Soccer Game Prediction

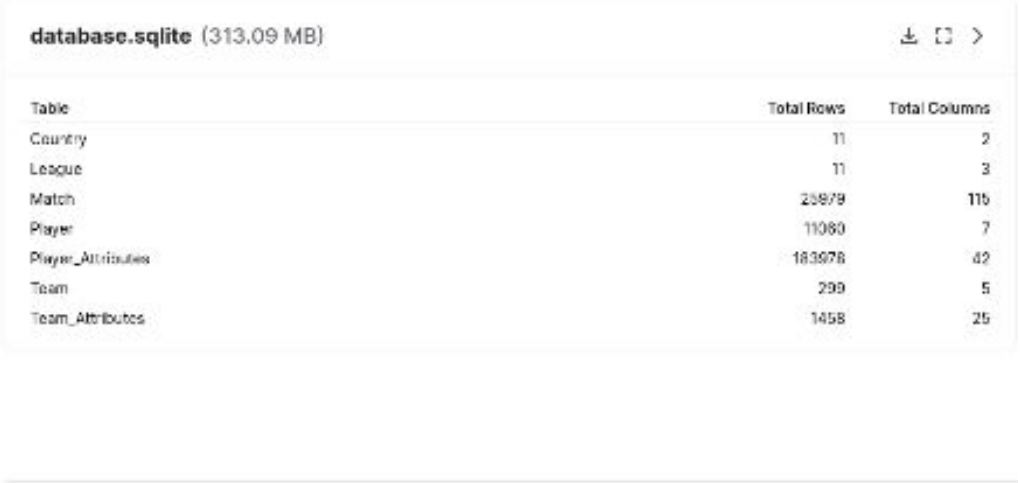
Chloe Atchabahian
Anson Lee

Introduction

- Goal: Predict outcome of a soccer match!
- Database from kaggle.com
  - 25000 matches
  - Data on each player and team
  - Due to missing data, 4000 of these matches were pruned from the final dataset.
- Logistic regression
- SVM
- Neural network.

Data Preprocessing

- Matches Table
  - Score
  - Date
  - Every player on home and away teams
- Stats on each player
  - FIFA stats
  - Birthday
  - Weight/height
- Feature Vectors
  - Team attributes
  - Every player Attributes
- Y values
  - one-hot encoding of the winner
    - Home win: [1, 0, 0]
    - Away win: [0, 0, 1]
    - Draw: [0, 1, 0]
- Benchmark: Bookkeeper odds
  - 10 different bookkeepers
  - 53% accurate

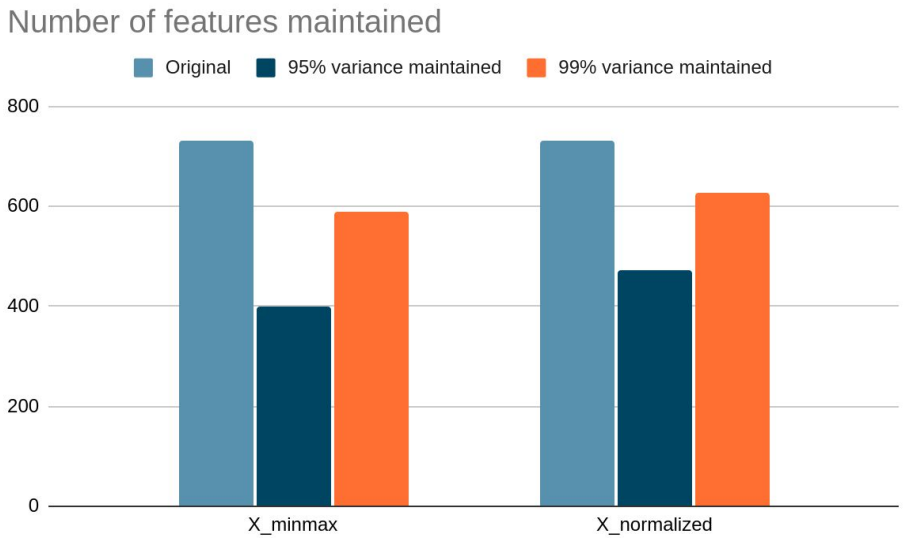


database.sqlite (313.09 MB)

