A LITTLE MACHINE LEARNING IN PYTHON

BY SOROOSH AZARY MARHABI

AZARY.IR

ABOUT ME

Artificial Intelligence MSc @ Sharif

Full Stack Web Developer @ Torob.ir

Come work with us!

A FICTIONAL PROBLEM

Solved using machine learning algorithms

A FICTIONAL FUN GAME

with in-app-purchasing

You can buy gold

Lets boost up revenue!

CHURN RATE

Cheaper to keep current users

After he's left is too late

PREDICT LEAVERS

And give them free stuff

WHAT WE KNOW

For each player:

- Minutes played / week
- Money spent / week
- Who has left, in the past weeks

PREDICTION

Using data of past 2 weeks,

Predict leavers of this week

CLASSIFICATION

Classify an input to a class

Some prespecified classes

TRAINING DATA

Minutes & money of 2 weeks ago

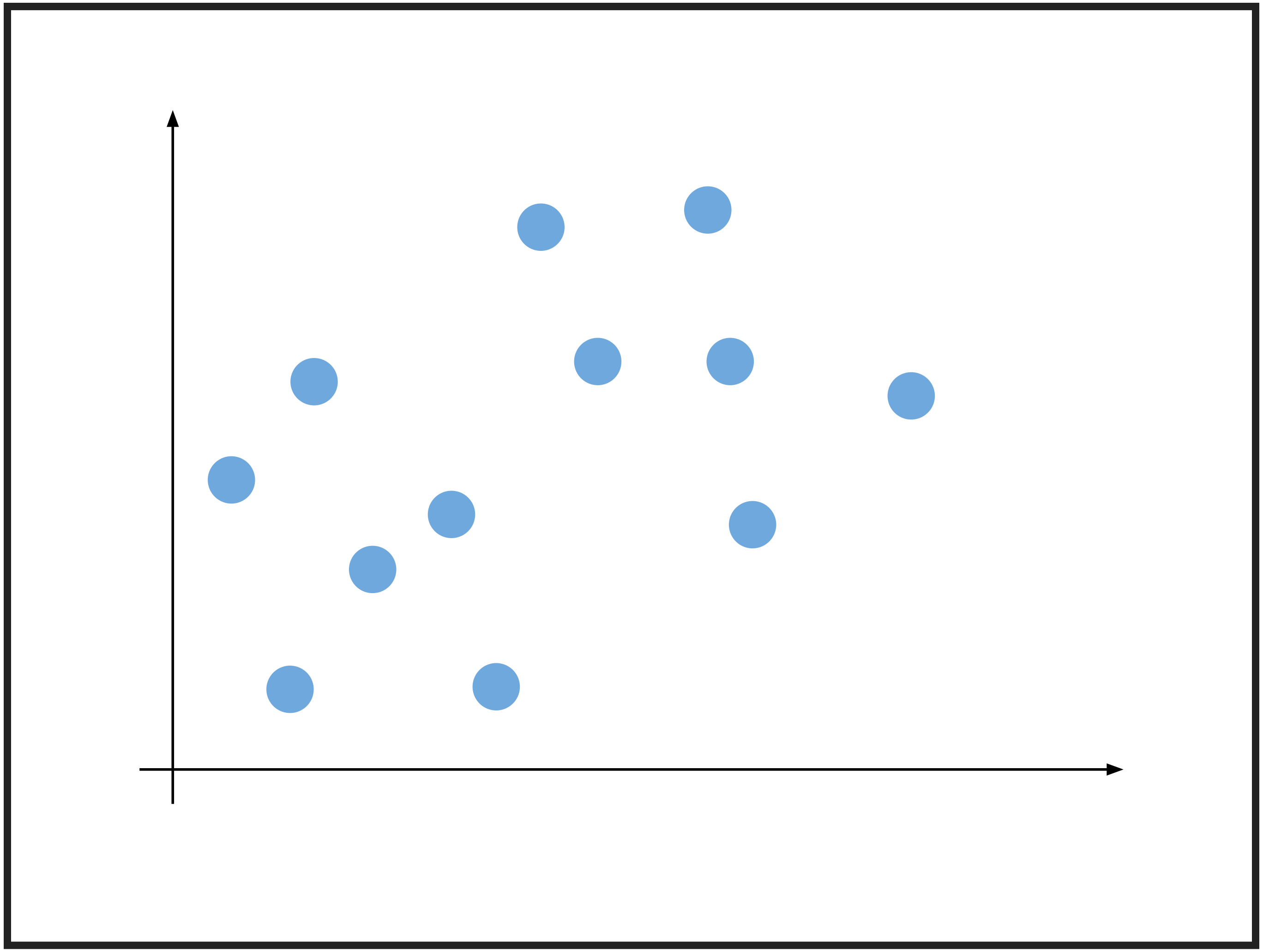
Left or stayed in the game in past week

PREDICTION DATA

Minutes & money of last week

Who will leave in this week

VISUALISATION



LABELS

PREDICT

THE LINE

SVM

"Support Vector Machine"

Finds best line

SCIKIT-LEARN TO RESCUE

A kit of Scipy

Very mature

Lots of proven algorithms

scikit-learn.org

CODE

```
from sklearn.svm import SVC

train_parameters = get_parameters()
train_classes = get_classes()
predict_parameters = get_parameters_to_predict()

classifier = SVC()
classifier.fit( train_parameters, train_classes )

predicted_classes = classifier.predict( predict_parameters )
```

SVM PROPERTIES

So fast

Low error rate

"Linear separability" problem

LACK OF LUCK

K-NEAREST NEIGHBORS

Finds k nearest training samples

Decides by their labels

KNN VISUALISATION

KNN CODE

```
from sklearn.neighbors import KNeighborsClassifier

train_parameters = get_parameters()
train_classes = get_classes()
predict_parameters = get_parameters_to_predict()

classifier = KNeighborsClassifier( n_neighbors = 5 )
classifier.fit( train_parameters, train_classes )

predicted_classes = classifier.predict( predict_parameters )
```

KNN PROPERTIES

No training time

Slow prediction, when training is large

SPENDING PROBLEM

Who will spend how much?

Promotions to maximize revenue

WHAT WE KNOW

For each player:

- Minutes / week
- Money spent / week, for the past weeks

TRAINING DATA

For each player:

- Minutes, for 2 weeks ago
- Money spent, for last week

PREDICTION DATA

For each player:

- Given minutes played last week,
- Predict money for this week

VISUALISATION

REGRESSION

Connect the dots!

A real valued function

VISUALISATION

CODE

```python
from sklearn.linear_model import ElasticNet

train_parameters = get_parameters()
train_spendings = get_spendings()
predict_parameters = get_parameters_to_predict()

regressor = ElasticNet()
regressor.fit( train_parameters, train_spendings )

predicted_spendings = regressor.predict( predict_parameters )
```

REGRESSION PROPERTIES

Bad extrapolation

THAT'S IT!