Fundamental Concepts in Data Insight:

Demo: Machine Learning Concepts

Fundamentals for a General Audience

The following code sets up the datasets needed for the demo. It obtains a stop and search dataset for "The City of London" and inserts it into a relational database.

```
stopsearch = pd.read_csv('data/stopandsearch.csv').dropna()
database = sqlite3.connect(':memory:')
query    = database.cursor()

stopsearch.to_sql('stopsearch', database, if_exists='replace')
```

What dataset are we using?

```
stopsearch.sample(10)
```

	Age	Gender	Date	Outcome
252	21.0	Male	2021-03-29T10:12:24+00:00	False
139	29.5	Male	2021-03-18T02:53:22+00:00	True
287	13.5	Male	2021-03-31T06:56:46+00:00	True
214	29.5	Male	2021-03-26T06:58:28+00:00	False
148	29.5	Male	2021-03-19T01:23:37+00:00	False
0	21.0	Male	2021-03-01T03:11:45+00:00	True
99	29.5	Male	2021-03-12T10:07:18+00:00	False
262	29.5	Male	2021-03-30T05:57:21+00:00	True
258	42.5	Male	2021-03-30T03:26:17+00:00	False
263	21.0	Male	2021-03-30T06:38:22+00:00	False

How do I query this dataset?

```
SELECT AVG(Outcome) AS RateOfSuccess
FROM stopsearch
WHERE age < 18
```

```
result
```

	RateOfSuccess
0	0.233333

How do I predict probability of a stop-and-search success?

As with all machine learning predictive problems, we are looking to predict an unobserable y in a new situation, where we can only observe x.

Here suppose we want to predict whether a police officer will find an illegal item of interest (y) given only the supects age x (NB. this dataset is specific to a small policing area, where age could be a significant predictive factor).

To do this we will use the `k`-Nearest neighbors algorithm (kNN).

kNN is a predictive machine learnig algorithm, it says,

1. query your historical dataset for `k` similar points in x (eg., `k = 3`, so *3 rows*)
2. report the average of their historical value for y

You, the practicioner, choose `k`.

A small `k` averages a few similar rows, a big `k` nearly the whole dataset. The bigger the `k`, the more your predictions will all be very similar. A big `k` is useful if there's lots of noise in the dataset and if if all your suspects are quite similar.

What's a k-NN query?

Suppose I choose `k = 5`, and I'm interested in a prediction for a `suspect_age = 18`,

```
SELECT
    18              AS "Suspect Age",
    AVG(Outcome)    AS "Prob(FindingItem)"
FROM stopsearch
ORDER BY ABS( age - 18 )
LIMIT 5
```

```
pd.read_sql(sql, database)
```

	Suspect Age	Prob(FindingItem)
0	18	0.340741

The query here *ranks* searches by how similar their suspect's age was, `ABS(age - {suspect_age})` and chooses `k` of them, `LIMIT {k}`.

`ABS` here means *ignoring the sign*, so if my age was 25 then we count people 23 and 27 as equally similar to me.

How would we automate this analysis with python?

Above we have used SQL and manually written the algorithm (in SQL) to compute our prediction. We do not have to do this.

Almost all algorithms are pre-written, and can be fully executed in one line of code,

```
model = KNN(5).fit(stopsearch[['Age']], stopsearch['Outcome'])
```

We call the solution of a machine learning problem a *model*. It is the device which allows us to make predictions,

Here now, it is very easy to predict a tip for various ages: $25, 10, 45,$

```
model.predict_proba([
    [25],
    [10],
    [45],
])
```

```
array([[0.6, 0.4],
       [1. , 0. ],
       [1. , 0. ]])
```

Above, `[0.6, 0.4]` means a 60% chance of `Outcome = False`, and a 40 change of `True`.

So, $P(FindItem|Age = 25) = 40\%$

Aside: Expanding this example

We can also involve the other columns: gender, time of day, month. Also, the original dataset contains lat/long locations which are likely to be predictive.

Analysis of these variables was omitted for clarity.

Summary & Review

The purpose of this demo was to *illustrate* how simple machine learning systems are in practice. Though some experience is required in programming, the amount of programming needed to create a machine learning solution is minimal.

In realworld projects, the vast majority of programming goes into obtaining, cleaning, preparing and reporting on datasets; or into the automation systems behind applications: **not** into machine learning itself!