Table	Total Rows	Total Columns
Country	11	2
League	11	3
Match	25979	115
Player	11060	7
Player_Attributes	183978	42
Team	299	5
Team_Attributes	1458	25

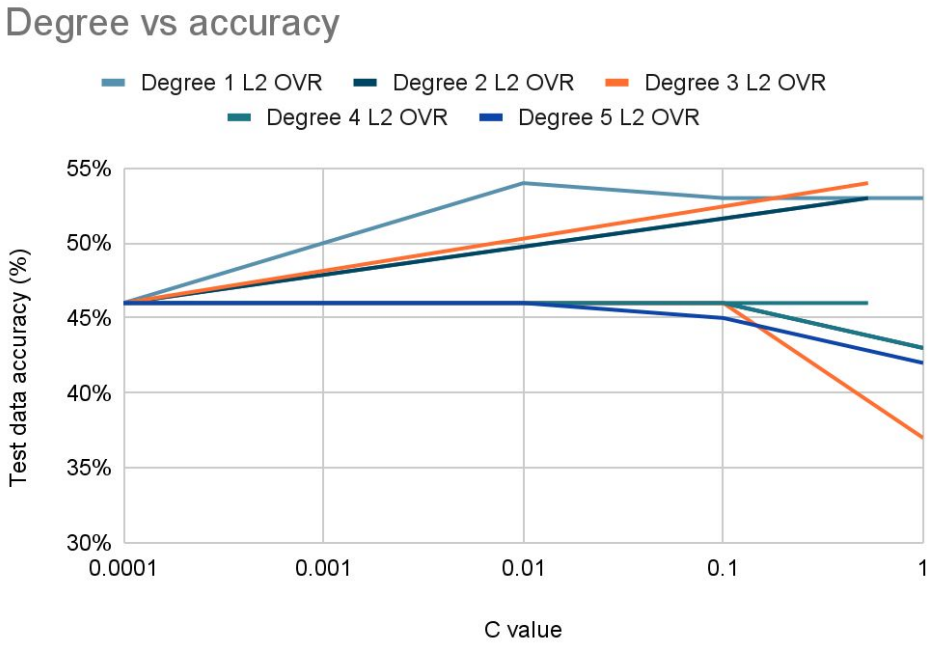
PCA

- Large number of features initially
- Principal Component Analysis (PCA) model was used to reduce the dimension of the features and increase the speed of learning.
- 730 features in original feature space to
  - 399 features in the minmax X matrix
  - 472 features in normalized X matrix
  - 95% of the variance maintained!

