Data Classification Assignment

(a) Problem Statement:
Can we develop a predictive model for forecasting the outcomes of International Football/Soccer games based on historical data?
The objective is to to predict the outcome of football matches beforehand by using suitable data about the Countries/Teams that are playing each other.

(b) Data Sources (Data size: 4729)

I) Matches dataset from 2006-2018 (Team Country Names, Match Date, Competition Name, Goals scored by both teams) [1]

II) Squad dataset of individual competitions (Country, Coach Name, Player Name, Position, Caps) [1]

Other Data Sources

I) FIFA rating (Team Country Names, Match Date, Competition Name, Goals scored by both teams) [2]

II) Year-wise Population of Countries (Country, Population from 1960-2022) [3]

III) Year-wise GDP of Countries (Country, GDP from 1960-2021) [4]

IV) Year-wise average years of schooling in Countries (Country, Continent, ISO Code, avg years of schooling from 1990-2021) [5]

(c) Preprocessing

i) Within data collection: While combining the multiple data sources, did data cleaning and formatting. Additionally, replaced missing "Average Age" attribute with average and "Average Rating" attribute with minimum.

ii) Missing values: The data has roughly 700 missing values for Country data i.e. population, GDP etc. Replaces them with the average value.

iii) Outlier Detection: Filtered data by removing outliers using COF. COF computes the local density deviation of a given data point wrt its neighbors. It considers as outliers the samples that have a much lower density compared to their neighbors.

iv) Normalization: This is to ensure all features of the dataset having different ranges and scaled to a specific uniform range.

v) Discretize: Binning of numerical data into bins represented as classes. Here, I divided the years of the matches into bins of 4.

vi) Encoding: Converting nominal data into numeric by assigning sequential numbers for some features and one-hot encoding for some features.

(d) Modelling

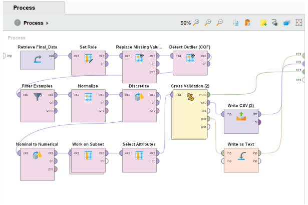
i) KNN is a distance based model that assigns labels to new data samples based on the classes of its K-nearest data points.
While KNN is flexible in-terms of its classification boundary and has strong explainability, it is a very expensive algorithm. The distance computation during inference and storage requirements are quite high compared to some other algorithms.
Main hyperparams: K, distance metric

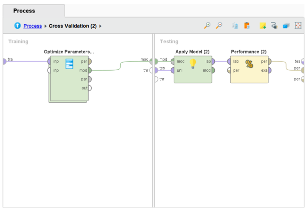
ii) SVM tries to find a plane that separates classes optimally such that the margin between the bordering data points i.e. the support vectors is maximized.
With the help of different kernels, SVM can adapt to linear and non-linear data distributions. While SVM is good at dealing with high-dimensional data, it is not ideal for very-large or noisy datasets.
Main hyperparams: kernel

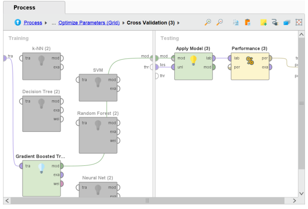
iii) Decision trees are hierarchical models that parses through different possibilities of one/more attributes to finally lead to one of target labels. The nodes are added to the tree based on a metric that evaluates the best possible split for the dataset.
DTs are convenient to interpret and robust to non-linear data, but they can also be open to over-fitting and unstability.
Main hyperparams: depth, samples required for split, metric

iv) Random forest is simply a combination or ensemble of multiple decision trees. The final classification is based on any voting strategy applied to the classes predicted by each decision tree.
Other than the pros of decision trees, random forest also helps prevent the over-fitting problem in DTs. However, it is relatively less interpretable and can be very computationally intensive.
Main hyperparams: no. of trees, depth, metric, voting strategy

v) Gradient boosting is also an algorithm that combines smaller models to improve the final classification performance. However instead of combining the trees during prediction like in random







forest, gradient boosting combines trees earlier one-at-a-time. Gradient boosting generally improves performance and is better at dealing with complex data. Gradient boosting with a large number of trees would require a lot of computation.
Main hyperparams: no. of trees, depth, learning rate for gradient

vi) Neural networks are intuitively a set of nodes that process data iteratively, such that the nodes are improved through gradient optimization. They are modeled on the human brain and its neurons.
Neural networks are great at handling a wide range of data linear/non-linear. They can deal with a large amount of data and noisy data as well. While the inference in neural networks is quick, the training can be resource intensive. Especially if the data is very complex or large in size. Moreover, it can be difficult to choose the right parameter values to ensure optimal performance and to avoid overfitting.
Main hyperparams: layers, learning rate, momentum

Model	Score (%)
KNN (scratch)	39.55
KNN	52.21
SVM	53.53
DT	50.46
RF	55.92
GB	**56.69**
NN	52.70

Table 1: Comparing results of different classification models

(e) The two algorithms used for running Grid Search are Gradient Boosting and Neural Networks. The two hyper-parameters for each algorithm are: No. of trees (5, 10, 15, 20) and Maximum depth (5, 10, 15, 20) for Gradient Boosting 2, Learning rate (0.001, 0.01, 0.1) and Momentum (0.2, 0.5, 0.9) for Neural networks 3.

No. of trees	Maximum depth	Score (%)
5	5	55.12
5	10	53.68
5	15	52.83
5	20	52.88
10	5	54.90
10	10	53.96
10	15	53.49
10	20	53.28
15	5	55.26
15	10	54.91
15	15	53.89
15	20	53.47
20	**5**	**55.35**
20	10	55.33
20	15	54.65
20	20	53.40

Table 2: Results of Grid search on Gradient Boosting over the following parameters: No. of trees, Maximum depth

Learning rate	Momentum	Score (%)
0.001	0.2	47.81
0.001	0.5	47.81
0.001	0.9	51.13
0.01	0.2	50.84
0.01	0.5	50.94
0.01	0.9	51.94
0.1	0.2	51.88
0.1	**0.5**	**52.09**
0.1	0.9	51.24

Table 3: Results of Grid search on Neural Networks over the following parameters: Learning rate, Momentum

(f) Conclusion

The best performing models are the ensembled decision tree based algorithms, i.e. Random Forest and Gradient Boosting. These two algorithms perform roughly 3% better than other algorithms. However, it is important to note that the final accuracy score for all models is in the range $50-56\%$. This indicates that this approach fails at predicting the outcome of a football match. A likely diagnosis is that the data considered for the task is insufficient. Other than the use of Countries demographic data and the players age & Fifa rating, we may also need to consider other factors such as news, twitter sentiment surrounding the team, coaching staff etc.

While Gradient Boosting achieves the best performance, all models seem to be limited to a 50% accuracy, leading to the conclusion that the problem requires more data and feature engineering.

(g) Github Repository: `https://github.com/surya1701/PA-Data-Classification`

References

[1] `https://github.com/jfjelstul/worldcup`

[2] `https://sofifa.com/`

[3] `https://worldpopulationreview.com/`

[4] `https://data.worldbank.org/indicator/SP.POP.TOTL?end=2020&start=2005`

[5] `https://globaldatalab.org/`