DEAN WAN - GA DATA SCIENCE

FINAL PROJECT

BEATING THE DROP

BEATING THE DROP

BUIL[illegible]G A SQUAD THAT COULD BEAT THE DROP ON MINIMAL [illegible]ONEY

- Scrape Web [illegible]ta from a horrible website.
- Create model that [illegible]s player stats [illegible]game stats.
- Remove collinearity with o[illegible]players and team strategy
- Predict how much eac[illegible]layer is w[illegible]th
- Predict game st[illegible]s from player stats
- Predict h[illegible]w many points a team can get off the[illegible]players

DEAN WAN - GA DATA SCIENCE

FINAL PROJECT

TRANSFER GM

HOW MUCH IS A PLAYER WORTH?

ART, SCIENCE AND SPECULATION

- Personnel recruitment is critical.
- How much should you pay for a player?
- What factors influence price?
- What exactly are you paying for?

HOW MUCH IS A PLAYER WORTH?

ART, SCIENCE AND SPECULATION

The Challenge:

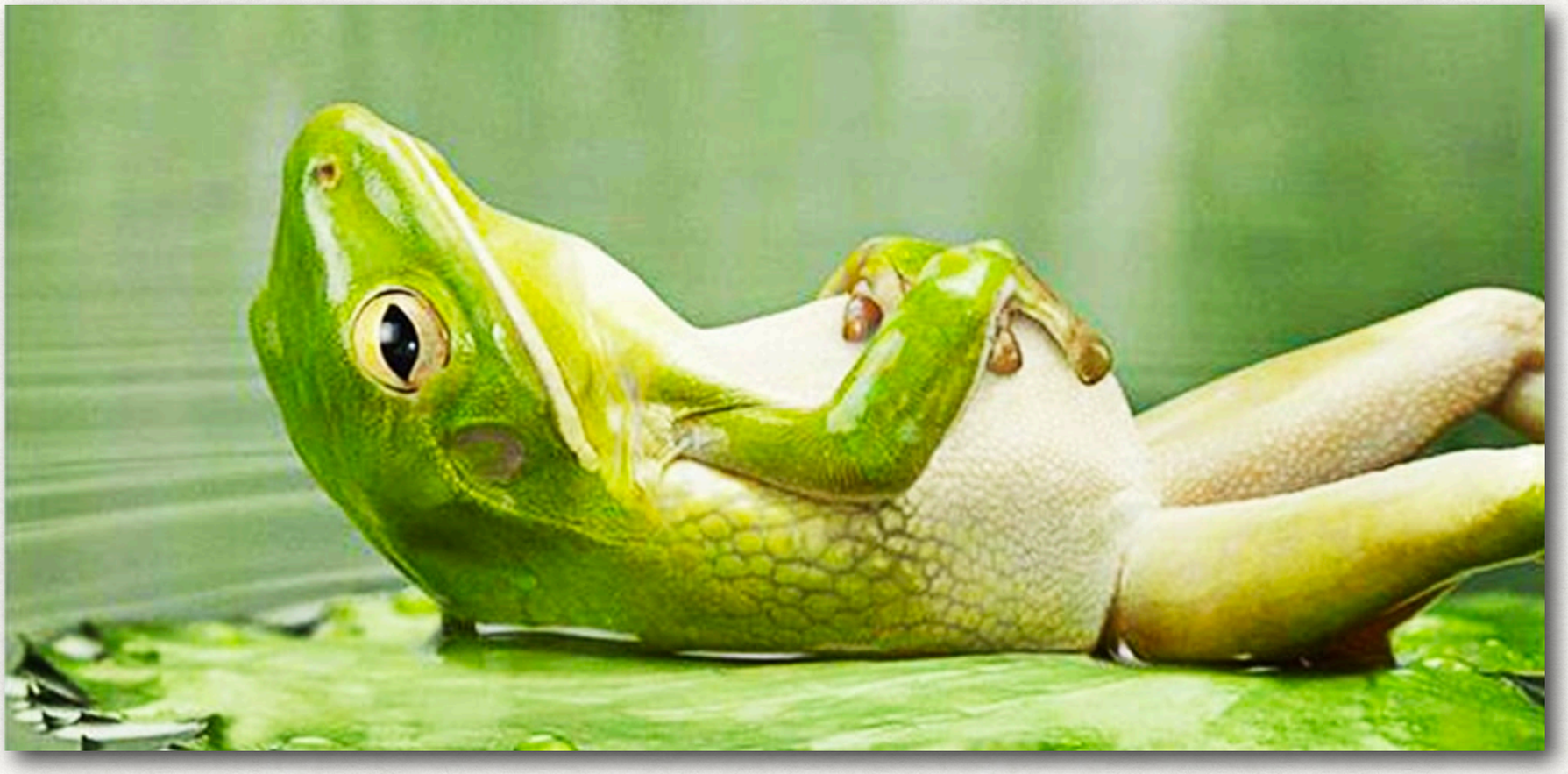
Given a player's ratings for a number of key skills at a given point in time, can we predict their market value?

