03_forelesning

February 23, 2021

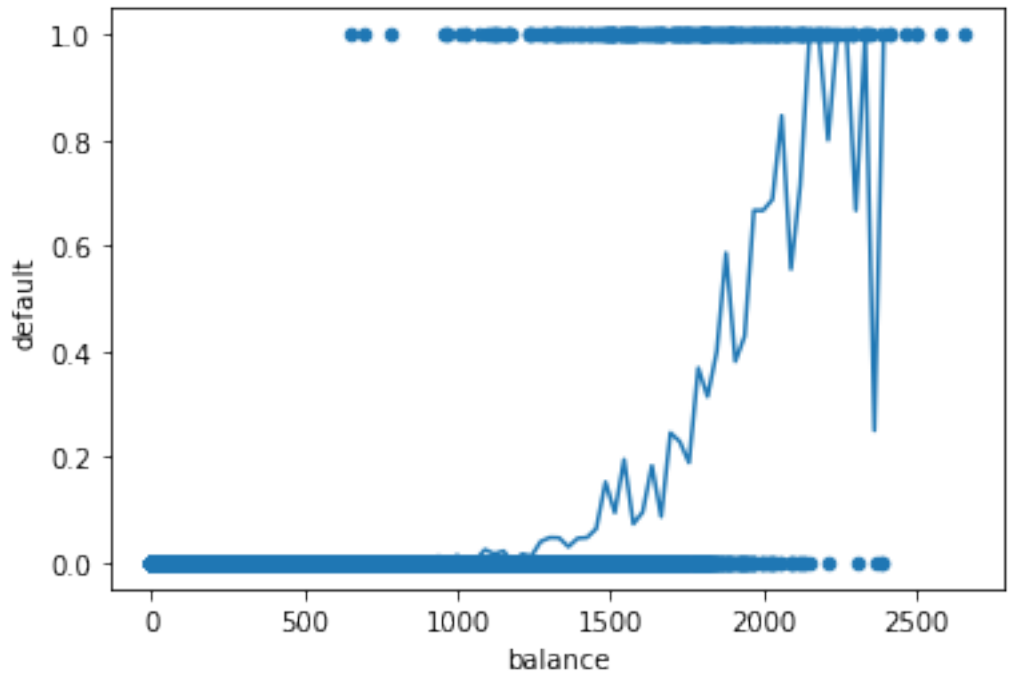
0.1 Repetisjon fra forrige gang

Forrige gang plottet vi et datasett med mislighold va kredittkortgjeld.

[6]:
```
# pakker
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Laste inn datasettet

data = pd.read_csv("../data/default.csv")
# rydde i data
data.replace("Yes", 1, inplace=True)
data.replace("No", 0, inplace=True)

data.plot.scatter("balance", "default")

x = np.linspace(0, 3000, 100)

df_yes = data[data["default"] == 1]
hist_yes = np.histogram(df_yes["balance"], bins=x)[0]
#display(hist_yes)

df_no = data[data["default"] == 0]
hist_no = np.histogram(df_no["balance"], bins=x)[0]
#display(hist_no)

andel_mislighold = hist_yes/(hist_yes+hist_no)
plt.plot(x[:-1], andel_mislighold)
```

```
<ipython-input-6-c9ccf0ba0e9a>:24: RuntimeWarning: invalid value encountered in
true_divide
  andel_mislighold = hist_yes/(hist_yes+hist_no)
```

[6]:
```
[<matplotlib.lines.Line2D at 0x133b41910>]
```



Dette var det vi gjorde forrige gang. I tillegg prøvde vi å tilpasse en sigmoid funksjon til dataene.

