Surf Condition Forecasting

Habiba Kapasi

Department of Atmospheric and Oceanic Science, University of California Los Angeles

AOS C111: Introduction to Machine Learning for Physical Sciences

Dr. Alexander Lozinski

I. Introduction

Surf conditions depend on a mix of oceanographic factors and these measurements are constantly monitored to forecast how the waves will shape up. Most surfers regularly check surf-forecasting apps because not every swell produces rideable waves. Since surfing is such a big part of Southern California culture, and something I enjoy myself, I wanted to explore which machine learning models could best predict surfing conditions and which ocean variables play the biggest role in determining whether a day scores high or low.

II. Data

The dataset that I used is Kaggle's "SoCal Surf Forecast Apr-May 24" which was created by scraping Surfline, a well known forecasting site. It includes 16-day surf conditions along with oceanographic variables and location data for 16 different Southern California surf breaks. Measurements were recorded every three hours, from midnight to 9:00 p.m.

Before training any machine learning models, the dataset needed quite a bit of cleaning and preprocessing. I started by removing extra whitespace and line breaks from the column names to make them easier to reference. The dataset creator had already done some initial cleaning like stripping the "kts" unit from the wind column and creating a separate numeric version. There were a couple cases like this, so I removed the duplicates. I also dropped columns that weren't useful for prediction, such as "Probability %" which was a reflection of Surfline's internal model accuracy and would not add anything meaningful to my model.

Next, I sorted the data by date and time so the model could be trained on earlier observations and tested on later ones. The original dataset was sorted by location first, but ordering everything chronologically made more sense for forecasting as it represented a more realistic approach.

Handling the wave height range column required some extra work because Surfline reports values like "3-4 ft" or "4-5+ ft". The original creator had already split each range into a lower and upper height column. For values that included a "+" I added 0.5 ft to the upper bound to reflect a slightly higher estimate. I also created an average wave height column, which I thought might help the model capture a better general wave height.

After that, I one-hot encoded the categorical features like surf break, county, and break type. Since regression models can't take string values directly, this step allows them to be able to interpret the different locations. For the time variable, I used cyclical encoding

instead of keeping the hour as a number from 0 to 21. This was done to avoid inaccurate gaps, like the jump between the 0 hour (12:00 a.m.) and the 21st hour (9:00 p.m.) and allows the model to interpret time as a repeating, clock-like cycle. I also converted the date column to a proper datetime object and extracted the day of the year so the model could learn the seasonal patterns.

Finally, Surfline rates conditions using categories like Very Poor, Poor, Poor to Fair, Fair to Good, and Good. The dataset creator had already mapped these to integers from 1 to 5, and I kept that column as my target variable.

Before training the models, I wanted to do some exploratory data analysis to better understand the dataset. The first graph I created shows the distribution of the surf ratings and the percentage of observations for each rating. This is important because it reveals class imbalances. As seen in the figure below, ratings of 1 and 5 are quite rare, while 2, 3, and 4 occur more frequently. It makes sense that the models would be more likely to predict the more common ratings (2, 3, and 4) rather than the extremes.