The Hypothesis

Players with higher market values will have higher ratings for key statistics for their respective position - i.e. finishing for forwards, vision & passing for midfielders, tackling for defenders and reflexes for goalkeepers.

DATA COLLECTION

THE ART OF BEING ZEN



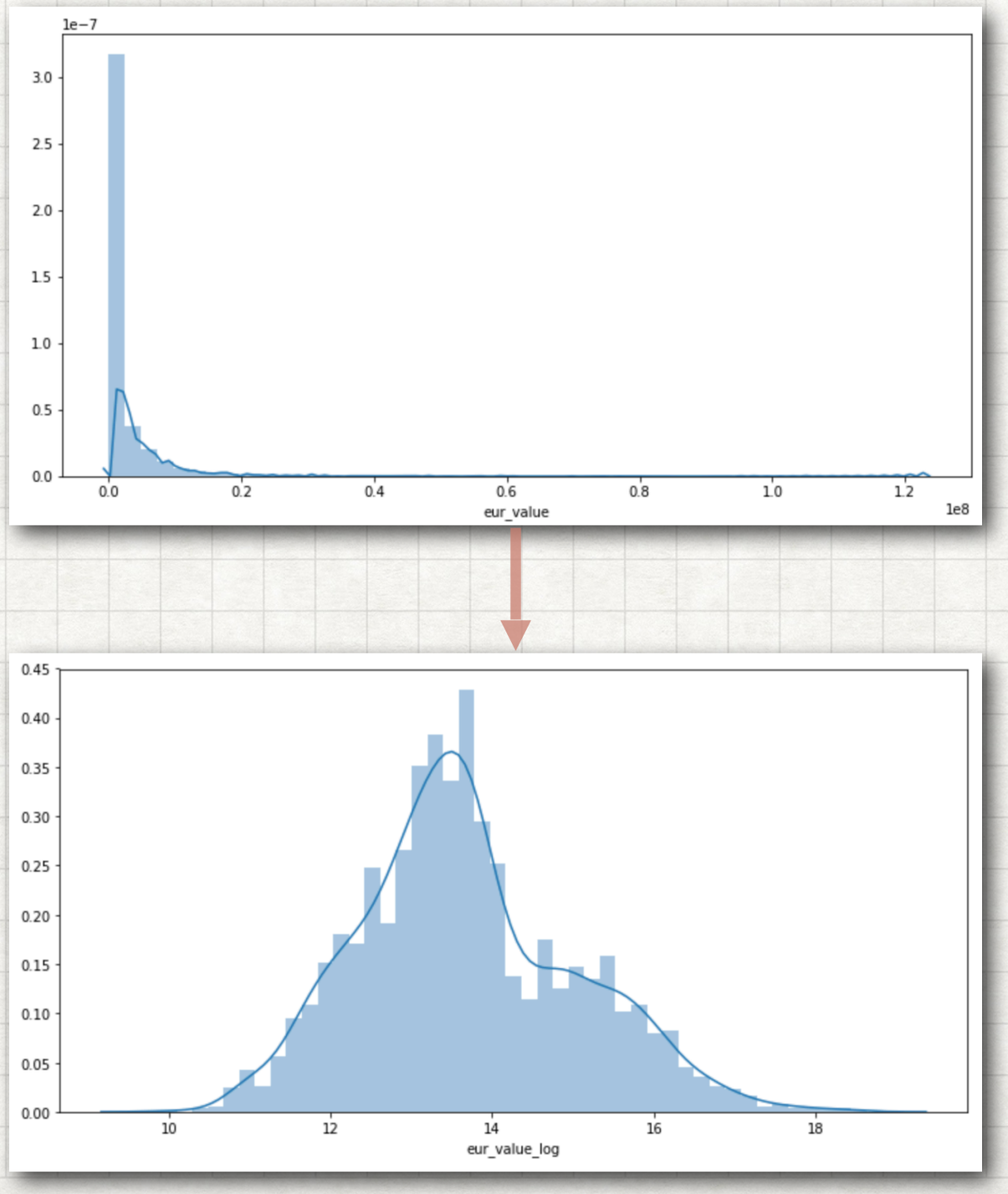
DATA CLEANING & WRANGLING

THE ART OF MINOR ANNOYANCES

DATA CLEANING AND WRANGLING

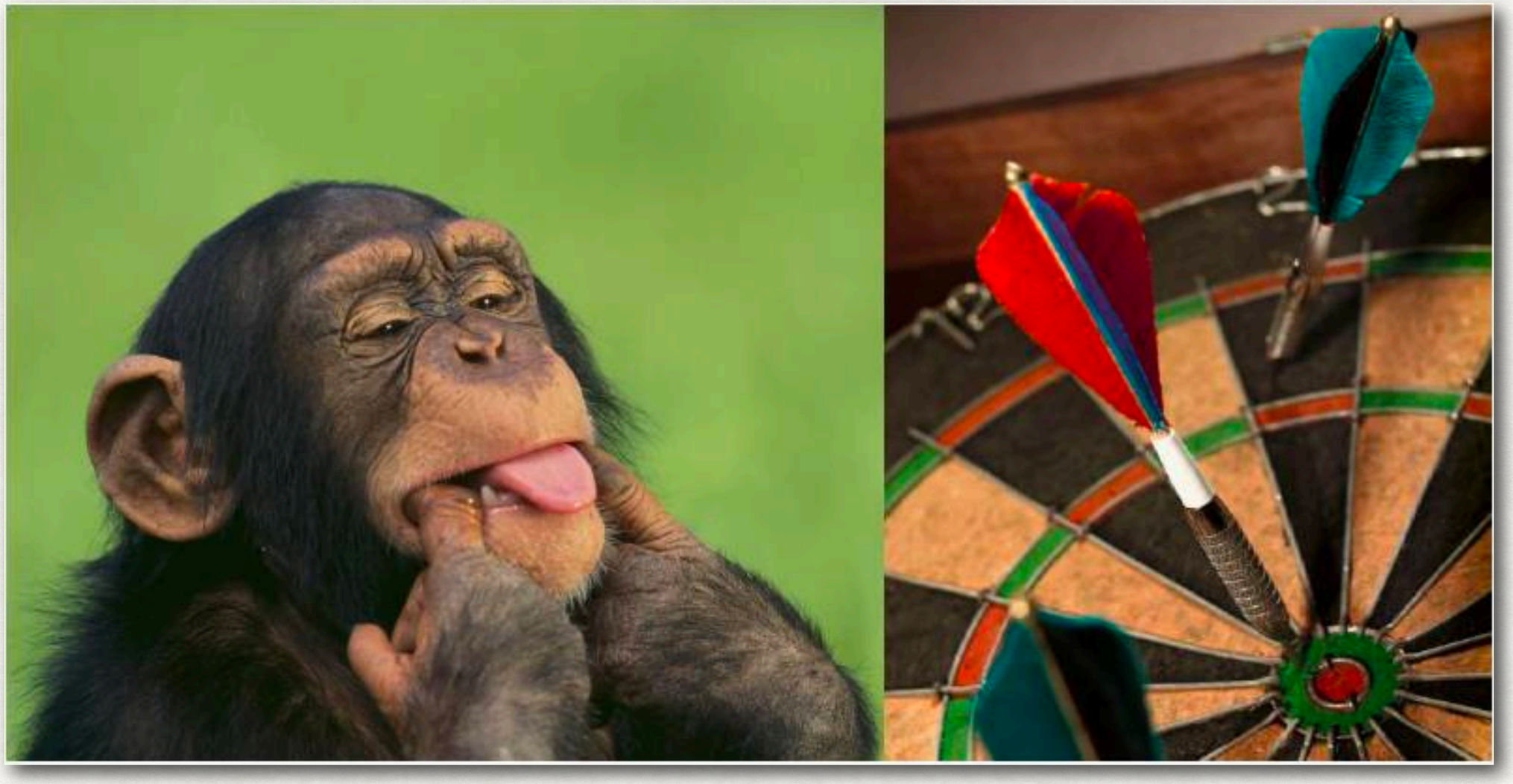
NIGGLES EVERYWHERE

- Looping through dataframes & applying ‘if’ logic.
- 40MB CSV - some preprocessing required.
- Python isn’t multilingual, encoding is a pain.
- Normalising skewed data
- Over 50 features per player



MODEL SELECTION

PICKING THE BEST MODEL FOR OUR NEEDS AND DATA



MODEL SELECTION

PICKING THE BEST MODEL FOR OUR NEEDS AND DATA

Problem Type:
Supervised learning regression problem (estimating a market value).

- Ordinary Least Squares Linear Regression (inc. Regularisation)
- Random Forest Regression
- Support Vector Regression

Success Criterion:
The 10-Fold Cross Validated average Mean Square Error (using R2 score where available as a secondary measure of success).

MODEL SELECTION

LINEAR REGRESSION

What is it?

Predicting an output based on input (independent) variables, assuming there is a linear relationship between the input and output (dependent) variable.

Procedure

1) Divide players into broad positional groups
2) Select the right features (domain knowledge, collinearity, RFE)
3) Plug these variables into a dataframe.
4) Train-Test Split
5) Apply LinearRegression() in scikit-learn
6) Assess MSE
7) Optimise via feature elimination/regularisation

MODEL SELECTION

LINEAR REGRESSION

<u>Optimisations</u>

> Feature Removal (Domain knowledge, collinearity, RFECV)