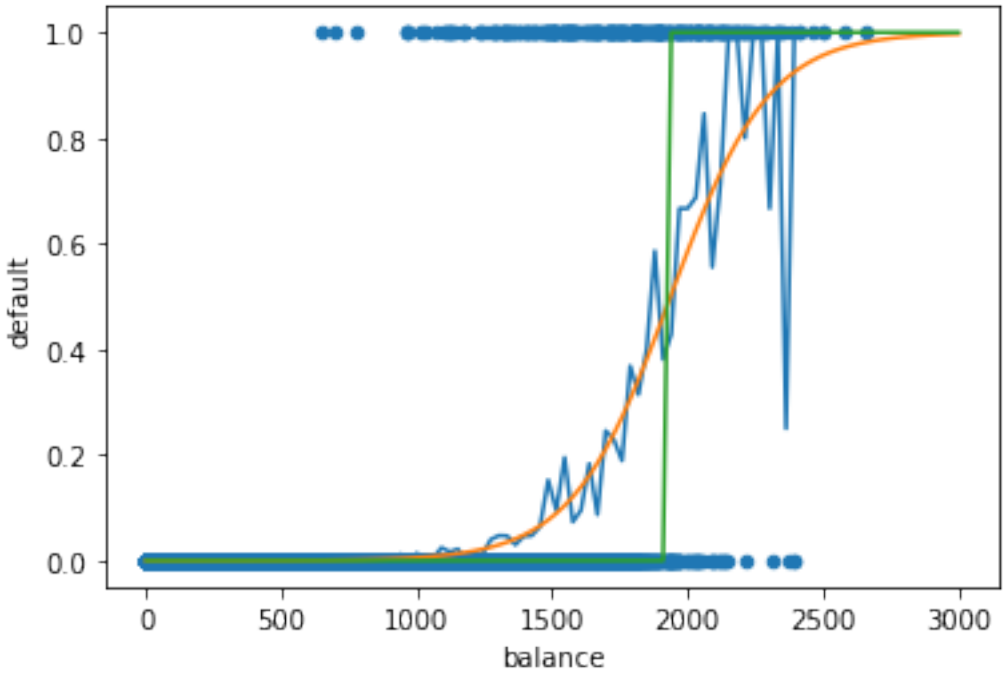
1 Live-programmering <1>

Nås skal vi først bruke scikit-learn til å gjøre logistisk regresjon på dette datasettet

[7]:
```
# <1>
# antar vi her beregnet andel mislighold fra før
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
fit = model.fit(data[["balance"]], data["default"])

print("Coefficients: ", model.coef_)
print("Intercept: ", model.intercept_)

beta0 = model.intercept_[0]
beta1 = model.coef_[0][0]

def sigmoid(x, beta0, beta1):
    z = beta0+beta1*x
    return np.exp(z)/(1+np.exp(z))

prediction = model.predict(np.transpose([x]))
```

```
data.plot.scatter("balance", "default")
plt.plot(x[:-1], andel_mislighold)
plt.plot(x, sigmoid(x, beta0, beta1))
plt.plot(x, prediction)
plt.savefig("default.png", dpi=300)
```

```
Coefficients:  [[0.00549892]]
Intercept:  [-10.65132824]
```



1.1 Flere prediktive variable <2>

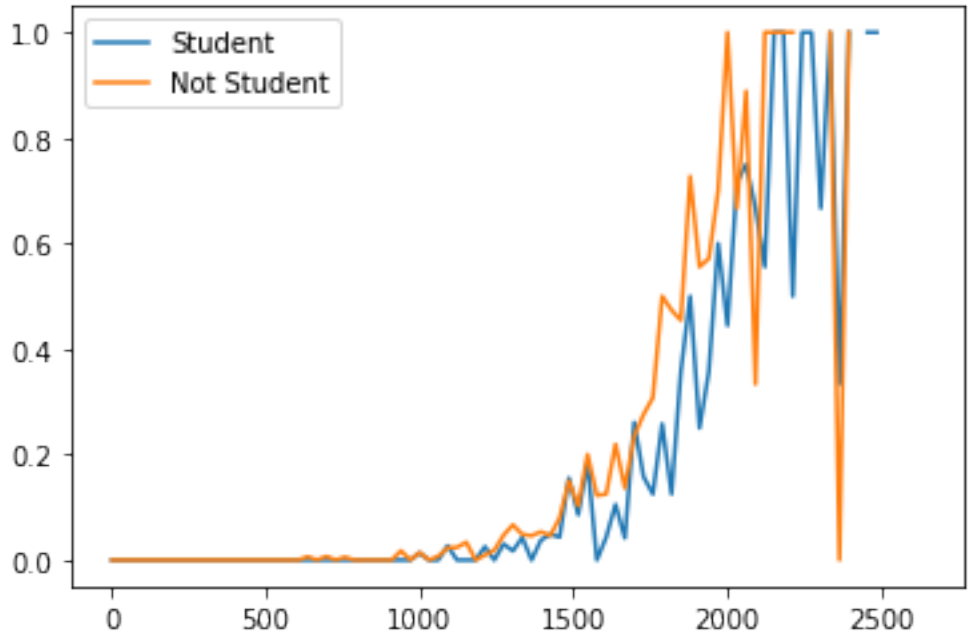
Under her plotter vi mislighold som funksjon av kredittkortsaldo, men fordelt på om kunden er student eller ikke.

