MSDS 601

Linear Regression

Final Project

Report

Group Members:

- Lawrence Lin
- Nestor Teodoro Chavez
- Surbhi Prasad

Table of Contents

Introduction

Pokémon originally started off as a trading card game and then grew into a television show and a video game with the help of Niantic and Nintendo. This project utilizes the Pokémon dataset that encompasses 1118 entries that span Pokémon across different 'generations.' Namely, it contains traditional Pokémon and new Pokémon. The raw dataset also contains 49 columns that range from name & pokédex number to ghost attack effectiveness. The dataset brings in information for all users. In other words, it provides general information and blends it with explicit naming conventions. However, it also provides information regarding capture rates and egg cycles for those more invested in Pokémon.

Research Questions

From the Pokémon dataset, we want to understand :

1. What are the relevant pokemon characteristics that can be influencing one of the parameters**, speed** (our target variable).
2. Check for the relationship between speed and other attributes of the Pokémon. Check how many have linear relationships with speed.
3. Quantify impact of chosen features on speed. Check accuracy of finalised fitted model to predict speed in future.

Data Overview

Resource: [Kaggle Dataset]

Dimensions: (1118, 49) [1118 rows and 49 columns]

Snapshot

	name	pokedex_number	abilities	typing	hp	attack	defense	special_attack	special_defense	speed	...	ground_attack_effectiveness
0	Bulbasaur	1	Overgrow~Chlorophyll	Grass~Poison	45	49	49	65	65	45	...	1.0
1	Ivysaur	2	Overgrow~Chlorophyll	Grass~Poison	60	62	63	80	80	60	...	1.0
2	Venusaur	3	Overgrow~Chlorophyll	Grass~Poison	80	82	83	100	100	80	...	1.0
3	Venusaur Gmax	3	Overgrow~Chlorophyll	Grass~Poison	80	82	83	100	100	80	...	1.0
4	Venusaur Mega	3	Thick Fat	Grass~Poison	80	100	123	122	120	80	...	1.0

5 rows × 49 columns

The columns included in the data include various attributes regarding Pokémon. These attributes cover ~45 pokemon variables with 14 of them categorical or nominal and rest as numerical variables. There are no null values except in evolves_from column as each pokemon has their own unique characteristics. The data was already in good shape and didn't require much cleaning. Based of summary, there can be seen definite outliers in column weight with maximum values outside bounds of interquartile range +/-1.5*IR. The variables we use in in our final model are

```
weight + hp + attack + defense + egg_cycles + capture_rate +
special_attack + C(shape) + C(can_evolve)
```

Weight, hp, attack, defense, and special_attack are floats. Can_evolve is a boolean that denotes whether a pokemon can evolve or not. Shape is a categorical variable showing the shape of the pokemon. Egg_cycles is a float that shows how long until the corresponding pokemon hatches.

Outliers

speed	1118.0	69.479428	30.036549	5.00	45.000	67.0	90.00	200.0
height	1118.0	21.427549	58.158211	1.00	5.000	10.0	16.00	1000.0
weight	1118.0	993.305009	2067.911505	1.00	88.000	302.5	800.00	10000.0
gen_introduced	1118.0	4.129696	2.337694	1.00	2.000	4.0	6.00	8.0

Methods

Used Model: Linear regression

This project is going through understanding data cleaning, outlier removal before interpretation,learning about variables relation with speed for all 1118 pokemons and variables selection based on relevance, whether linear regression is fitting here for speed regression, then moves to model fitting, model evaluation, and model selection, evaluation if all model assumptions are fulfilled. We take an iterative approach to building the model by first picking all predictors we'd like to analyze in our initial model. This comes with the initial data exploration. We then implore the various method selection criterias Mallow's Cp and AIC/BIC to identify candidate

models. Finally, we attempt to fix our dataset by getting influential points and removing normality issues with various tests and re-run our model along with these statistical tests to understand if we've resolved problems or model improvement.

