Poker Dataset Training

By Prince, Lakshya, LV

TABLE OF CONTENTS

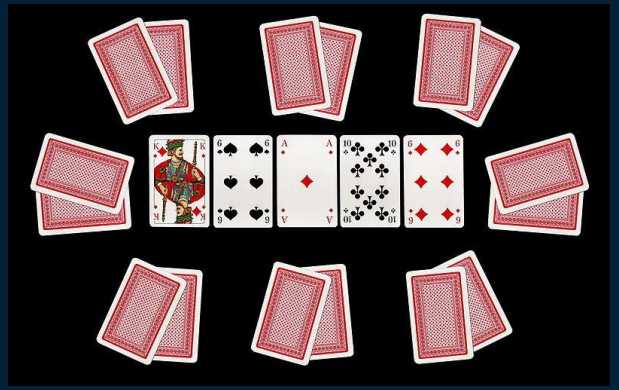
01

Introduction

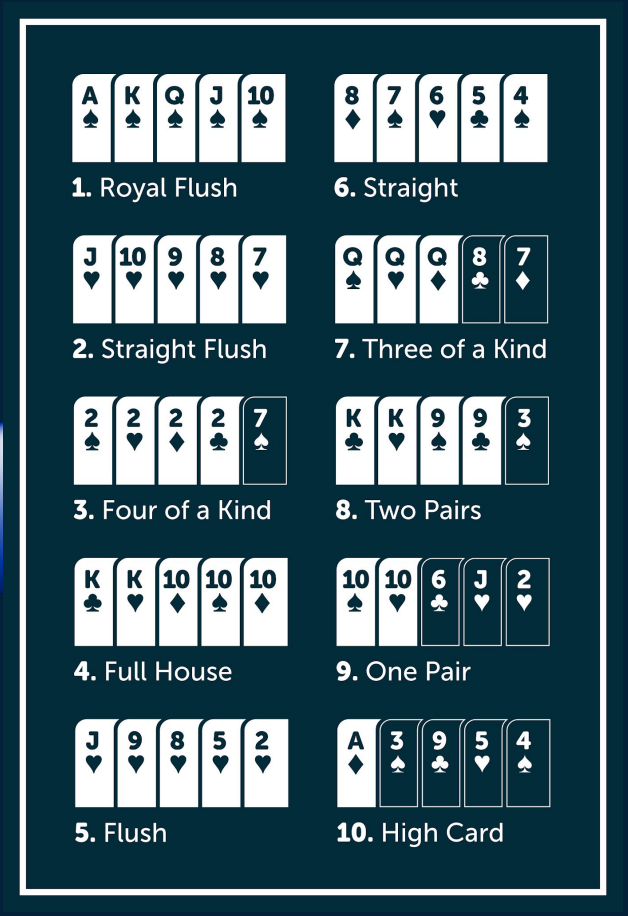
Poker Dataset

What is Poker ?

- 2 Cards one after another distributed to each player.
- 5 Cards on the table dealt by Dealer (person distributing the cards)
- Best 5 cards are picked for rankings that determines the win.
- 10 types of Patterns



Target Variables (0-9)



0.000154%

The royal flush is a case of the straight flush. It can be formed 4 ways (one for each suit), giving it a probability of 0.000154% and odds of 649,739 : 1.

DataSet Structure & Features.

	S1	C1	S2	C2	S3	C3	S4	C4	S5	C5	CLASS
0	1	10	1	11	1	13	1	12	1	1	9
1	2	11	2	13	2	10	2	12	2	1	9
2	3	12	3	11	3	13	3	10	3	1	9
3	4	10	4	11	4	1	4	13	4	12	9
4	4	1	4	13	4	12	4	11	4	10	9

1. S1 "Suit of card #1" Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
2. C1 "Rank of card #1" Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)
3. S2 "Suit of card #2" Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
4. C2 "Rank of card #2" Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)
5. S3 "Suit of card #3" Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
6. C3 "Rank of card #3" Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)
7. S4 "Suit of card #4" Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
8. C4 "Rank of card #4" Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)
9. S5 "Suit of card #5" Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
10. C5 "Rank of card 5" Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)
11. CLASS "Poker Hand" Ordinal (0-9)