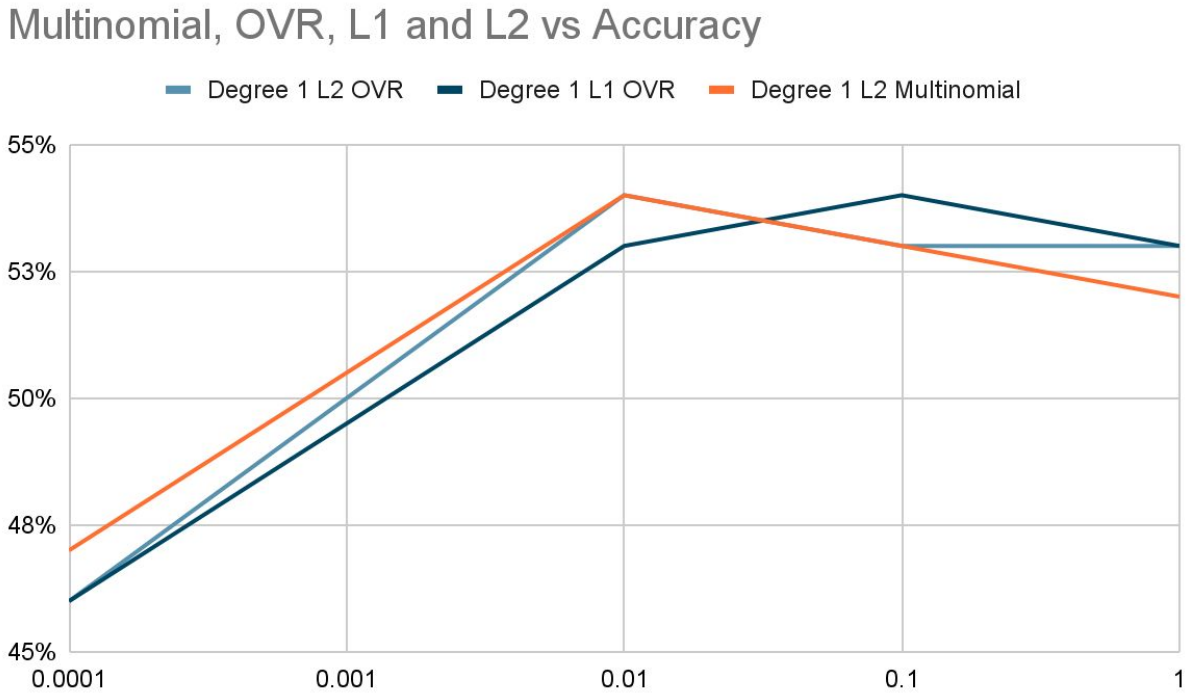


Logistic Regression

- Attempts on regularization
- Multinomial algorithm was used instead of the original one-versus-all method;
- Polynomial feature transformation was also attempted,
  - Nystroem method for kernel approximation

