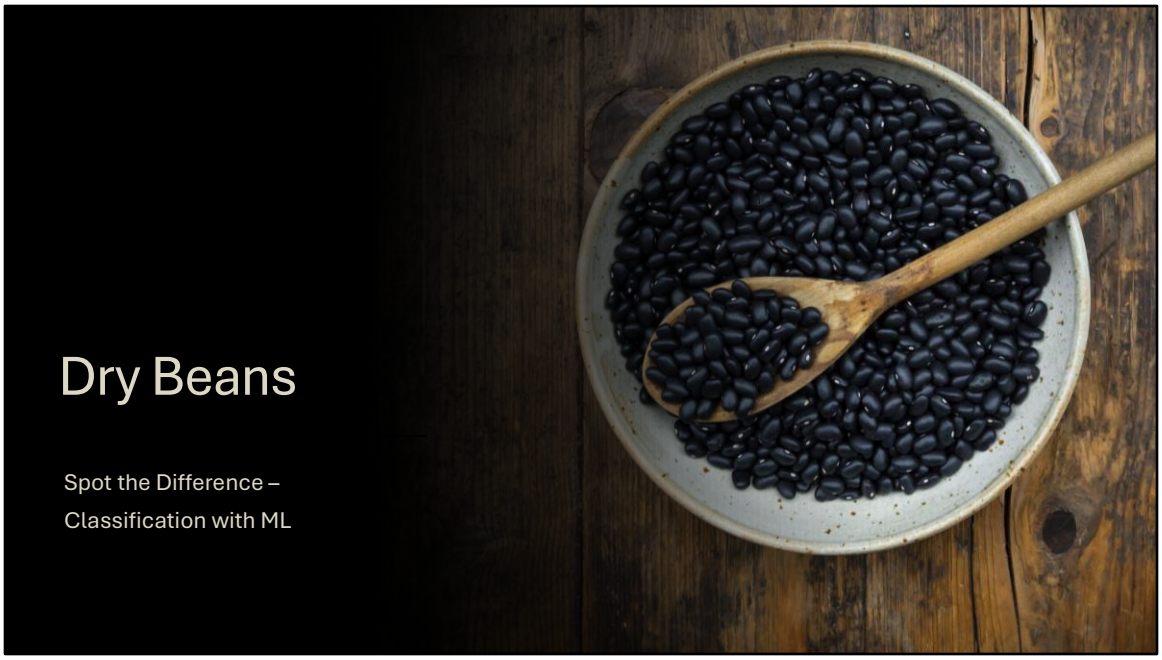


Today we are trying to categorise 7 types of white(ish) bean - why might we want to do this?

- Beans - hugely important market, possibly the most produced edible legume crops in the world.
- When you buy beans as a consumer, they are already categorised nicely in bags.
- The main use case for being able to efficiently categorise them is for farmers making sure you are planting the right thing in the right place.

Not that difficult, you might think - but let's have a look at the beans we are trying to categorise...

This was originally a Turkish dataset, so we have Turkish names. What can we see here?

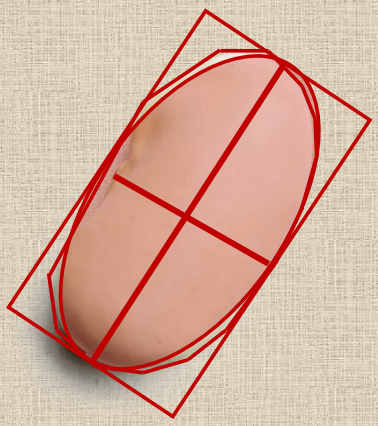
- Barbunya: aka Borlotti/Cranberry bean, used in Pilaki. Beige with red stripes, large round oval
- Bombay & Cali – slightly more kidney shaped
- Horoz looks like a pill
- Very little to tell between the others

The main point of this beautifully-constructed slide is to demonstrate that they are not always that easy to tell apart.

A word on the dataset:

- The data is based on over 13,000 photos of these 7 types of beans.
- The data is already converted into numbers, largely based on pixel area and various geometric relationships.
- We are training our model here on the assumption that the image recognition system is decent.

