLUSTERING

```
X = df[['Age', 'Area Income', 'Daily Time Spent on Site']][df['Clicked on Ad']==1].values
y = df['Clicked on Ad'].values
```

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

```
cluster_range = range(1,10)
cluster_errors=[]

for num_clusters in cluster_range:
    clusters = KMeans(num_clusters)
    clusters.fit(X)
    cluster_errors.append(clusters.inertia_)
```

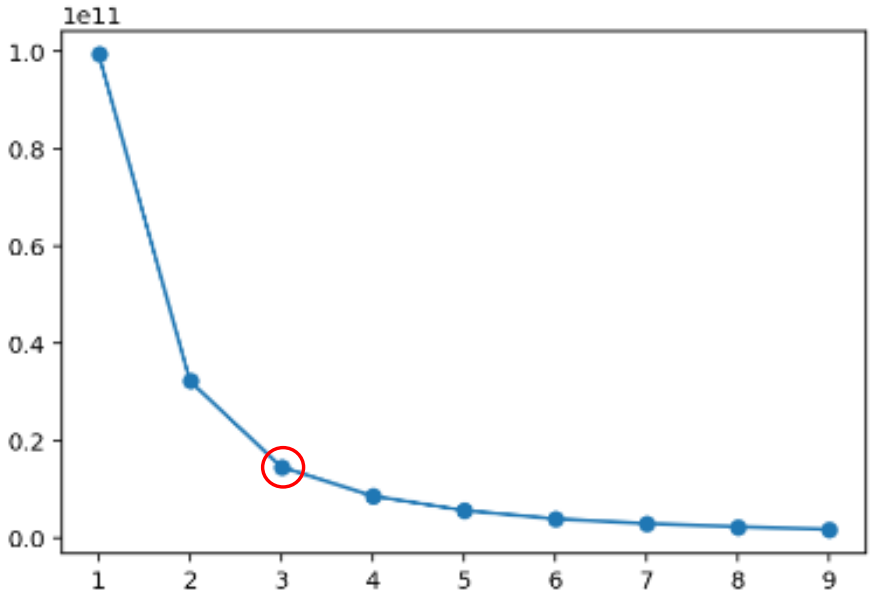
```
clusters_df = pd.DataFrame({'num_clusters':cluster_range,'cluster_errors':cluster_errors})
```

```
plt.plot(clusters_df.num_clusters, clusters_df.cluster_errors, marker = 'o')
```

```
from sklearn.cluster import KMeans as kmeans
means = KMeans(n_clusters=3)
means.fit(X)
y_kmeans = means.predict(X)
```

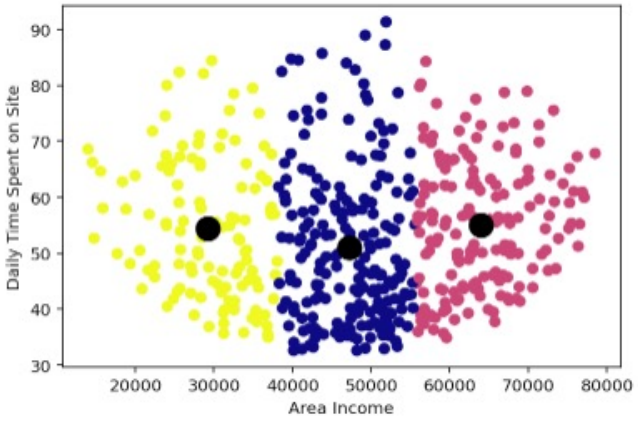
```
centers = means.cluster_centers_
centers
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans)
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.xlabel('Age')
plt.ylabel('Area Income')
```

- Ab drei Gruppen gibt es kein großen Unterschied im Error des Scores.



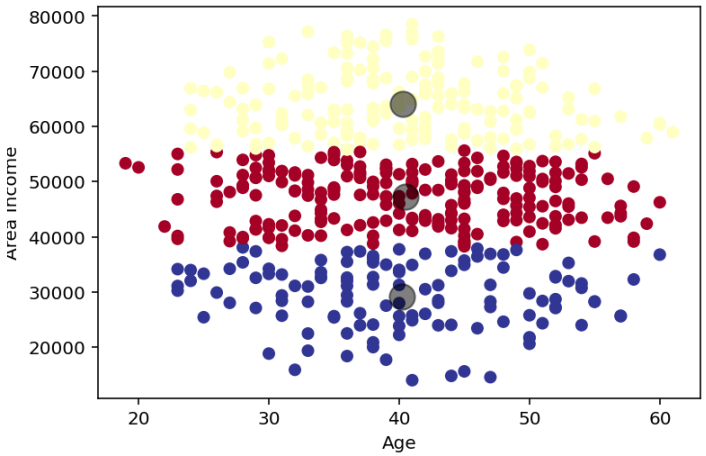
KMEANS - CLUSTERING

Text(0, 0.5, 'Daily Time Spent on Site')