DataSet Overview

Instances

1025010

Features

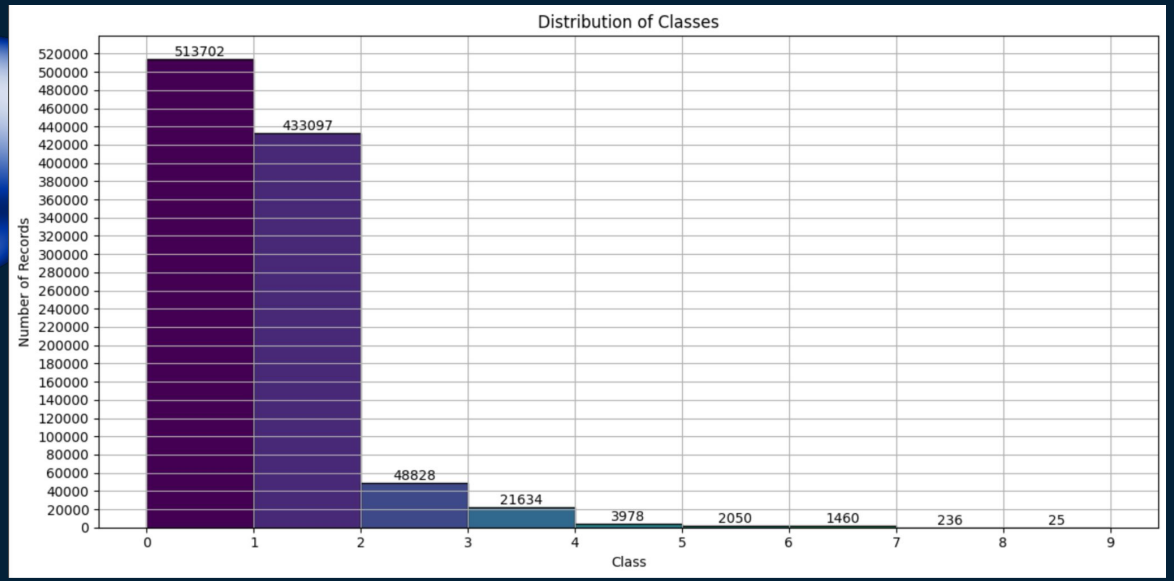
10
Categorical, Integer

Dataset

Multivariate

Split

70% Training
30% Testing




```
Training dataset shape:  (717507, 10)
Testing dataset shape:  (307503, 10)
```

02 Models

Poker Dataset

Decision Tree

- Performance: Accuracy of 63%
- Characteristics: Decision trees are a type of model that makes decisions based on a set of binary rules. They're quite interpretable but can be prone to overfitting if not pruned correctly.
- Analysis: Given that poker hand classification is quite a complex problem with potential non-linear relationships, a simple decision tree might not capture all the complexities. However, decision trees can serve as a good baseline and are very fast to train.