> Regularisation

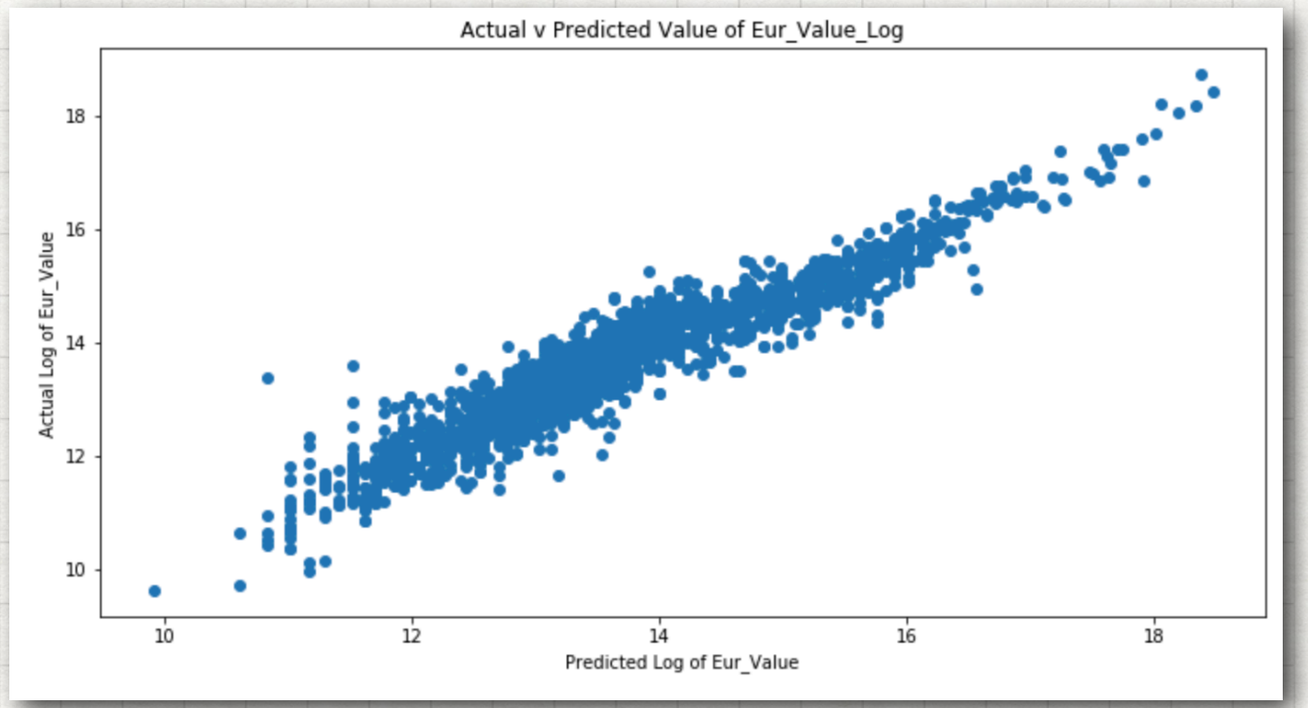
<u>Results</u>

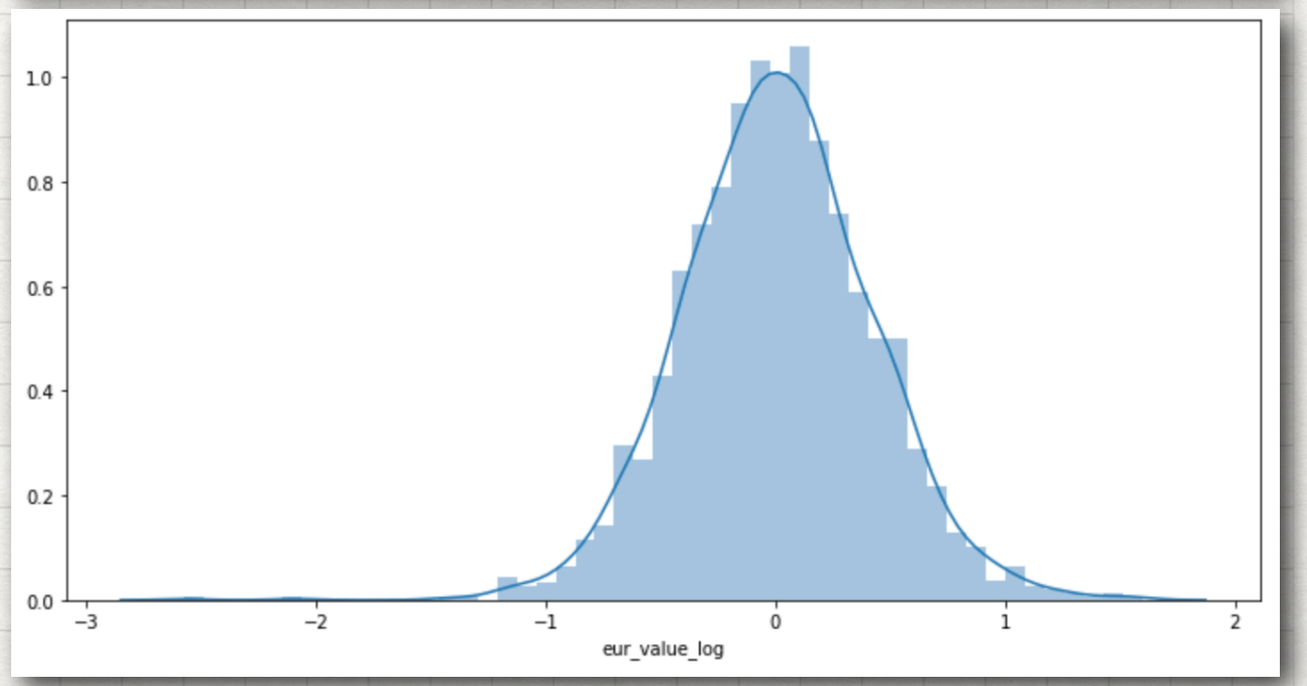
> The average 10-Fold CV MSE across the 4 different positional groups for base OLS between ***0.11 and 0.19.***

> Regularisation and RFE didn't make a notable impact on MSE, except for Goalkeepers!