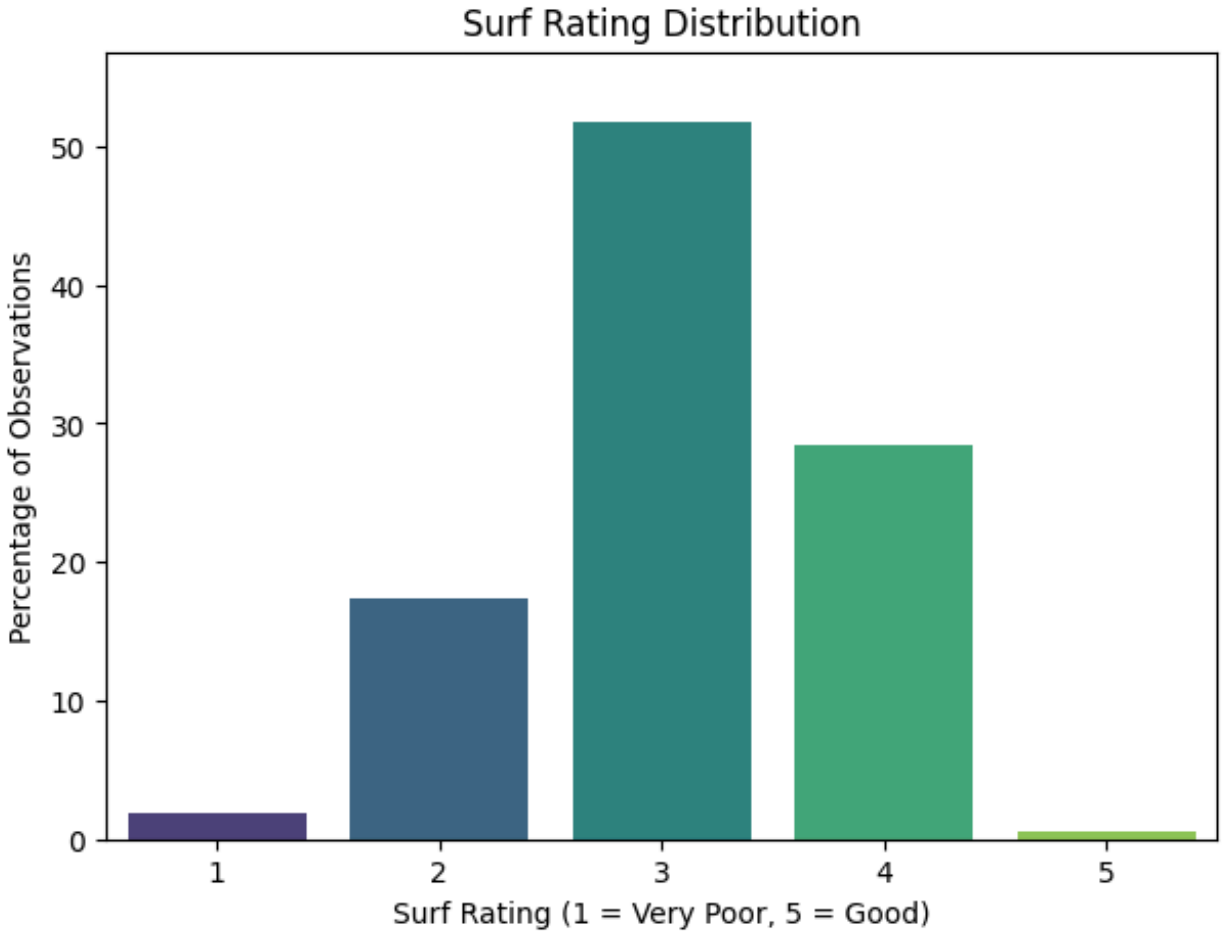


Figure 1: Distribution of surf ratings showing the percentage of observations in each category.

Next, I looked at how each oceanographic variable correlated with surf rating. I created a correlation heatmap and a bar plot to focus specifically on the relationships between surf rating and the other variables. From these figures, we can see that wind has the strongest negative correlation, meaning that as wind increases, surf ratings tend to decrease. The "Tim_sin" column has the highest positive correlation, while "Time_cos" is negatively correlated, suggesting that surf ratings generally increase as the day progresses. Similarly, the other correlations can be observed from the bar plot.

