

Progress

- I have done research on chess and how modern techniques are used for chess including mini-max and alpha beta pruning, I now have a better understanding of how chess works.
- I also did research on what Naïve Bayes is and how it works as a classifier. I compiled a number of papers on the topic and had a read of some of these papers. To help with understanding I also implemented a version of Naïve Bayes to a dataset on breast cancer data.

Update Snippets

Using Machine Learning Algorithms to Predict Outcomes of Chess Games Using Player Data

https://repository.rit.edu/cgi/viewcontent.cgi?article=13036&context=theses

BayesChess: A computer chess program based on Bayesian networks q

https://tinyurl.com/2f9fuw9y

Predictive Modelling of a Chess Player's Style using Machine Learning

https://nhsjs.com/2024/predictive-modelling-of-a-chess-players-style-using-machine-learning/

Analyzing Positional Play in Chess using Machine Learning

https://tinyurl.com/2zuwvh92

Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid

https://cdn.aaai.org/KDD/1996/KDD96-033.pdf

Machine Learning Algorithms to Predict Chess960 Result & Develop Opening

https://arxiv.org/pdf/2310.18938

```
def naive bayes categorical(df, X, Y):
features = list(df.columns)[:-1]
prior = calculate_prior(df, Y)
Y pred = []
 for x in X:
    labels = sorted(list(df[Y].unique()))
    likelihood = [1]*len(labels)
    for j in range(len(labels)):
         for i in range(len(features)):
             likelihood[j] *= calculate likelihood categorical(df, features[i], x[i], Y, labels
    post_prob = [1]*len(labels)
    for j in range(len(labels)):
         post_prob[j] = likelihood[j] * prior[j]
    Y pred.append(np.argmax(post prob))
 return np.array(Y pred)
```

Next Steps

Now I need to find a good chess dataset to use for Naïve Bayes

Then I need to preprocess this data to make it fit for Naïve Bayes

Then use this data to make a Naïve Bayes Classifier.