<u>Conclusion</u>

Reasonable predictive power for an intuitively simple model.





MODEL SELECTION

RANDOM FOREST REGRESSION

What is it?

Random Forest Regression uses the principles of decision trees. The Random Forest creates many decision trees by bootstrapping and random feature selection to create a more accurate regression model.

Procedure

1. Break dataset up into positional groups
2. Train-Test Split
3. Run Random Forest Regression using default inputs
4. Optimise hyper-parameters

MODEL SELECTION

RANDOM FOREST REGRESSION

<u>Optimisations</u>

GridSearchCV the following:

> Number of trees for the Random Forest

> Number of features

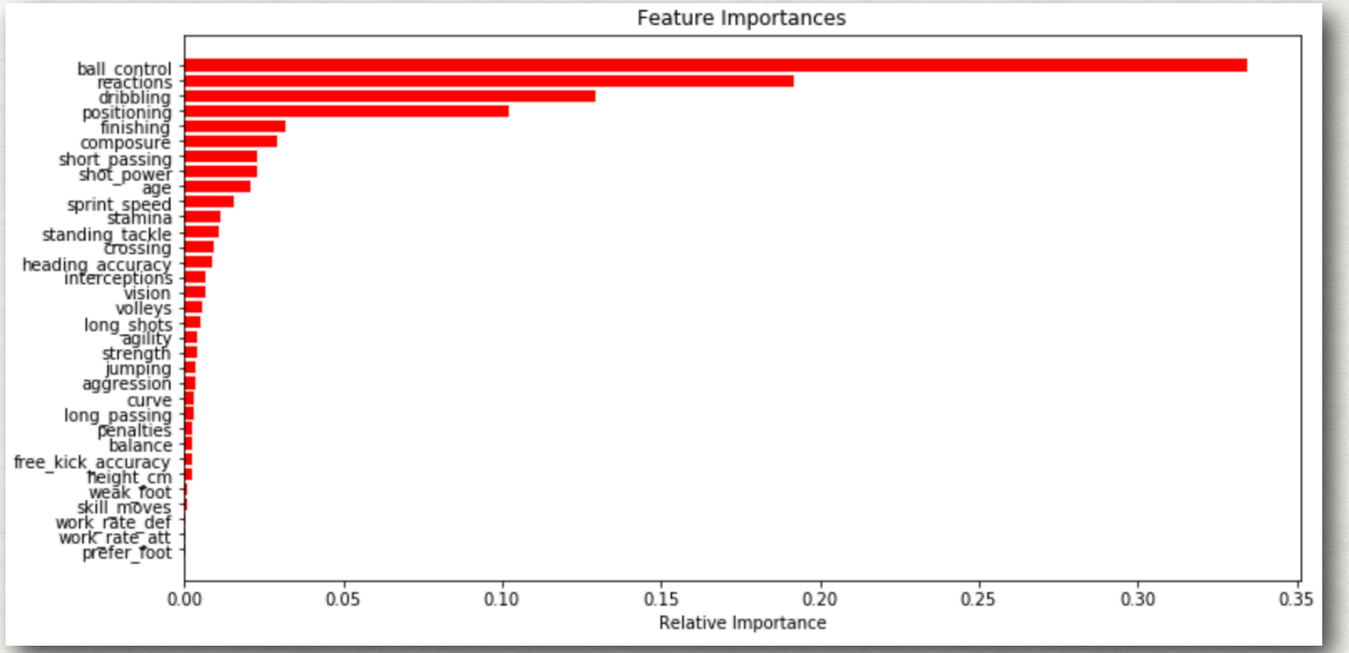
> Samples required for a leaf (i.e. end result)