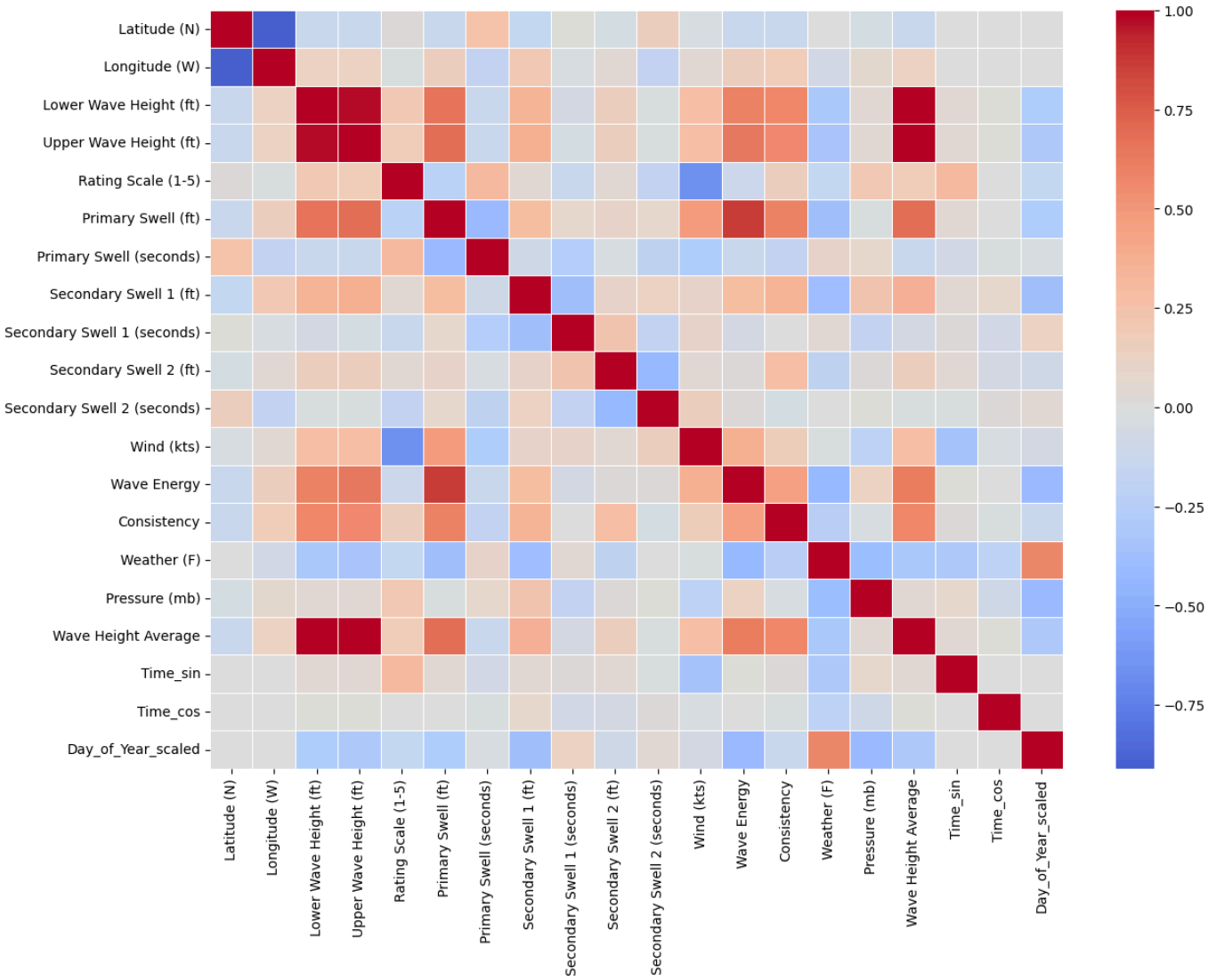


Figure 2: Correlation heatmap displaying correlation among all oceanographic variables in the dataset.

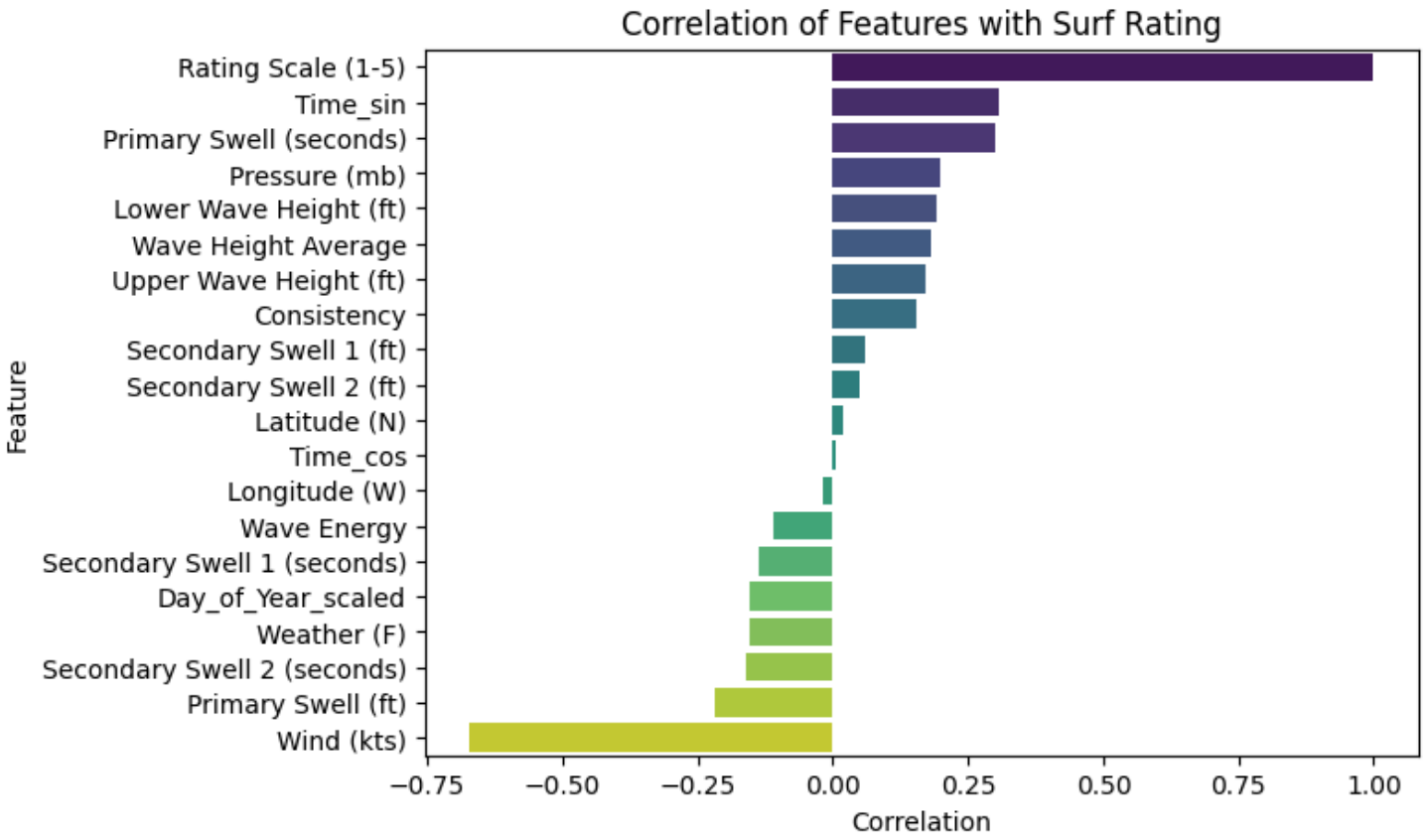


Figure 3: Bar plot showing each oceanographic variable's correlation with surf rating.

Finally, out of curiosity, I examined the average surf rating by time of day to see which hours tend to have the best conditions. As shown in the graph below, surf ratings peak around the ninth hour, dip afterward, and rise again near the twenty-first hour.