EDA

Speed: Target Variable

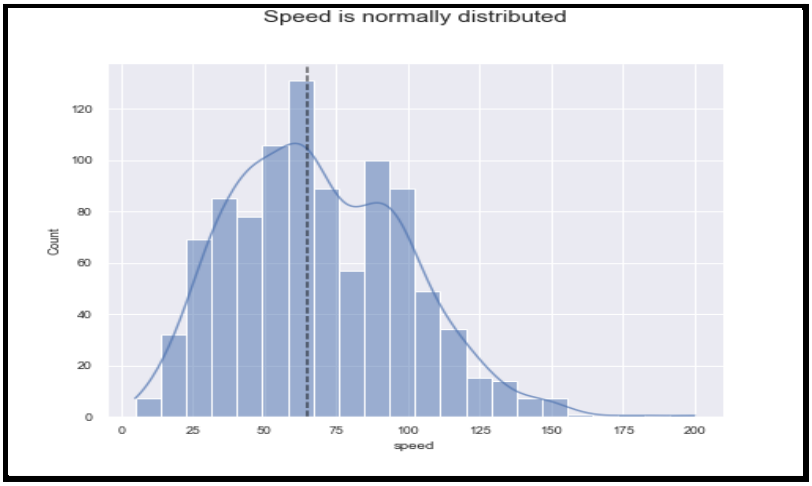

Figure 1: Histogram of Speed Distributions

The global median speed of pokemon in this dataset was about 67. The distribution of speeds among pokemon appears to be normal with a slight right skew. The gaussian kernel density estimation shows a normal probability distribution.

Shape:

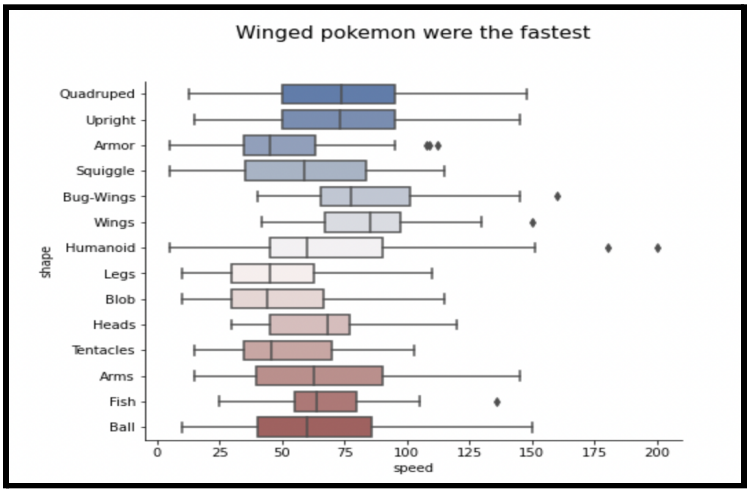

Figure 2: Boxplot of Speed vs Predictor

Among the different shapes of the pokemon, the range of speeds for each shape was relatively similar with the exception of the Bug-Wings shape and Wings

shape. The minimum speed of Bug-Wings and Wings shaped pokemon was around 40, while their median was over 75. That is considerably faster than the global median speed of 67.

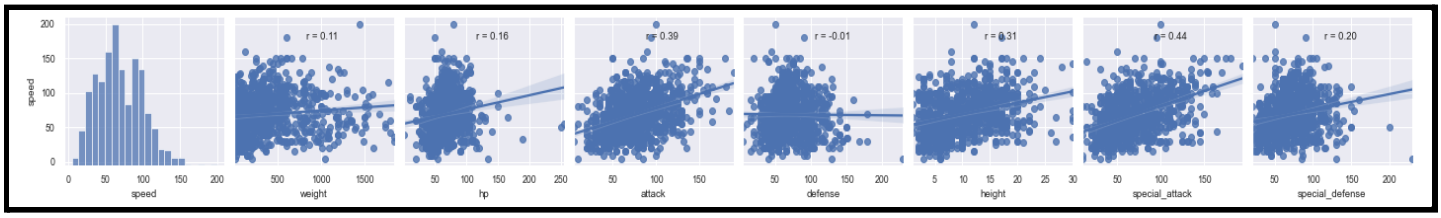

Figure 3: Numerical Features vs Speed

In this pairplot, we observe the scatterplots of speed with respect to some of our predictors. There are high correlations between speed and attack, height, special_attack, and special_defense. Scatterplots between speed and other attributes do not yield correlations as high as correlations between speed and the pokemon's stats. Hence linear relationships can be taken for some of the variables with speed.