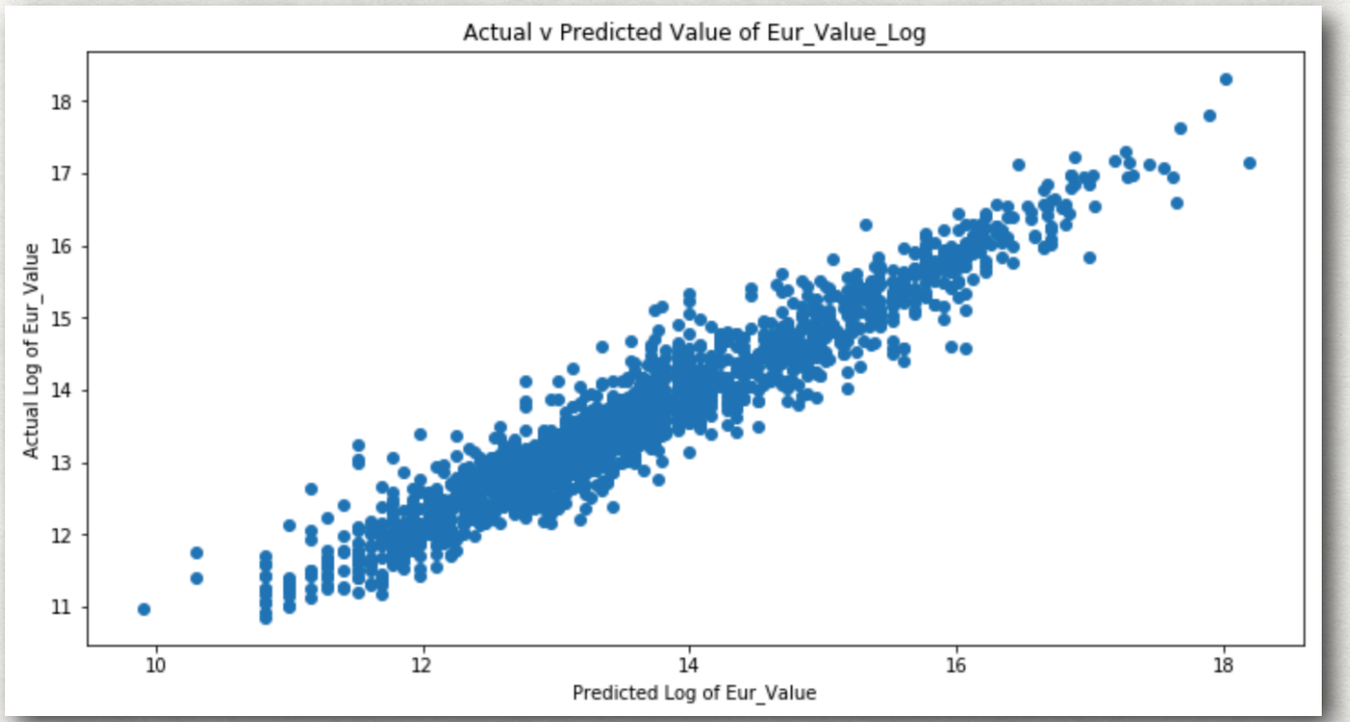
<u>Results</u>

Average 10-Fold CV MSE **~0.14,** a notable improvement on the optimised Linear Regression model

<u>Conclusion</u>

An improvement on the Linear Regression model.