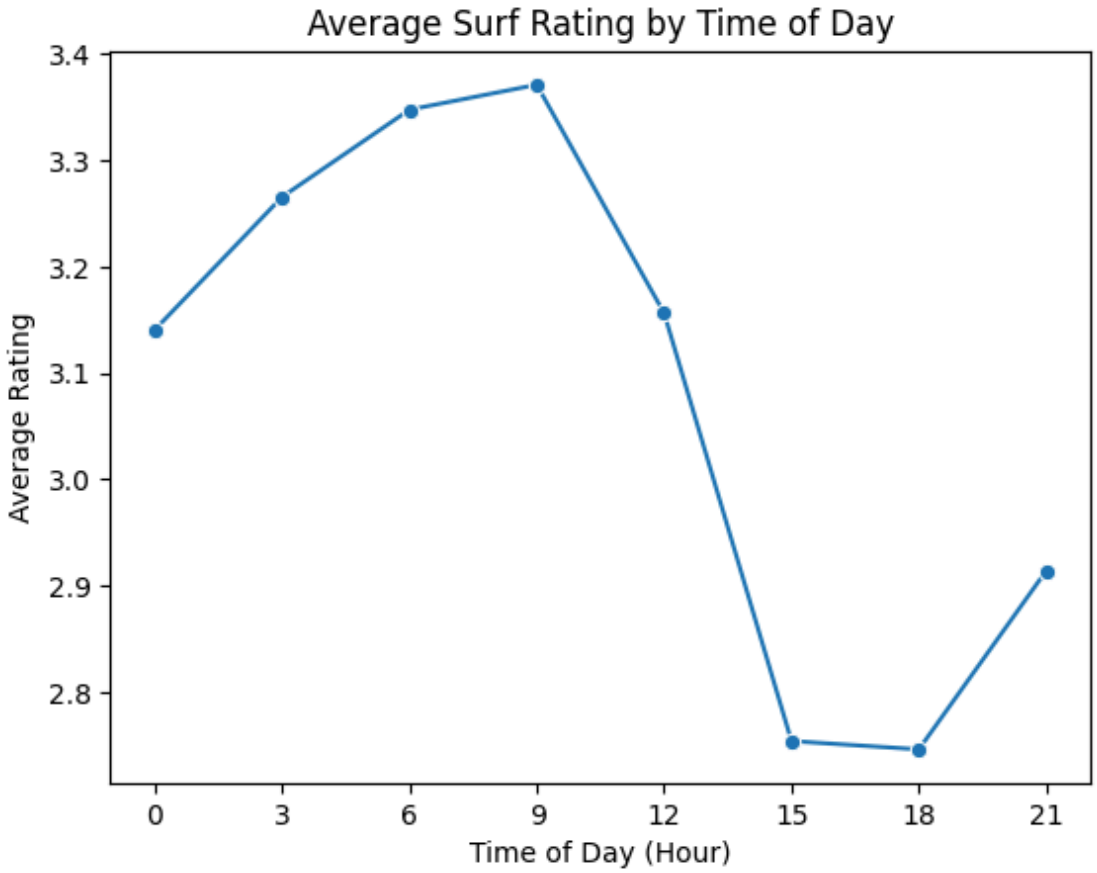


Figure 4: Average surf rating across the hours of the day.

III. Modeling

I decided to train three different machine learning models to determine which one could most accurately predict the surf rating. At first, I planned to use classification models, but I realized that the target variable is ordinal. For example, if the true surf rating is 5, a classification model would treat a prediction of 4 and a prediction of 1 equally wrong, even though predicting a 4 is much closer and more reasonable. Since regression models account for error distance, they made more sense for this problem. The three models I selected were Ridge Regression, Decision Tree Regressor, and Support Vector Regression.

I started by splitting my data into train and test categories. The test size I used is 0.2 so the models are trained on 80% of the data and tested on the other 20%. Since my data was chronological, I made sure it was split the same way each time instead of randomly. All the features columns were put into X_data, and the rating column (target variable) was assigned to y_data. The split was done using sklearn's train_test_split function.

The first model I trained was Ridge Regression. I used a pipeline that included scaling because Ridge can be sensitive to differences in magnitude, and using unscaled features could cause inaccurate coefficient values. I thought this would be an interesting model to use because it adds regularization which helps prevent overfitting. Since my dataset has many correlated variables, the regularization can help reduce the impact of less important features.

After training the model, I decided to look at two types of predictions, one as a continuous output, and another rounded to an integer from 1 to 5, similar to how the target variable is presented. I also wanted to look at a coefficient plot to see which features were weighted most heavily. As the figure below shows, the wave height values have the highest positive coefficients, which makes sense given how strongly wave size influences surf rating.

At first, I was confused about why the “primary swell (ft)” variable had a large negative coefficient. However, this is due to multicollinearity. The primary well height is strongly correlated with other wave height features (since swells are what create the waves) and this can be seen in the correlation heatmap above. The Ridge regularization adjusts the coefficients to balance out these overlapping relationships.