[8]:
```
data_student = data[data["student"] == 1]
data_not_student = data[data["student"] == 0]

x = np.linspace(0, 3000, 100)

# Student
df_yes = data_student[data_student["default"] == 1]
hist_yes = np.histogram(df_yes["balance"], bins=x)[0]
```

```
df_no = data_student[data_student["default"] == 0]
hist_no = np.histogram(df_no["balance"], bins=x)[0]

# Plotte for student
andel_mislighold = hist_yes/(hist_yes+hist_no)
plt.plot(x[:-1], andel_mislighold, label="Student")

# Ikke student
df_yes = data_not_student[data_not_student["default"] == 1]
hist_yes = np.histogram(df_yes["balance"], bins=x)[0]

df_no = data_not_student[data_not_student["default"] == 0]
hist_no = np.histogram(df_no["balance"], bins=x)[0]

# Plotte for ikke student
andel_mislighold = hist_yes/(hist_yes+hist_no)
hax = plt.plot(x[:-1], andel_mislighold, label="Not Student")
plt.legend()
```

```
<ipython-input-8-cf431b4d3c92>:14: RuntimeWarning: invalid value encountered in
true_divide
  andel_mislighold = hist_yes/(hist_yes+hist_no)
<ipython-input-8-cf431b4d3c92>:25: RuntimeWarning: invalid value encountered in
true_divide
  andel_mislighold = hist_yes/(hist_yes+hist_no)
```

[8]: <matplotlib.legend.Legend at 0x133b41730>



2 Logistisk modell med flere prediktive variable

[10]:
```
display(data)
X = data[["balance", "student"]]
Y = data["default"]

model = LogisticRegression()
model.fit(X, Y)

display(model.coef_)
display(model.intercept_)
```

	Unnamed: 0	default	student	balance	income
0	1	0	0	729.526495	44361.625074
1	2	0	1	817.180407	12106.134700
2	3	0	0	1073.549164	31767.138947
3	4	0	0	529.250605	35704.493935
4	5	0	0	785.655883	38463.495879
...	...	...	...	...	...
9995	9996	0	0	711.555020	52992.378914
9996	9997	0	0	757.962918	19660.721768
9997	9998	0	0	845.411989	58636.156984
9998	9999	0	0	1569.009053	36669.112365
9999	10000	0	1	200.922183	16862.952321

