# sgij_EDA

August 14, 2019

The notebook performs exploratory data analysis (EDA) on the SGIJ dataset

[91]:

```
import math
from datetime import date
from datetime import datetime
import numpy as np
import pandas as pd
from pandas.plotting import register_matplotlib_converters
import matplotlib
import matplotlib.pyplot as plt
import mysql.connector
from sklearn.cluster import KMeans
from sklearn import metrics
%matplotlib inline
```

Connect to MySQL databasef from credentials

[92]:

```
config = {
  'user': 'root',
  'password': 'thingtrack',
  'host': '127.0.0.1',
  'database': 'gaming',
  'raise_on_warnings': True
}

try:
  cnx = mysql.connector.connect(**config)
except mysql.connector.Error as err:
  if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
    print("Something is wrong with your user name or password")
  elif err.errno == errorcode.ER_BAD_DB_ERROR:
    print("Database does not exist")
  else:
    print(err)
```

Execute query and obtain dataset from database

[93]:

```
# create dataset from database
cursor = cnx.cursor()

query = ("SELECT pl.birthdate, SUM(ac.profit) AS profit"
         " FROM player pl, account ac"
         " WHERE pl.operator_id = ac.operator_id"
         " AND pl.player_id = ac.player_id"
         " GROUP BY ac.operator_id, ac. player_id")

cursor.execute(query)

# return a list of tuples
result = list(cursor.fetchall())
```

Transform date attribute and create tuples

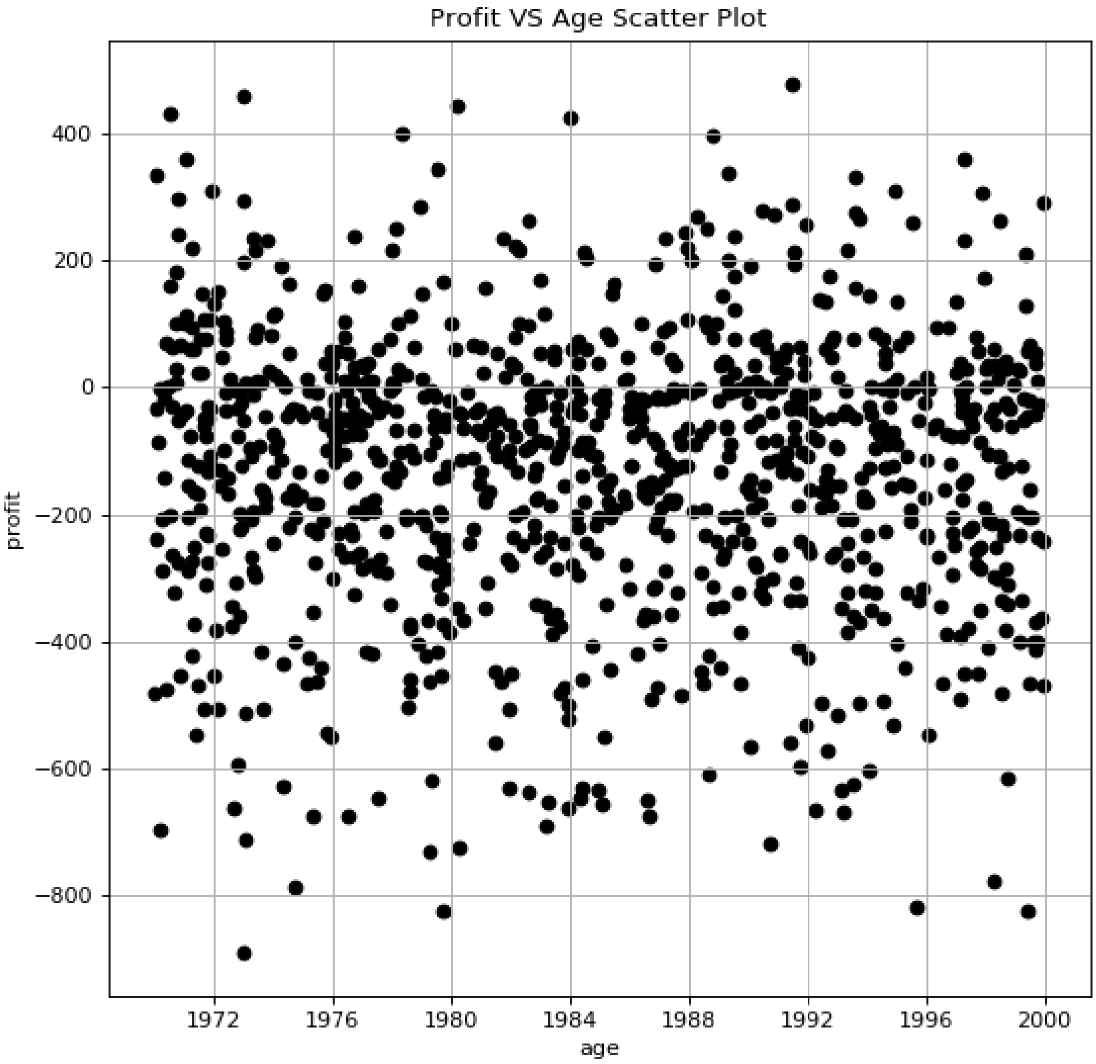
[94]:

```
# separate tuples
dates, profits = zip(*result)

#print(dates)

# convert string to date
#dates = [pd.to_datetime(d) for d in dates]

helper = np.vectorize(lambda x: datetime.combine(x, datetime.min.time()).
 ↪timestamp())

X = []
for row in result:
    X.append([helper(row[0]), row[1]])
```