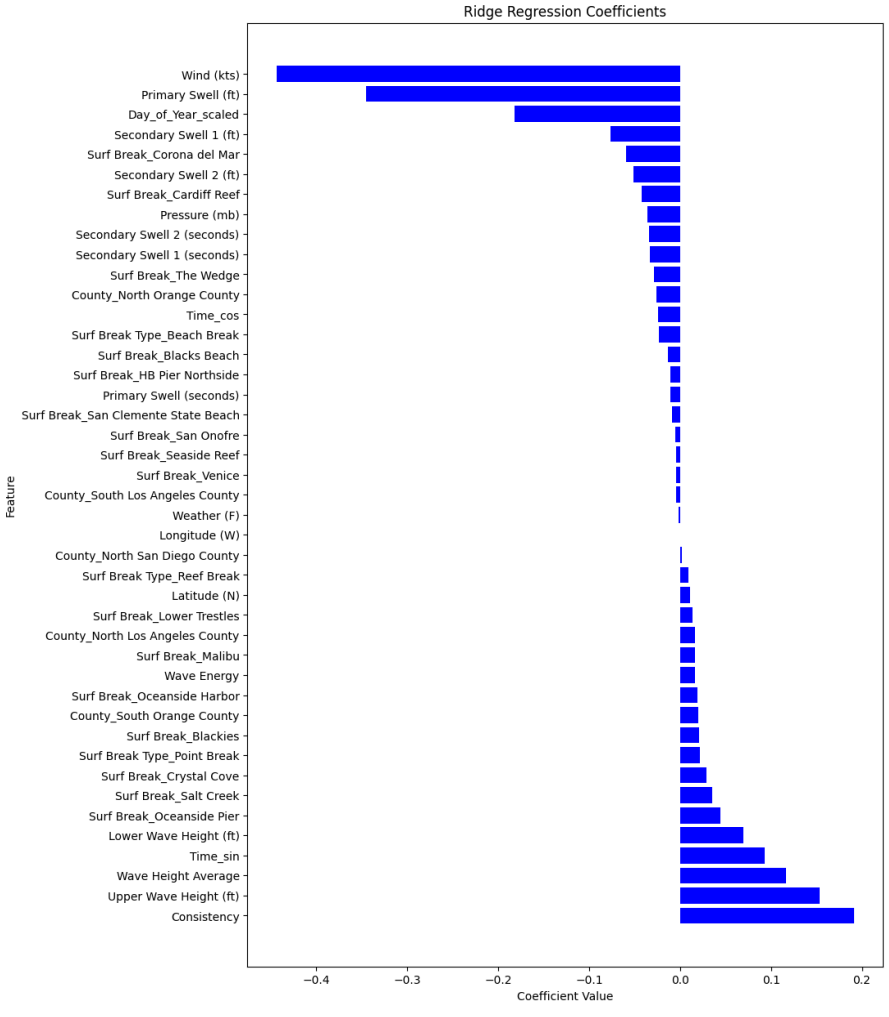


Figure 5: Graph showing Ridge Regression coefficients for all features in the dataset.

The second model I trained was the Decision Tree Regressor. Unlike Ridge and SVR, this model does not require scaling. The Decision Tree Regressor works by splitting the data into groups so that each group is as similar as possible in terms of the target variable.. I trained the model and predicted both the raw and rounded values. After experimenting with different maximum depths, a depth of 5 seemed to give the lowest errors, as shallower or deeper trees tended to increase the error metrics.

I also wanted to create a feature importance graph to see which features the tree was using for its "best splits". As shown in the figure below, wind and average wave height have two of the highest feature importance values. This indicates that these features contribute most to the models decisions and heavily influence the predicted surf rating. I also found it interesting that the model picked up on seasonal changes, with the day of the year feature coming third in importance, reflecting the general trend of higher surf ratings earlier in the year.

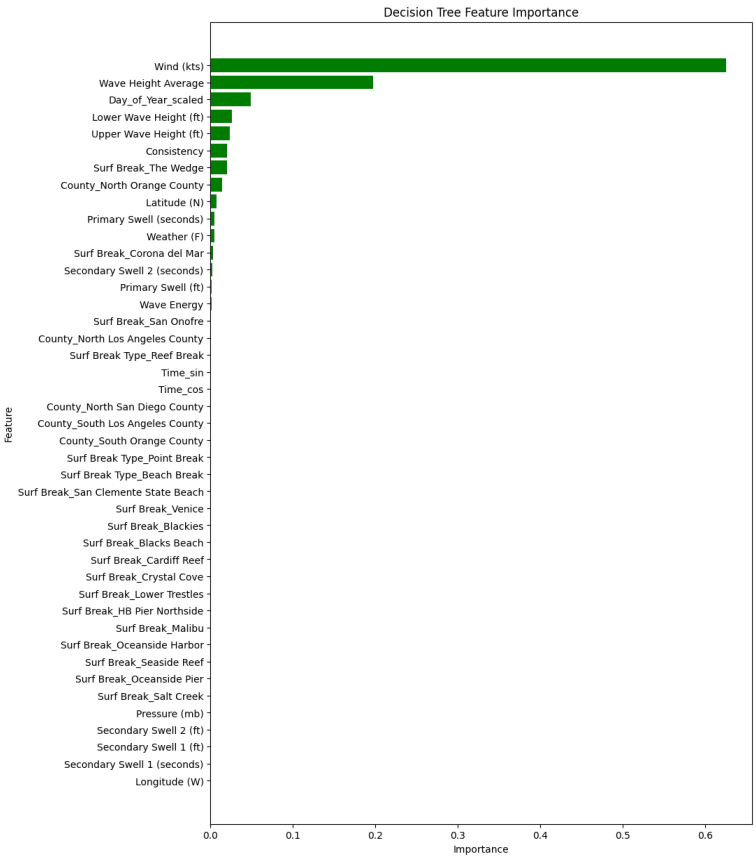


Figure 6: Graph displaying feature importance for Decision Tree Regressor.