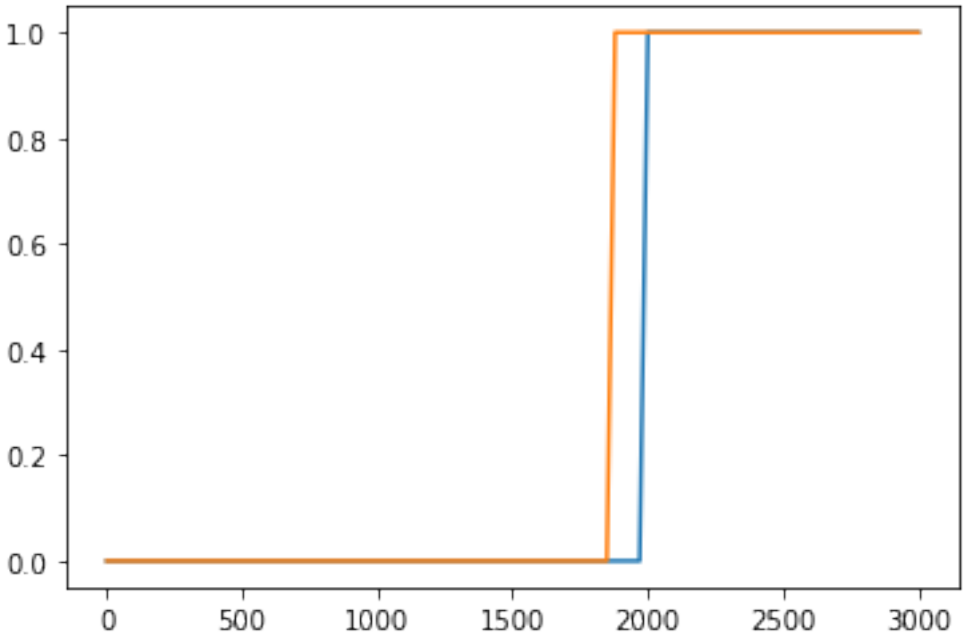
[10000 rows x 5 columns]

```
array([[ 0.00573175, -0.69968031]])
```

```
array([-10.7447422])
```

[12]:
```
x = np.linspace(0, 3000, 100)

inputs = pd.DataFrame({"balance" : x, "student": np.ones(x.shape)})
y = model.predict(inputs)
plt.plot(x, y, label="Student")

inputs = pd.DataFrame({"balance" : x, "student": np.zeros(x.shape)})
y = model.predict(inputs)
plt.plot(x, y, label="Not student")
```

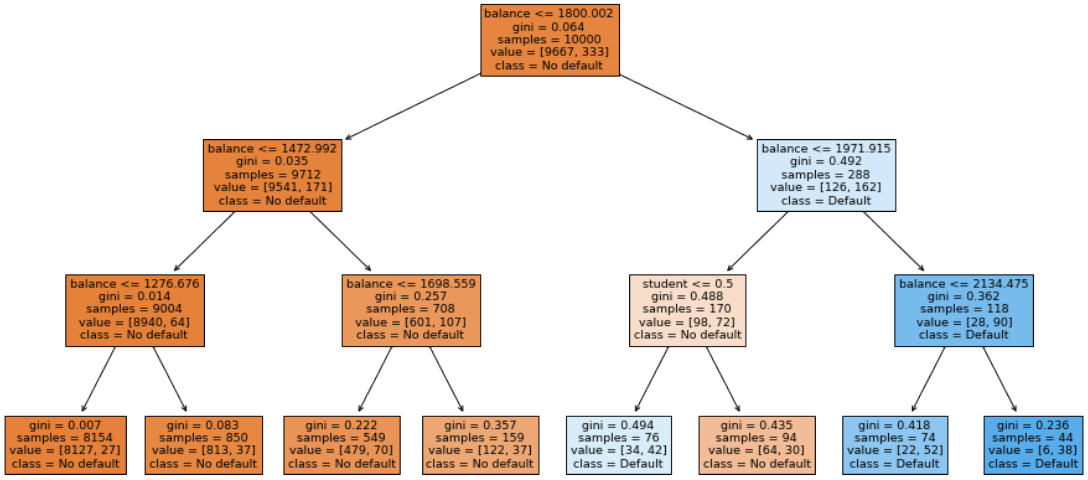
[12]:
```
[<matplotlib.lines.Line2D at 0x133caf880>]
```



Merk at her plottet vi selve prediksjonen, og ikke sigmoidfunksjonen inne i modellen. Derfor er den så bratt. Den går rett fra 0 til 1 når sigmoiden passerer 1/2.

3 Beslutningstre med studentdata

[115]:
```
from sklearn import tree
clf = tree.DecisionTreeClassifier(max_depth=3)
clf.fit(X, Y)

plt.figure(figsize=(16, 8))
tree.plot_tree(clf, feature_names=X.columns, class_names=["No default",␣
↪"Default"], filled=True)
print()
```



[116]:
```
prediction = clf.predict(X)
from sklearn.metrics import accuracy_score
accuracy_score(Y, prediction)
```

[116]: 0.9737

[ ]: