Regression Problem: FIFA

Start Assignment

- Due 23 Jun by 23:59
- Points 30
- Submitting a website url
- Available 31 May at 18:00 - 24 Jun at 23:59

FIFA

In sports prediction, large numbers of factors including the historical performance of the teams, results of matches, and data on players, have to be accounted for to help different stakeholders understand the odds of winning or losing.

In this project, you are tasked to build a model(s) that predict a player's overall rating given the player's profile.

Use the data set *males legacy(csv)* for training the model(s) and the data set *players_22* for testing/evaluating the model(s): The datasets can be found under Module 5 >> FIFA Datasets.

- **FIFA Dataset:** FIFA complete player dataset is a collection of detailed attributes for every player registered in the latest edition of the FIFA database. The description of the dataset is given **here (https://www.kaggle.com/datasets/stefanoleone992/fifa-23-complete-player-dataset)** . However, you are encouraged to download a copy of the dataset given on Canvas.

1. Demonstrate the data preparation & feature extraction process [5]
2. Create feature subsets that show maximum correlation with the dependent variable. [5]
3. Create and train a suitable machine learning model with cross-validation that can predict a player's rating. [5]
4. Measure the model's performance and fine-tune it as a process of optimization. [5]
5. Use the data from another season(***players_22***) which was not used during the training to test how good is the model. [5]
6. Deploy the model on a simple web page using either (Heroku, Streamlite, or Flask) and upload a video that shows how the model performs on the web page/site. [5]

- **NB:**
  - **Make sure you train at least 3 models, evaluate them and decide on a suitable model for deployment. Your best model is expected to be an ensemble model.**
  - **Ensure that your web page allows the user to input new data and displays both the**

player rating and the confidence score of the model.

SUBMISSION: You are to submit a *Github URL to your repository*

- **Create a GitHub Repository and name as FULLNAME._SportsPrediction**
- **Push / Upload the following files on your repository:**
  - **Google Colab / Jupyter Notebook named as FULLNAME*._SportsPrediction.ipynb***
  - **The Python version of your Colab / Jupyter Notebook**
  - **Source code(s) for the model deployment.**
  - **Video demonstrating how your web page works**

Some rubric (2) (1)

Criteria	Ratings					Pts
Data Preprocessing	**5 to >3.0 Pts Excellent** Demonstrated removing useless varaiables, EDA, imputation and encoding.	**3 to >2.0 Pts Good** Demonstrated removing useless varaiables, EDA, imputation No encoding.	**2 to >1.0 Pts Fair** Demonstrated removing useless varaiables, EDA, No imputation No encoding.	**1 to >0.0 Pts Poor** Demonstrated removing useless varaiables, No EDA, No imputation No encoding.	**0 Pts No marks** No Data Preprocessing	5 pts
Feature Engineering	**5 to >3.0 Pts Excellent** Created feature subsets which show better correlation with the overall rating and scaled the independent variables.	**3 to >2.0 Pts Good** Created feature subsets which have better performance with the overall rating and scaled the independent variables.	**2 to >1.0 Pts Fair** Created feature subsets which have better correlation with the overall rating but did not scale the independent variables.	**1 to >0.0 Pts Poor** Created feature subsets which have better performance with the overall rating and did not scale the independent variables.	**0 Pts No marks** No feature engineering done	5 pts
Training	**5 to >3.0 Pts**	**3 to >2.0 Pts**	**2 to >1.0 Pts**	**1 to >0.0 Pts**	**0 Pts**	

Models	**Excellent** Created and trained with cross-validation either RandomForest, XGBoost, Gradient Boost Regressors that can predict a player rating.	**Good** Created and trained with cross-validation non-Ensemble Regressors that can predict a player rating.	**Fair** Created and trained without cross-validation either RandomForest, XGBoost, Gradient Boost Regressors that can predict a player rating.	**Poor** Created and trained without cross-validation non-Ensemble Regressors that can predict a player rating.	**No marks** Trained useless models for the Job at hand.	5 pts
Evaluation	**5 to >3.0 Pts** **Excellent** Used MAE or RMSE and then fine tuned model, train and tested it again.	**3 to >2.0 Pts** **Good** Used MAE or RMSE and then fine tuned model, trained and but didn't tested it again.	**2 to >1.0 Pts** **Fair** Used MAE or RMSE and never fine tuned the model.	**1 to >0.0 Pts** **Poor** Used other metrics besides MAE and RMSE	**0 Pts** **No marks** No evaluation done	5 pts
Test with new data set	**5 to >3.0 Pts** **Excellent** Used the data(players_22) to test how good is the model with completely new data.		**3 to >0.0 Pts** **Good** Only used the split (test data) from players_21.		**0 Pts** **No marks** No testing done	5 pts
Deployment	**5 to >3.0 Pts** **Excellent** Deployed the model on a simple web page using either (Heroku, Streamlite or Flask) and shared a link to the video that shows how the model performs on the web page/site		**3 to >2.0 Pts** **Good** Deployed the model on a simple web page using either (Heroku, Streamlite or Flask) and but the video is not ONLINE	**2 to >0.0 Pts** **Fair** There is proof of code that deploys models but failed to do the actual deployment.	**0 Pts** **No marks** Nothing was done.	5 pts
						Total points: 30