What do we know?



Major Axis Length

Minor Axis Length

Extent (Area vs Bounding Box)

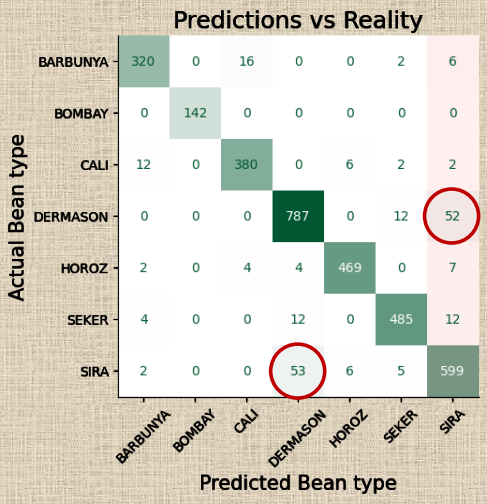
Solidity (Area / Convex Shell)

Shape Form 4: $\frac{A}{\frac{L}{2}*\frac{l}{2}*\pi}$

In the dataset we have 16 features – 12 dimensions and 4 shape forms.

To give an example of a few of the features we are working with *(slide illustrations one-by-one)*

Model Performance



Overall accuracy: 93.51%

Baseline accuracy: 25.01%

Average precision: **94.68%**

Average recall: **94.56%**

Toughest bean: **SIRA**
90.08% recall
88.35% precision

What we really want to know – how good is our model at categorising the beans.

Pretty good based on overall accuracy.
Baseline accuracy is based on guessing the most common bean type in the data (Dermason).

Averages here are unweighted, meaning each bean type counts the same for precision and recall.

Under the hood

Support Vector Classifier

- C-value: 2
- Kernel: “RBF”
- Gamma: “scale”

Alternatives trialled:

- RandomForestClassifier
- LogisticRegression
- K-Nearest Neighbours
- DecisionTreeClassifier

Other considerations:

- Clustering
- PCA

After evaluating the performance of several classification models on the training data and hyperparameter tuning, I settled on using the SVC model with a C-value of 2.

I experimented with Kmeans clustering before classification, to see if that could identify patterns that would enhance the classification. As well as being potentially unreliable on new datasets, the performance with clustering first was no better.

I also tried Principal Component Analysis, which we will talk about on the next slide…

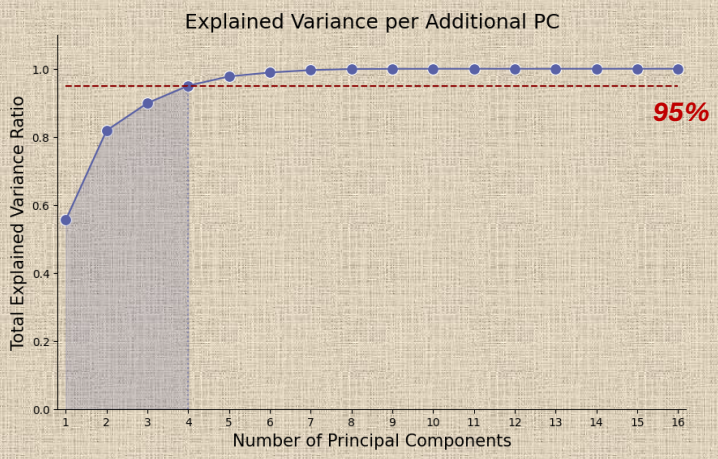
Principal Component Analysis

PC1: **Length**

PC2: **Width**

PC3: **Concavity**

PC4: **Squareness**



At first glance, PCA looked like it could be fantastic:
- 90% of the variance explained by 3 components, and 95% by four.

Ultimately, however, it did not improve training scores, so our chosen model does not use PCA.

It's not bad – but finding that 10% of your field is not what you thought you planted is probably not ideal.

Sira/Dermason confusion is key to this – if those are not beans which you farm, this could still be very useful.

Could potentially improve the model with more features, but that would likely require more expensive equipment to collect the data.