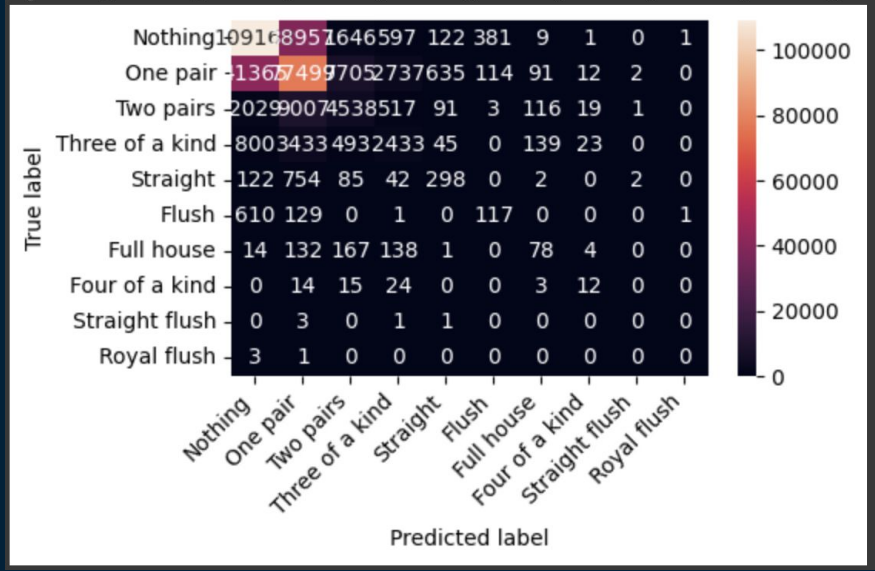
```
print("Number of nodes in the decision tree:", clf.tree_.node_count)
print("Depth of the tree:", clf.get_depth())
```

```
Number of nodes in the decision tree: 441143
Depth of the tree: 40
```

```
print("Accuracy for descision Tree is", accuracy(cm2))
Testing_error = 1 - accuracy(cm2)
print("Testing Error is:", Testing_error)
```

```
Accuracy for descision Tree is 0.6393466080005723
Testing Error is: 0.3606533919994277
```

```
Confusion matrix:
[[109168  38957   1646    597    122    381      9      1      0      1]
 [ 41365  77499   7705   2737    635    114     91     12      2      0]
 [  2029   9007   4538    517     91      3    116     19      1      0]
 [   800   3433    493   2433     45      0    139     23      0      0]
 [   122    754     85     42    298      0      2      0      2      0]
 [   610    129      0      1      0    117      0      0      0      1]
 [    14    132    167    138      1      0     78      4      0      0]
 [     0     14     15     24      0      0      3     12      0      0]
 [     0      3      0      1      1      0      0      0      0      0]
 [     3      1      0      0      0      0      0      0      0      0]]
```



Decision Tree Model

Problems with Decision Tree

- Adaptability Issues: Difficulty in adapting decision tree models to different poker environments. Poker strategies can vary widely, requiring continual updates to the model. While working on the model it also vary as their would be changing strategy on poker players and there are various different game format as well.

- Data Quality Issues: Importance of high-quality data for training decision tree models. potential data quality issues: bias, class imbalance which is where distribution of data between classes is highly skewed. So Class 1-5 have most data and rest have very less leading to imbalance.