Show Age vs Profit Scatter Plot

[99]:

```
plt.figure(figsize=(8, 8), dpi=80)
plt.scatter(dates, profits, color='k')
plt.title("Profit VS Age Scatter Plot")
plt.xlabel("age")
plt.ylabel("profit")
plt.grid()
plt.show()
```



Design de k-means with 2 clusters model for dataset

[96]:

```
model = KMeans(n_clusters=2).fit(X)
```

Print k-means centroides

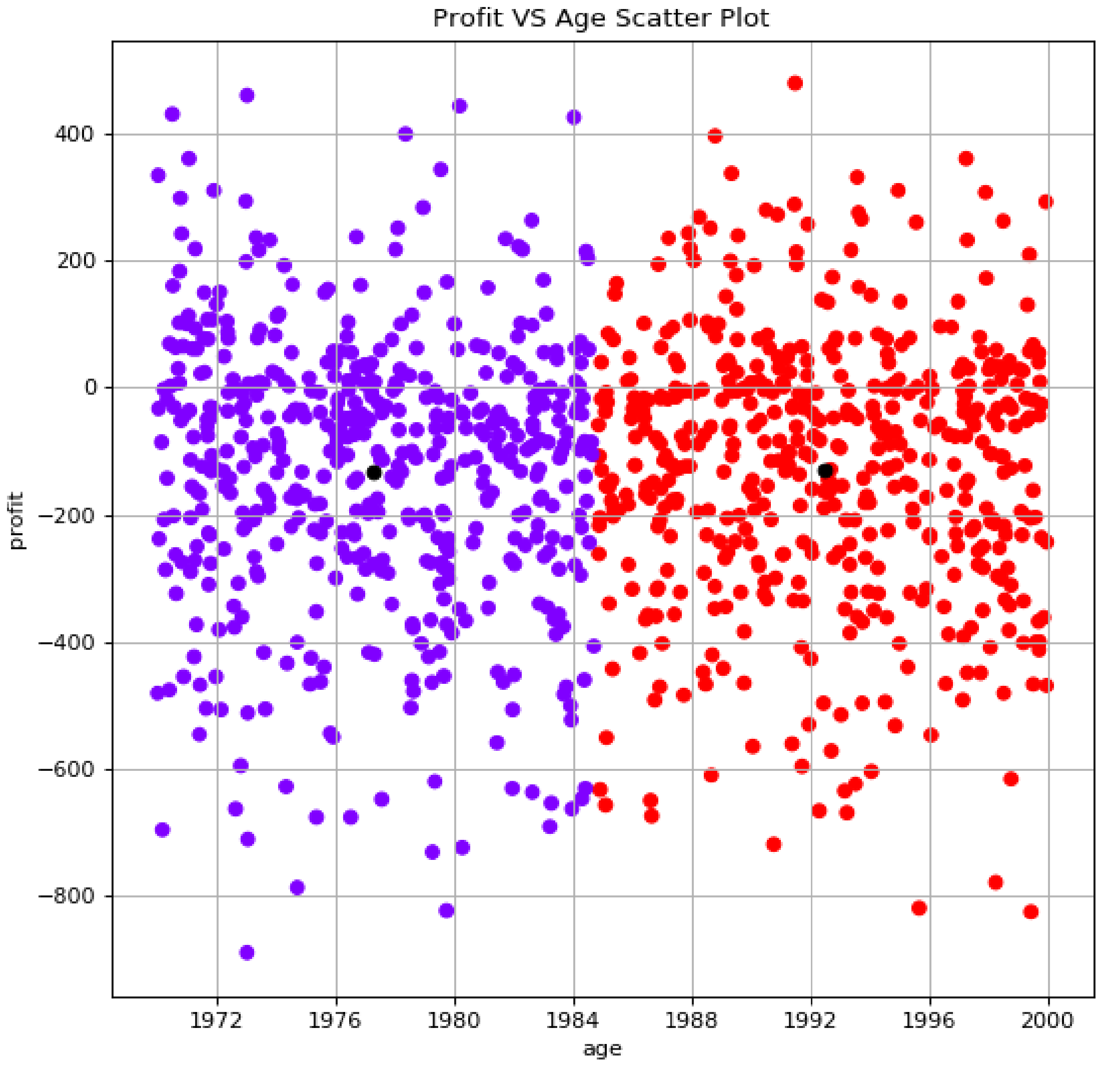
[97]:

```
centroides = []
for row in model.cluster_centers_:
    centroides.append((datetime.fromtimestamp(int(round(row[0]))), row[1]))

print(centroides)

centroides_dates, centroides_profits = zip(*centroides)
```

```
[(datetime.datetime(1977, 3, 31, 23, 49, 35), -131.3792415169661),
(datetime.datetime(1992, 6, 14, 19, 54, 21), -128.54108216432869)]
```

Plot the scatter plot and the centroides for tha dataset

```
[98]: plt.figure(figsize=(8, 8), dpi=80)
      plt.scatter(dates, profits, c=model.labels_, cmap='rainbow')
      plt.scatter(centroides_dates ,centroides_profits, color='black')
      plt.title("Profit VS Age Scatter Plot")
      plt.xlabel("age")
      plt.ylabel("profit")
      plt.grid()
      plt.show()
```



```
[ ]:
```

[ ]:

```
```