Architectural Decision Document for FIFA Wages

1. Data preprocessing stage

   Goal: convert data from various types into usable data for ML models

   Step 1 dealing with various types:

   - Remove the players whose wage data is unavailable. Since the goal is to predict wage, adding necessary guesses at this stage is not preferred.
   - Remove or replace unnecessary symbols such as "€", "K", etc.
   - Transform data into integer type. For example, a height of a player could be 5'11". It is converted to integer (5×12+11=71).

   Reason:

   Players having 0 or missing wages are removed. Here we do not want to make any prediction or guesses on the wage since wage is the “label” we want to predict.
   Some of the data is not usable at all. An object like 5’11’’ cannot be used in models. All the columns can be converted to numbers (integers).

   Consequence:

   All columns converted to integer type

   Step 2 replacing missing values

   - For some attributes that could not be zero (e.g. Height), the missing values are replaced with the median of the existing set
   - For some attributes that could be zero, like players' scores (e.g. GKDiving, or gate keeper diving), the missing values are replaced by zero.

   Reason:

   Missing values are shown as NaN. They cannot be passed to a model. So, it is necessary to deal with them. Without removing too many rows, it is necessary to replace missing values with some numbers. For numbers that cannot be zero, like player height, the missing value is replaced by zero. Otherwise it is very weird. Other attributes like the player’s statistics (like free kicking) is replaced by zero

   Consequence:

   All missing values (except those in wages, which are already dropped) are replaced

   Step 3 Obtaining the statistical summary

   - Obtain the statistical summary

   Reason:

   Get an intuitive overview of the columns.

   Consequence:

   Observed that the columns all have different units

   Step 4 Remove highly correlated columns

   - Run a correlation table
   - List the attributes that have high correlation (>0.90)

   Reason:

Attributes would absorb the effect of another attribute that is highly correlated with it. So leaving only one of them is suffice to analyze and avoid affecting the analysis.
Consequence:
These attributes are removed: "BallControl", "SprintSpeed", "SlidingTackle", "Interception", "Marking", "GKHandling", "GKKicking", "GKPositioning", "GKReflexes".

Step 5 Split data into training/test set

- Split the around 67% of the data into training and 33% into testing

Reason:
The key to evaluate the performance of a model is to test how good it performs on an unseen data. A model may have high score on the seen data, but the reason might be overfitting. The training set is served as the data for training the model, whereas the test set is the "unseen" set.
Consequence:
Split the data into training and test set

Step 6 Rescaling the set

- Rescale the data to standard normal (mean zero and standard deviation 1)

Reason:
The attributes have different units. It is very difficult to analyze the effect of different units. For example, the effect is large on prediction because the scale of the attribute is large.
Consequence:
All columns rescaled to standard normal set.

Step 7 Introducing performance metric

- Introduce Mean Squared Error (MSE) as metric
- $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$

Reason:
The value to be predicted is a continuous variable, so metrics like accuracy cannot be used. Then MSE is an intuitive measure to reflect the difference between predicted value ($\widehat{Y}_i$) and actual value (Y_i). For some more advanced metric like AIC, it cannot be applied to some of the models used later on (AIC need the number of parameters estimated, whereas random forest does not have to estimate parameters).

2. Model construction and training
Goal: construct models and train them on the training set, obtain the MSE score on the test set.

Step 1 Select the models

- Select linear regression
- Select random forest regression

- Select neural network

Reason:
Linear regression works for linearly separable data. It is very easy to interpret. At the end, we want to conduct some feature importance analysis, so we need some models that could provide a straightforward interpretation
Random forest is a machine learning model that has some complexity. It works better with data that is not linearly separable. It is treated as a middle level model.
Neural network is a powerful model for prediction. It is treated as a high level model.

Step 2 Constructing and training each model

- Construct the model with various hyperparameters
- Train the model
- Obtain a MSE score on the test set

Reason:
Hyperparameter: the parameter that can be changed only before the training.
Different hyperparameter could lead to different models, especially for random forest. And there's no conclusive way on which combination works the best. So, testing on as much hyperparameters would be a good idea. After training the model, we use the MSE score to compare which set of hyperparameter works the best and which model algorithm works the best.

3. Model Evaluation
   Goal: determine which model is the best for prediction

   - Find the lowest MSE score on test set
   - Designate the model with the lowest MSE as the best model for prediction

   Reason:
   As discussed, MSE measures how the prediction is different from actual value. The model with the lowest MSE performs the best prediction on unseen data. So that model is best for prediction.
   Consequence:
   Designated the best prediction model (random forest)

4. Model Deployment
   Goal: obtain a real-world impact of this project

   - Discuss how wage prediction could be used
   - Evaluate the feature importance

   Reason:
   It is necessary to analyze the real-world impact of this project.
   Consequence:
   The project is useful for soccer club owners and managers.

They could use feature importance analysis result to quickly glance for the most affordable player for them.
They could use the best wage prediction model to predict the wage of a given candidate. Then they’ll get an advantage.