The last model I trained was the Support Vector Regression Model (SVR). Like the Ridge model, I used a pipeline to scale the data. SVR works by trying to find a function that predicts the target variable within a margin of tolerance (epsilon) and does not penalize predictions which fall inside this tolerance. Because it is well suited for noisy datasets, I thought it would be good to try it on this data. I decided to use the RBF kernel for this model as it can capture the non linear relationships present in the dataset.

Just like the last two models, I predicted both the raw and rounded outputs. I also wanted to check the percentage of predictions landing outside the epsilon margin to see if the epsilon value needed adjustment. I initially started with an epsilon of 0.1 and found that 85% of my predictions were falling outside the margin. After increasing it to 0.5, only 22% of predictions were outside the margins, and the error metrics also improved. Needing a larger epsilon suggests the data has more small noise, and increasing it allows SVR to ignore this noise and focus on the general trend rather than overfitting.

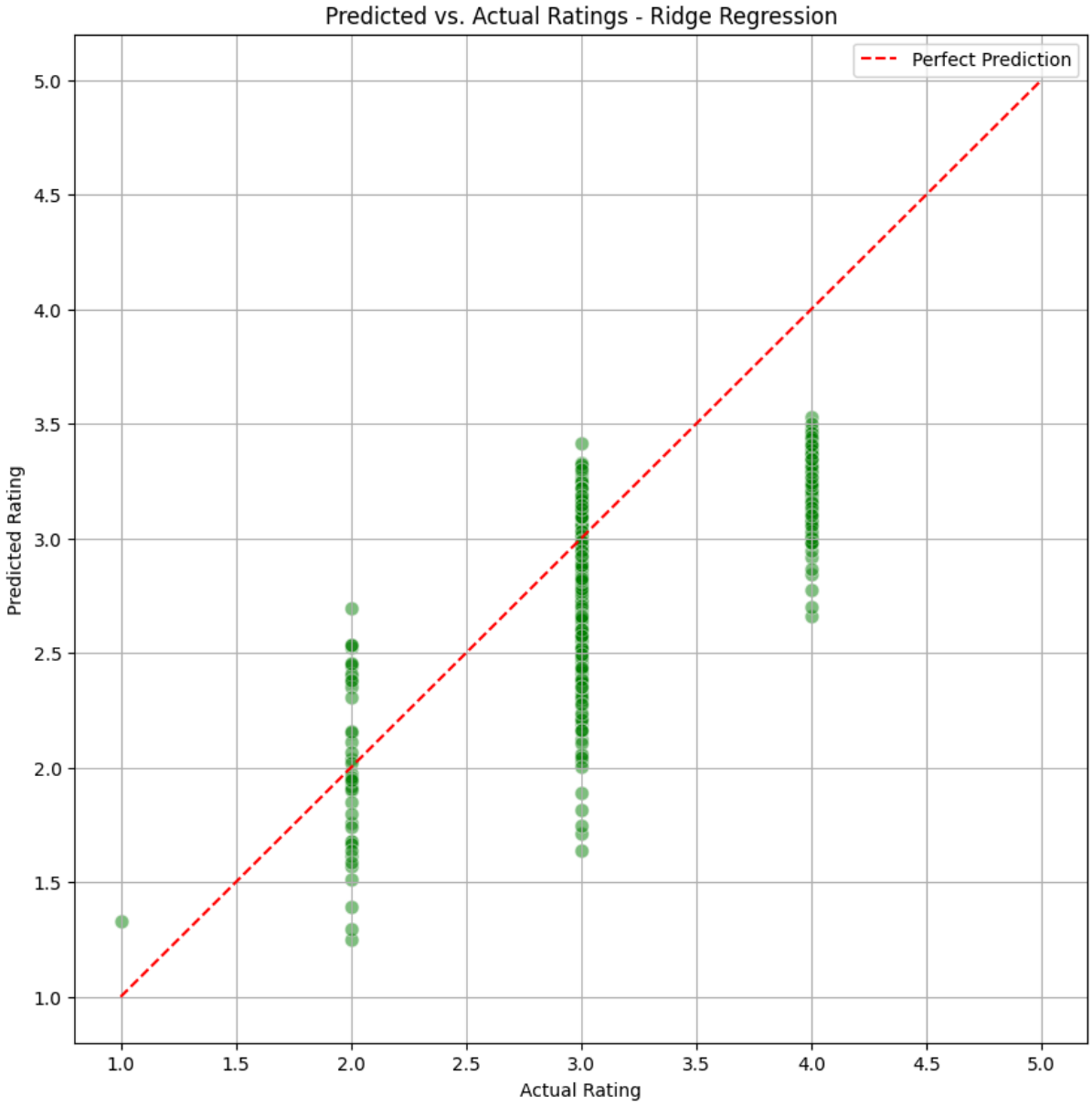
IV. Model Analysis

To determine which model performed the best, I looked at two error metrics. I calculated the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). RMSE is sensitive to outliers because of the squaring step, giving more weight to large errors, while MAE treats all the errors equally. Below, I have created a table summarizing the values for all three models.