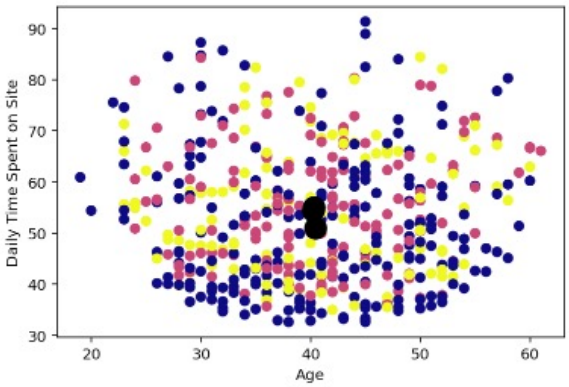
- Drei homogene Gruppen hinsichtlich „Area Income“ und „Daily Time Spent“ on Site bzw. Age
- Centroids verteilen sich gut

Text(0, 0.5, 'Area Income')

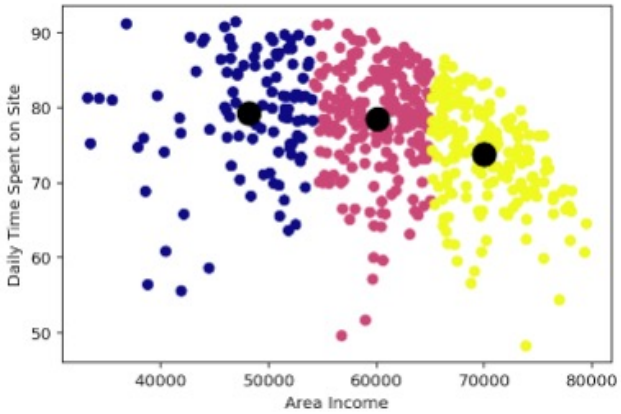


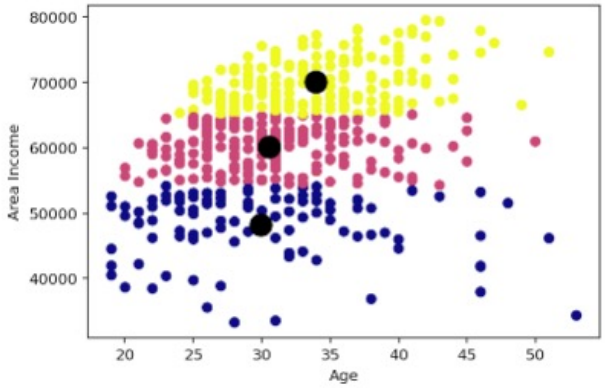
- Centroids alle beieinander, keine Gruppenunterscheidung nach den Merkmalen Age und Daily Time Spent on Site

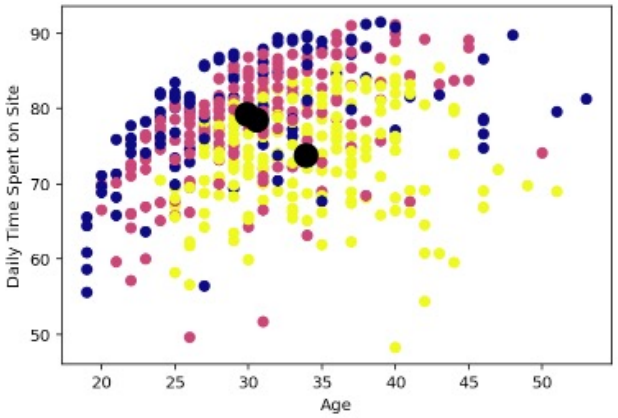
Text(0, 0.5, 'Daily Time Spent on Site')



KMEANS - CLUSTERING







ZUSAMMENFASSUNG ERGEBNISSE

- Kaum Geschlechterunterscheidung
- Ältere laute zeigen sich interessierter bei den Werbebannern
- Personen mit geringeren Einkommen zeigen größeres Interesse
- Starke Korrelation zw. Daily Time Spent on Time und Daily Internet Usage
- Über die Woche ähnliches Nutzungsverhalten
- Website Besuch sowie Klickrate am höchsten morgens (6-12) und nachts (22-6)
- Mehr Zeit im Internet bzw. auf der relevanten Website führt nicht zu mehr Klicks auf die Werbebanner
- Die Regressionsanalyse zeigt, das 91 % (wenn standardiert) des Models basierend auf den verwendeten Variablen erklärt wird
- Clusteranalyse identifiziert drei Gruppen, die heterogen zueinander sind, doch relativ homogen in sich selbst

WEITERE ANALYSE VORSCHLÄGE

- Wochentage und Tageszeiten könnten verkodiert werden, um diese in die Regression mit einzubeziehen
- Kontinente in weitere Analysen mit einzubeziehen
- Analyse auf Länder und / oder Städtebasis bei größeren Datensatz
- Natural Language Processing: Verwendung der „Ad Topic Line“ Daten. Analyse welche Worte bzw. Wortpaare am häufigsten angeklickt werden

EMPFEHLUNGEN

- Stärkere geschlechterspezifische Werbebanner
- Potenzial der jungen Menschen nutzen, da diese deutlich mehr Zeit im Internet verbringen
- Höhere Einkommensschichten abschöpfen mit aufgewerteter Ansprache
- Zwischen 6 und 12 Uhr bzw. Nachts zw. 22 und 6 Uhr Werbebanner schalten, meiste Frequenz auf Website und höchste Klickrate
- Asien und Europa stärker ausspielen, sind die aktivsten Weltregionen
- Potenziale in Südamerika nutzen


Gibt es noch Fragen?