MODEL SELECTION

SUPPORT VECTOR REGRESSION

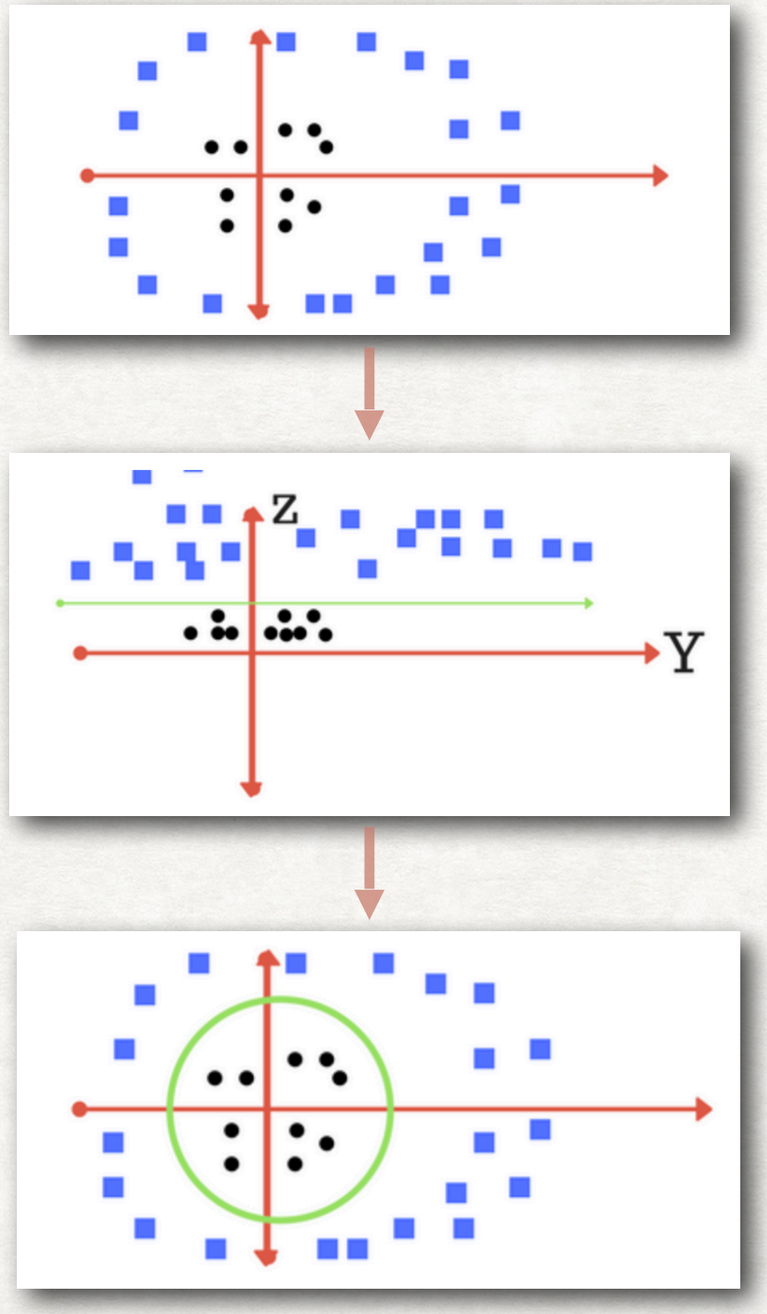
<u>What is it?</u>

We draw a line (hyperplane) that splits up data into groups (usually more dimensions)

We then transform the data back into the original planes.

<u>Procedure</u>

1. Break dataset up into positional groups
2. Normalise data via standard-scalar
3. Train-Test Split
4. Run SVR using default inputs, but vary the kernels
5. Optimise hyper-parameters



MODEL SELECTION

SUPPORT VECTOR REGRESSION

<u>Optimisations</u>

1) Kernel type - i.e. influences the way the transformation of the data occurs

2) Regularisation (via GridSearchCV)
   a) 'C' - cost of misclassification
   b) **γ** - Distance of points considered
   c) **ϵ** - The 'margin'

<u>Results</u>

Significant improvement in 10-Fold CV average MSE, between **0.06 and 0.105,** at the cost of explanatory power.

CONCLUSION

WHICH WAS THE BEST MODEL?

The Support Vector Regression model consistently provided the lowest 10-Fold CV average MSE:

> 0.064 for Forwards (RBF Kernel)
> 0.09 for Midfielders (RBF Kernel)
> 0.08 for Defenders (RBF Kernel)
> 0.105 for Goalkeepers (Linear Kernel)

The tradeoff is explanatory power.

To be fully complete, build a positional classifier.