Regression Analysis

Model Diagnosis

The model diagnosis on our dataset and predictors began with picking all predictors for the full model and calculating Variance Inflation Factor (VIF) scores. The next stage primarily consisted of identifying influential points by observing the external residual plot; ensuring heteroscedasticity was minimal by analyzing residual plots and running a Breusch-Pagan test; and checking assumptions of normality via normal distributions and QQ Plots. We did not check for nonlinearly between our target variable and our predictors because it is not common with Multi-Linear Regression (MLR) models.

VIF helps us diagnose issues caused by multicollinearity. In order to perform the test, we had to implement a few dummy variables on two of our parameters, shape and evolution.

	VIF Factor	features
0	65.890446	Intercept
1	2.349479	C(shape)[T.Arms]
2	2.422434	C(shape)[T.Ball]
3	2.008492	C(shape)[T.Blob]
4	1.560756	C(shape)[T.Bug-Wings]
5	2.006668	C(shape)[T.Fish]
6	1.422282	C(shape)[T.Heads]
7	4.301310	C(shape)[T.Humanoid]
8	1.527435	C(shape)[T.Legs]
9	4.937417	C(shape)[T.Quadruped]
10	2.014026	C(shape)[T.Squiggle]
11	1.564189	C(shape)[T.Tentacles]
12	5.211576	C(shape)[T.Upright]
13	3.020056	C(shape)[T.Wings]
14	2.446650	C(can_evolve)[T.True]
15	2.711778	weight
16	1.665466	hp
17	1.921830	attack
18	1.973323	defense
19	2.404971	height
20	1.670242	special_attack
21	2.175106	special_defense

Figure 4: VIF Diagnosis for Original Model

The data above shows that VIF values range from 1 to 5.3. In this case there isn't a cause for concern regarding Multicollinearity occurring with our current model. With the given VIF values from Figure 4, we are eligible to move into the next stage of model diagnosis.

In order to ensure that our model also didn't have any major signs of heteroscedasticity we decided to investigate the bandwidth of the residuals. In other words, in Figure 1, there is a graphed relationship between fitted values and their residuals. This was done in order to further understand which observations, if any, were not in alignment with the entire dataset.

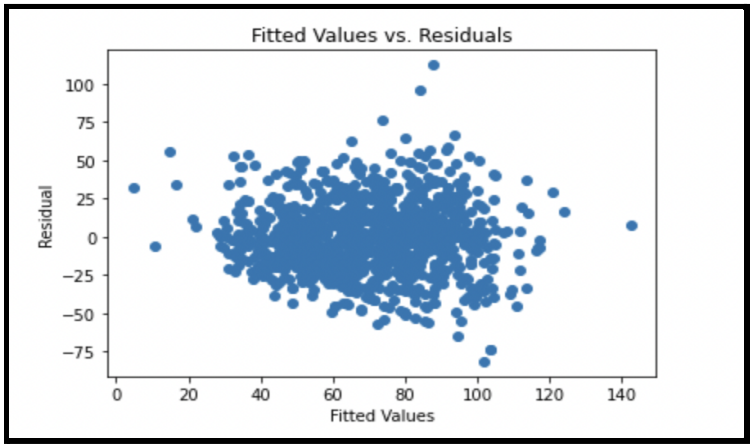

Figure 5: Fitted Values vs Residuals for Original Model

From this approach, we were able to verify our initial concerns that this dataset, although very accessible, did have some alarms regarding roughly ~1200

observations. These inconsistencies were marked as influential points that were dealt with later on in the model diagnosis. However, in order to truly trust the visual representation, we exposed our model to hypothesis testing through the use of the Breusch-Pagan Test. We leveraged Python's statsmodel library in order to reach a p-value of 3.35×10^{-6}. With a predetermined alpha value of 0.05, we accept the null hypothesis and therefore conclude that Heteroscedasticity is indeed a problem with the current stage of our model and the dataset.

We then attempted to remove the influential points and reuse the Breusch-Pagan Test. Unfortunately, we concluded again that heteroscedasticity still exists in our model and perhaps it's a cause for concern. For our model, we will continue to use the predictors and continue with diagnosing the model.

In Figure 6, we've used the original dataset in order to understand the assumptions of normality and if they've been violated. From the figure provided, we see that the normality assumptions hold true but there is a bit of uncertainty with the ends. As we remove the influential points and rerun the plot, we see that it is an even better fit. The normality assumptions should hold true for our model on the new dataset. QQ Plots should be diagonal in MLR. This will ensure that our model follows the model assumptions regarding normality.