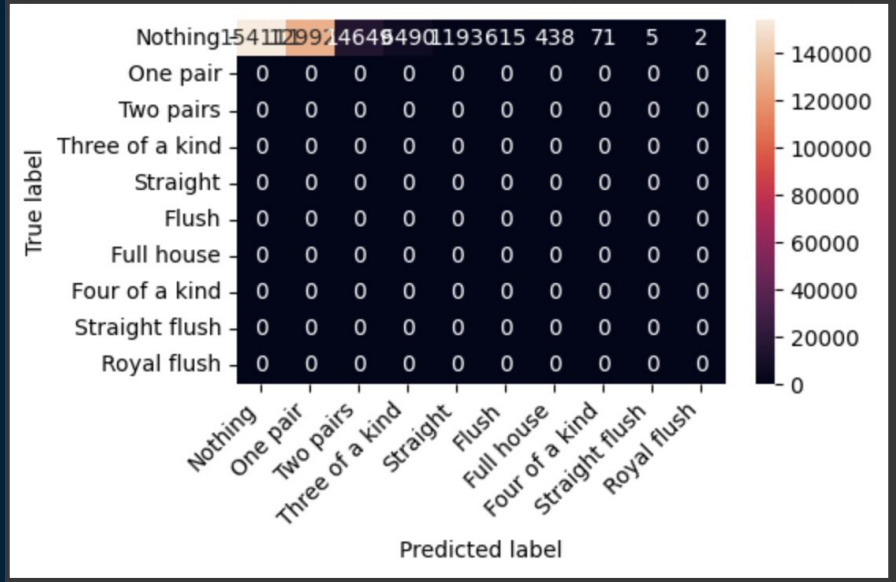
GaussianNB

- Performance: Accuracy of 50%
- Characteristics: Naive Bayes classifiers assume that the features are independent given the class labels. This assumption is rarely true in real life and especially not in a game like poker where the combination of cards greatly influences the hand's strength.
- Analysis: The assumption of feature independence could be the reason for the lower accuracy since poker hands are inherently about the relationship between cards.

```
gnb = GaussianNB()

# Training the classifier
gnb.fit(X_train_scaled, Y_train)

# Making predictions
Y_pred = gnb.predict(X_test_scaled)

# Calculating the accuracy of the model
accuracy = accuracy_score(Y_test, Y_pred)

print(f"Model accuracy: {accuracy * 100:.2f}%")
```

```
Model accuracy: 50.12%