	RMSE (raw)	RMSE (rounded)	MAE (raw)	MAE (rounded)
Ridge Regression	0.50671	0.58435	0.39912	0.34146
Decision Tree Regressor	0.40183	0.43617	0.27503	0.19024
Support Vector Regression	0.40333	0.47112	0.33260	0.22195

Based on these values, I would say that Decision Tree Regressor proves to be the best model in accurately predicting the surf rating score. SVR performs similarly, but Decision Tree has slightly lower errors, while the Ridge model performs the worst. After reviewing these metrics, I wanted to compare the predicted versus actual ratings. I have included the graphs below showing the raw predicted values, but the points overlap, making it difficult to clearly see the differences. To get a clearer picture, I calculated the count of all actual and predicted values using the rounded predictions and have presented the results in the tables below.

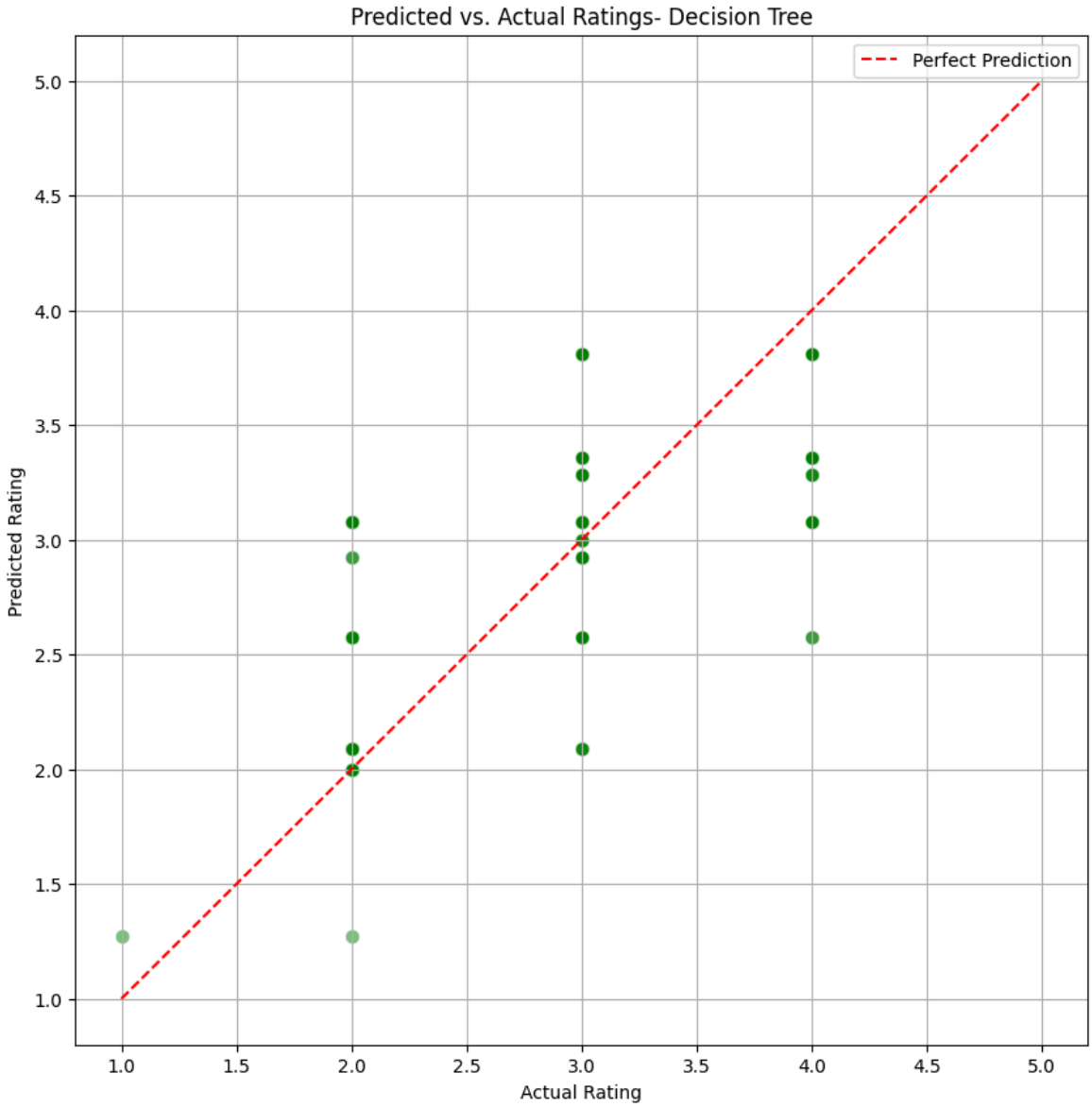
Ridge Regression Model:



Actual	Predicted	Count
1	1.0	1

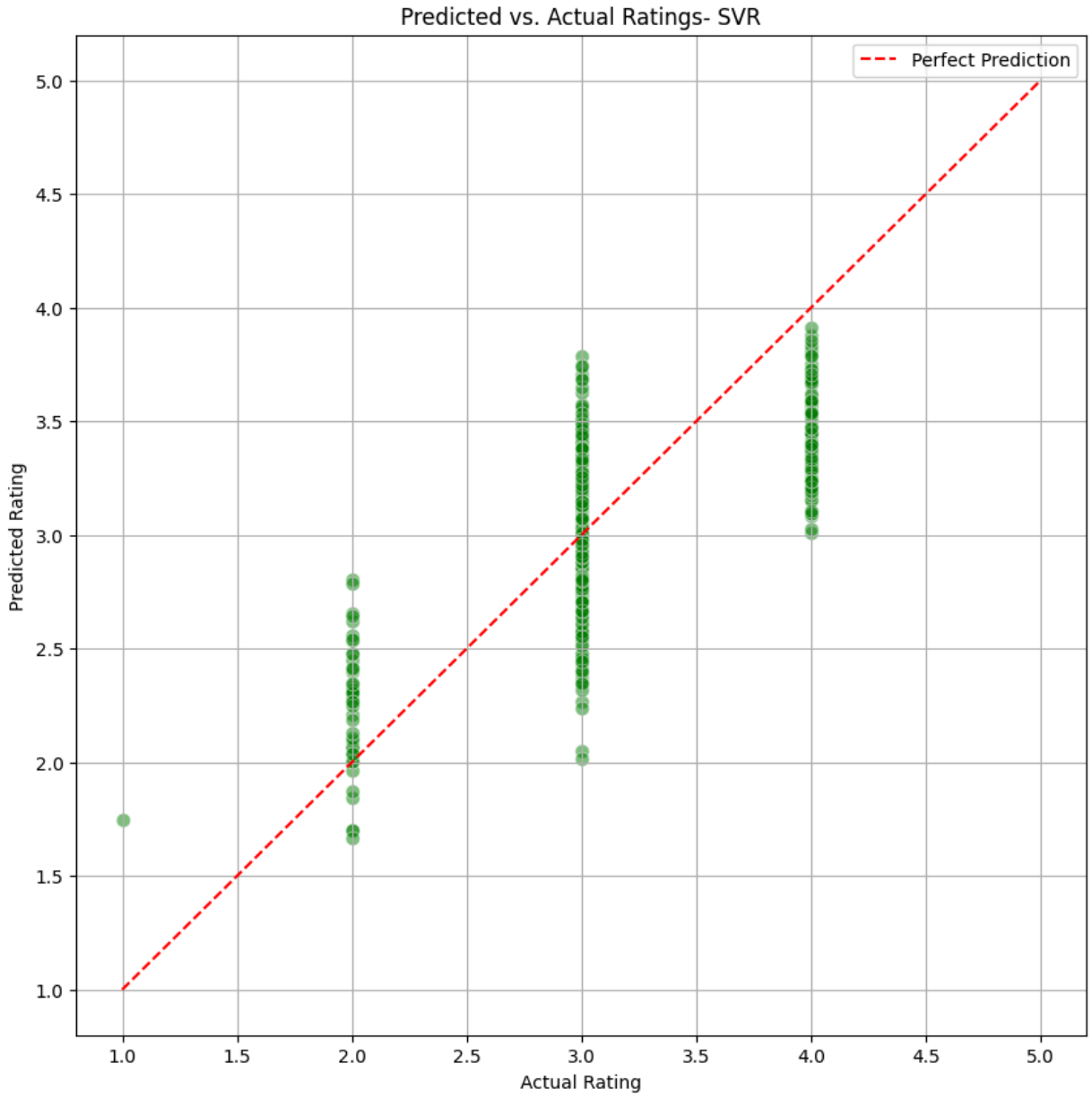
2	1.0	3
2	2.0	35
2	3.0	4
3	2.0	47
3	3.0	233
4	3.0	86
4	4.0	1

Decision Tree Regressor Model:



Actual	Predicted	Count
1	1.0	1
2	1.0	1
2	2.0	26
2	3.0	15
3	2.0	4
3	3.0	244
3	4.0	32
4	3.0	26
4	4.0	61

Support Vector Regression Model:



Actual	Predicted	Count
1	2.0	1
2	2.0	34
2	3.0	8
3	2.0	19

3	3.0	247
3	4.0	14
4	3.0	49
4	4.0	38

The graphs show that the Ridge regression model has a wider vertical spread, with points farther from the perfect prediction line compared to the SVR and Decision Tree models. The Decision Tree graph appears much more clustered because it produces predictions at specific values, with each prediction equal to the average of the training samples in the leaf where the test observation falls. Since the graphs can be difficult to interpret, the tables provide additional information. I have highlighted the rows where a perfect prediction is achieved. Ridge Regression has 270 perfect values, Decision Tree has 332, and SVR has 319. Interestingly, while SVR predicted more perfect 3's and 2's, it performed worse at predicting perfect 4's, which is a key reason why the Decision Tree model outperformed the SVR model overall.

V. Conclusion

In conclusion, based on the error metrics and supporting tables, the Decision Tree model performed the best in predicting surf rating as it had the lowest RMSE and MAE and achieved the highest number of perfect predictions.