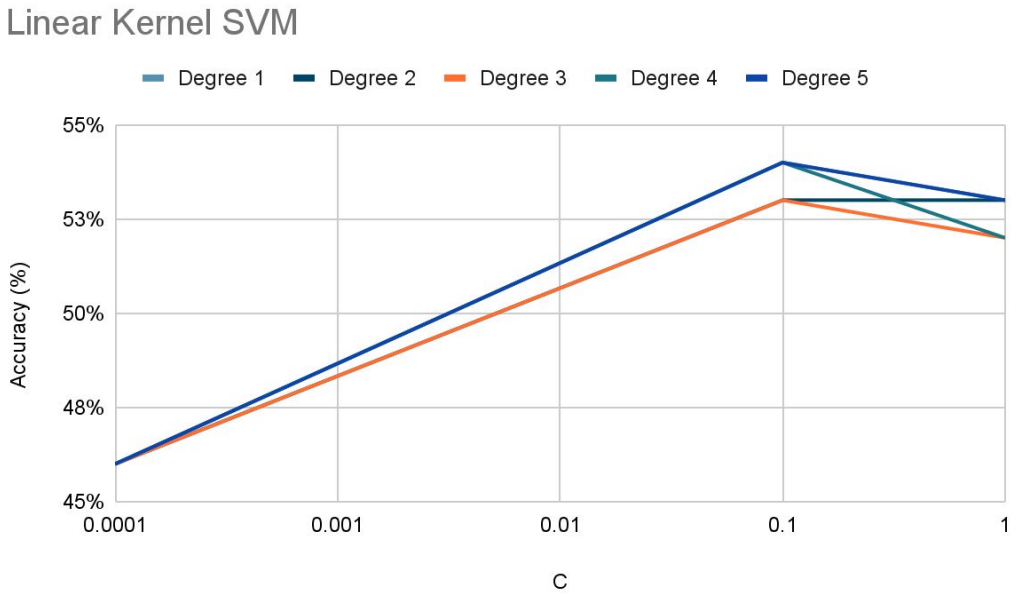


Logistic Regression

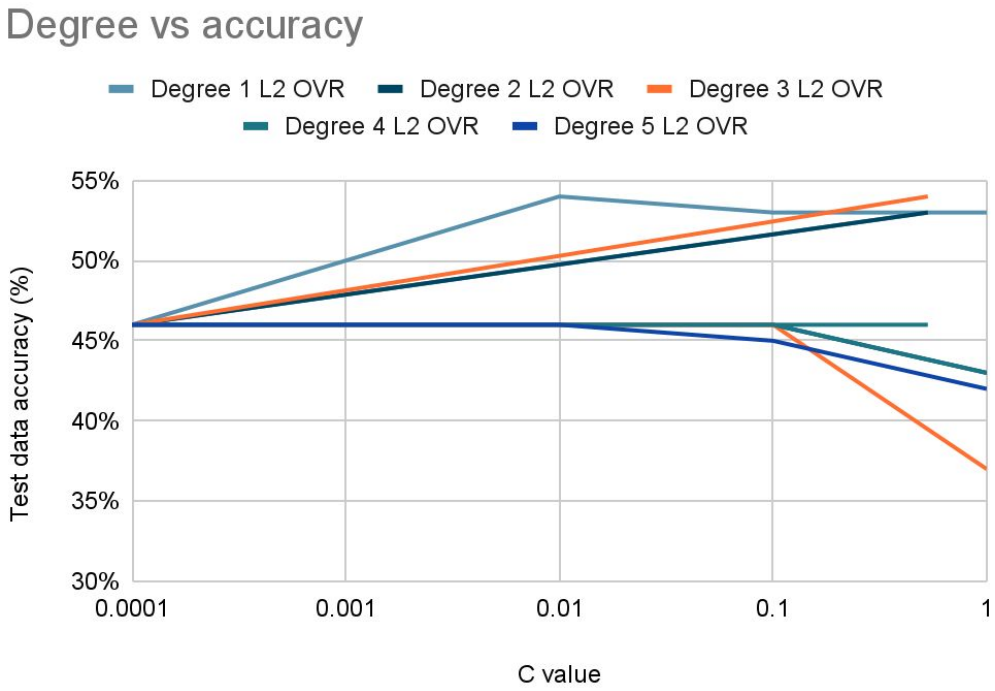


Support Vector Machines

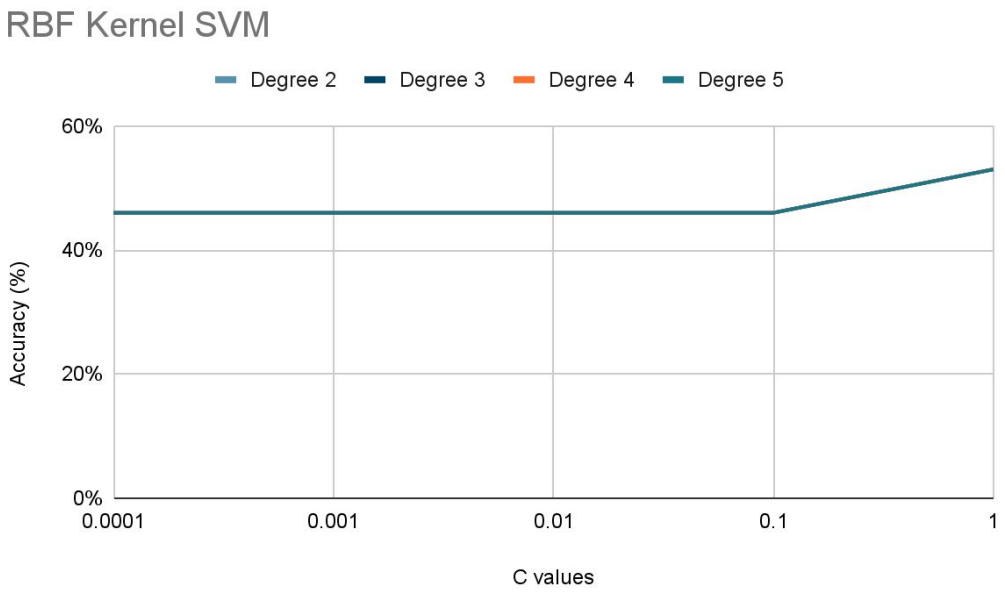
- Regularization
- C > 1 is observed to run significantly longer
- Kernel SVM's to capture non-linear patterns by transforming the feature vector.