```

```
Confusion matrix:
[[154111 129929  14649   6490   1193    615    438     71      5      2]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```



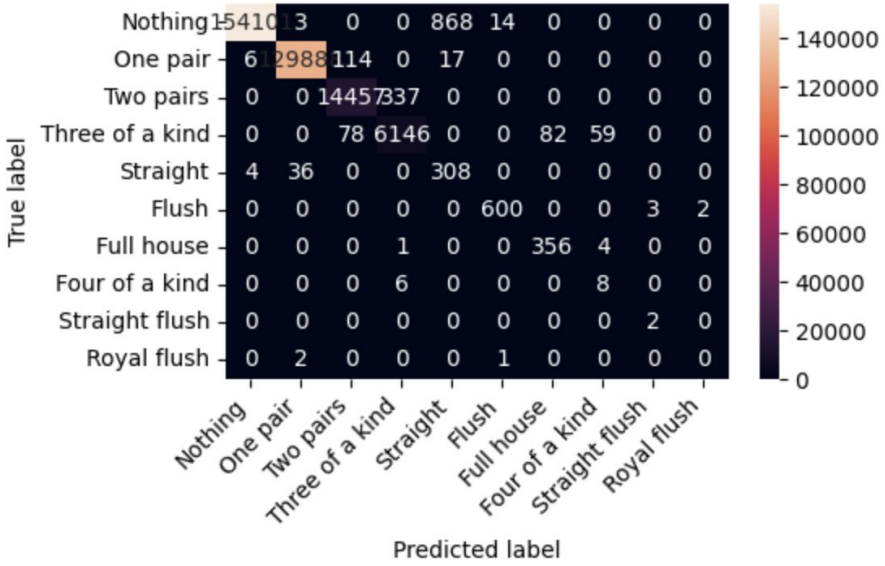
GaussianNB

Artificial Neural Network

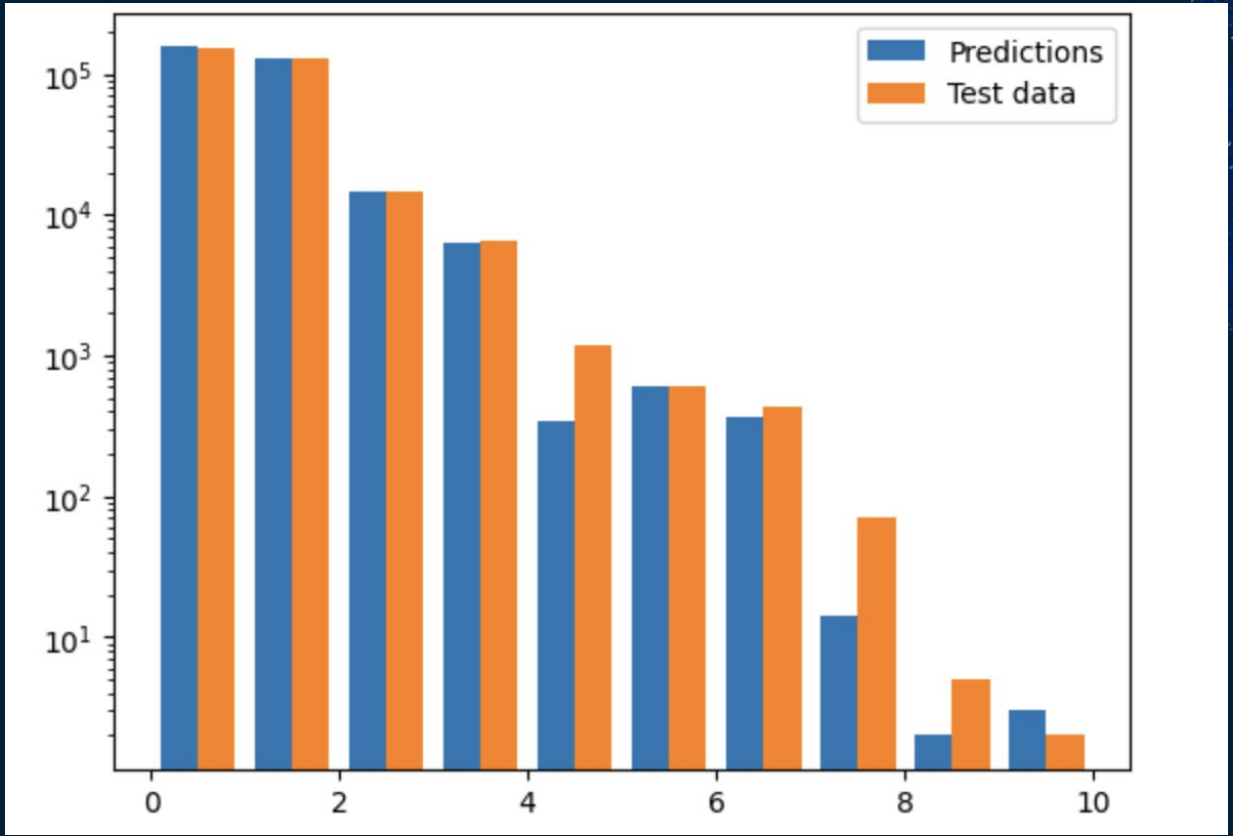
- Performance: Accuracy of 99.46% (ReLU Activation Function).
- Characteristics: ANNs are powerful models that can capture complex and non-linear relationships between features. They can be computationally intensive and require a lot of data to perform well.
- Analysis: The high accuracy suggests that the ANN was able to model the interactions between the cards very effectively. The network likely learned the complex patterns and dependencies between features that are critical for predicting poker hands.

Confusion matrix:

```
[[154101      3      0      0    868     14      0      0      0      0]
 [     6 129888    114      0     17      0      0      0      0      0]
 [     0      0  14457    337      0      0      0      0      0      0]
 [     0      0     78   6146      0      0     82     59      0      0]
 [     4     36      0      0    308      0      0      0      0      0]
 [     0      0      0      0      0    600      0      0      3      2]
 [     0      0      0      1      0      0    356      4      0      0]
 [     0      0      0      6      0      0      0      8      0      0]
 [     0      0      0      0      0      0      0      0      2      0]
 [     0      2      0      0      0      1      0      0      0      0]]
```

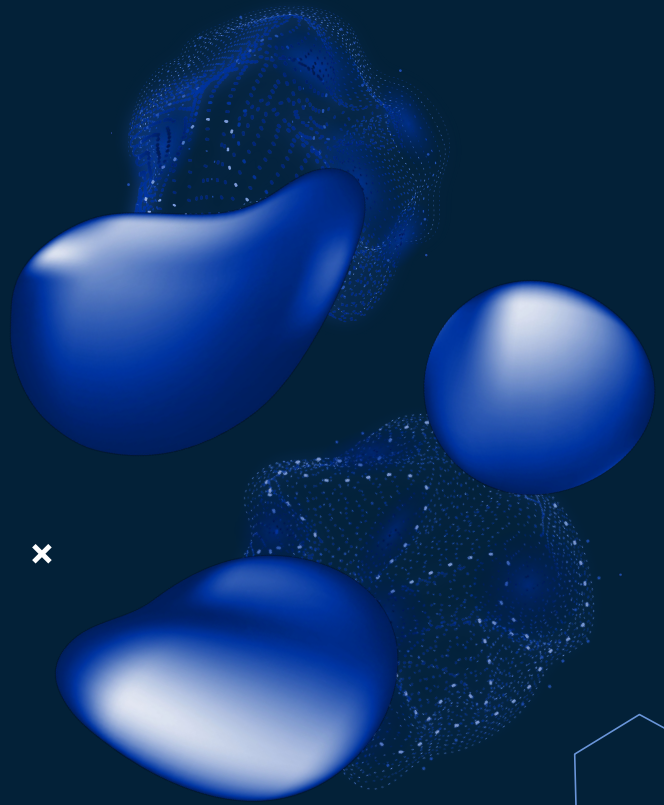


ANN Model



ANN Model Accuracy between classes.

What did we Learned ? (Conclusion)

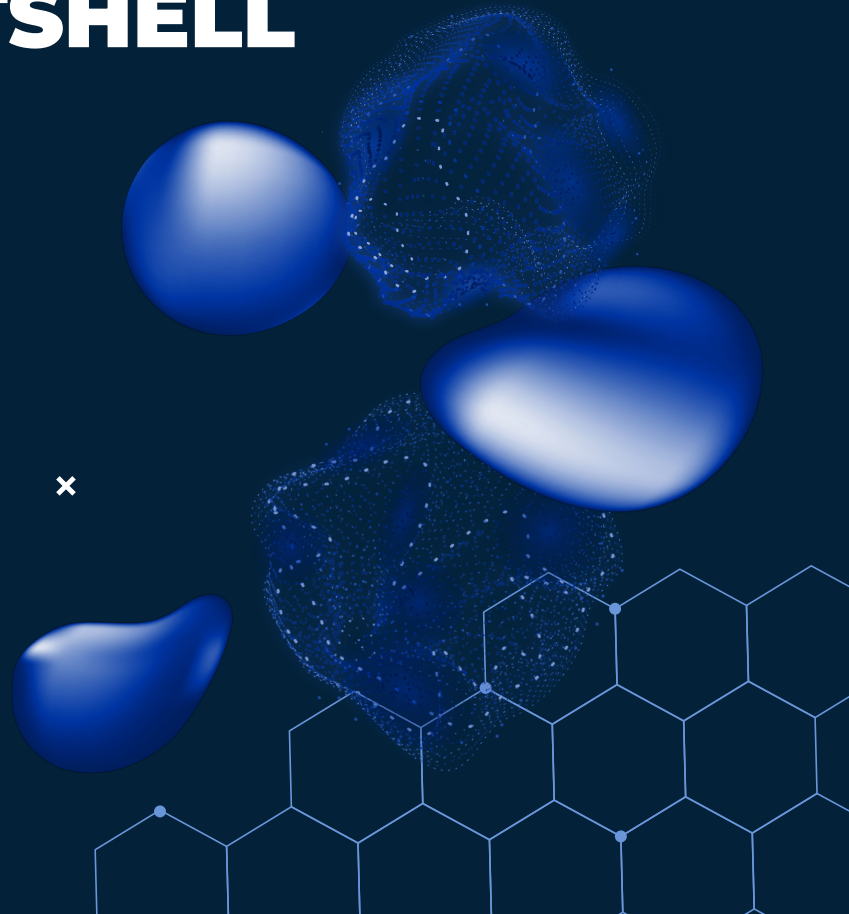


All IN A NUTSHELL

First and foremost how to handle imbalanced data.

Model Effectiveness: Among the models we tested, the ANN demonstrated high performance with an accuracy of 99.46%. We understand the capability of ANNs to model complex and nonlinear relationships effectively.

Learning Insights: The differences in model performance highlight the importance of choosing the right algorithm based on the specific characteristics of the data. It also points to the necessity of considering feature interactions in predictive modeling, especially in complex scenarios like poker hand prediction.



Scalable ?

Distributed Storage

Spread data across multiple machines instead of storing on a single machine.

Parallel Processing

Use multiple machines together to speed up analysis and model training.

Live Accuracy

Using this model to scale to real time live accuracy indicator while poker is played.

- Understanding Concepts to handle Imbalanced data like Cross Validation, SMOTE, ADASYN, Boosting, Bagging ... and applying them to understand them.

THANKS!

DO YOU HAVE ANY QUESTIONS?