Support Vector Machines

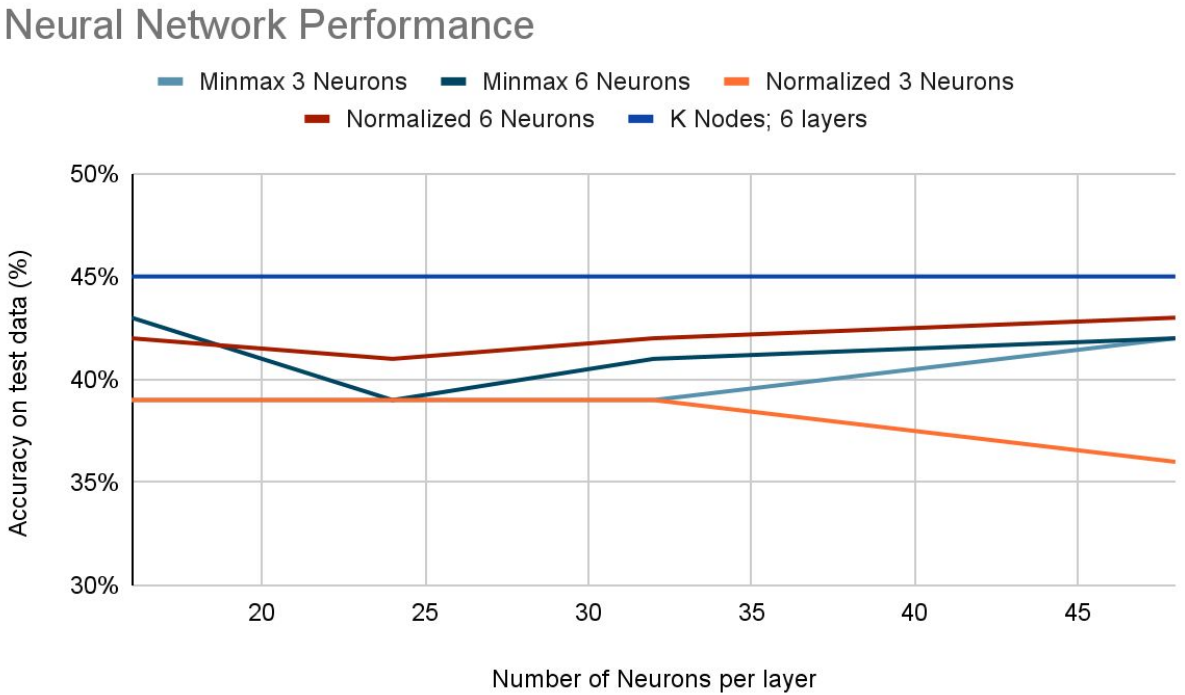


Support Vector Machines



Neural Networks

- Many different structures tried
- Average accuracy was 30-40%
- Best performance was with K nodes
  - Overfitting
- Not enough data for this to work.



Analysis of Results

- 2 of 3 Algorithms worked. Below are best results
  - Logistic Regression:
    - Degree 1; C = 0.01; 53% accurate
    - Degree 2; C= 1; 53% accurate
    - Degree 3; C=1; 54% accurate
  - SVM:
    - Linear Kernel; C=0.01; 53-54% accurate
    - Polynomial Kernel; C=1; 52% accurate
    - RBF Kernel; C=1; 53% accurate
  - Neural Networks
    - K neurons; 6 hidden layers; 45% accurate
    - 48 neurons; 6 hidden layers; 43% accurate

Conclusion

- Imbalanced f-scores for Logistic and SVM models
- Larger C for polynomial and RBF kernel can be helpful
  - Risk running much slower
- Success for logistic regression and SVM
  - both reached the 53% threshold.
- Failure for neural networks
  - Many different combinations of layers and nodes were tried, but none could cross 50% accuracy.