Moreover, the Jarque-Bera test statistics gives us a value that is less than alpha/2. In this case, we would see that our test fails to reject the null hypothesis that the residuals are approximately normally distributed. Due to the Jarque-Bera Test being very sensitive to outliers, this test statistic is very significant in that we can state that our residuals are approximately normal. This also agrees with our findings in Figure 7 below.

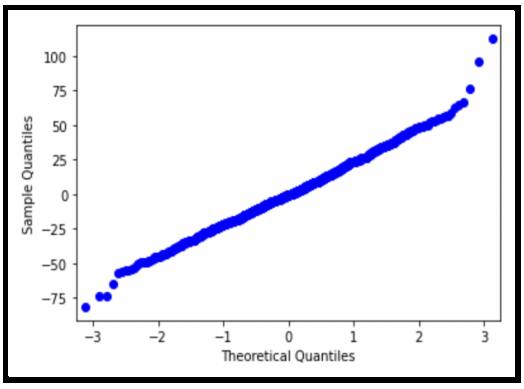

Figure 6: QQ Plot

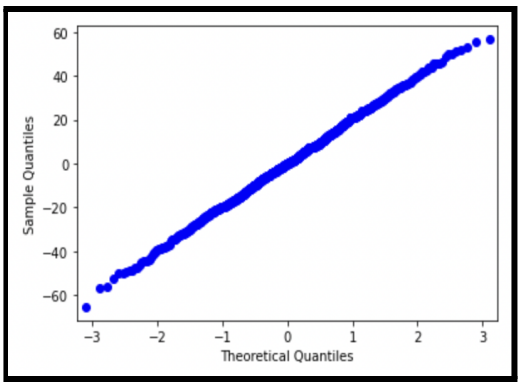

Figure 7: QQ Plot without Influential Points

Model Selection

We utilized feature selection by first running with more features than removed features that were insignificant in t-tests in model summary.

E.g. We removed `special_defense,gen_introduced,female_rate and legendary` as these were insignificant in model.

	coef	std err	t	P>|t|	[0.025	0.975]
weight	-0.0166	0.003	-5.780	0.000	-0.022	-0.011
hp	-0.3074	0.040	-7.730	0.000	-0.386	-0.229
attack	0.2046	0.034	6.081	0.000	0.139	0.271
defense	-0.3650	0.036	-10.200	0.000	-0.435	-0.295
height	0.6934	0.212	3.278	0.001	0.278	1.108
egg_cycles	0.2138	0.041	5.181	0.000	0.133	0.295
female_rate	-2.0742	3.485	-0.595	0.552	-8.914	4.766
capture_rate	-0.0656	0.014	-4.650	0.000	-0.093	-0.038
gen_introduced	-0.4181	0.314	-1.330	0.184	-1.035	0.199
special_attack	0.1310	0.031	4.279	0.000	0.071	0.191
special_defense	0.0191	0.040	0.476	0.634	-0.060	0.098

Then we ran best subsets models to narrow down few candidate models.For this, we ran linear regression on all possible combinations of model features where features used (finalised so far form EDA and initial few models fitted above) are:

```
weight + hp + attack + defense + egg_cycles + capture_rate +
special_attack
```

On skimming through adjusted R squared and Mallow's CP for these models, we chose two candidates and then further confirmed it from AIC, BIC values, which had lowest scores.

This removed the 'height' variable only from our earlier chosen model.

Final features:

```
weight + hp + attack + defense + egg_cycles + capture_rate +
special_attack + C(shape) + C(can_evolve)
```

	Predictors	Adjusted_R_Squared	Mallows CP	PredictorName
254	23	0.492968	11.000000	weight,hp,attack,defense,height,egg_cycles,cap...
249	22	0.486728	21.852570	weight,hp,attack,defense,egg_cycles,capture_ra...
253	22	0.486248	22.761918	hp,attack,defense,height,egg_cycles,capture_ra...
242	21	0.485071	24.023958	hp,attack,defense,egg_cycles,capture_rate,spec...
246	22	0.482616	29.654057	weight,hp,attack,defense,height,egg_cycles,cap...
247	22	0.480109	34.410485	weight,hp,attack,defense,height,egg_cycles,spe...
239	21	0.474640	43.835281	hp,attack,defense,height,egg_cycles,capture_ra...
222	21	0.473760	45.507498	weight,hp,attack,defense,egg_cycles,special_at...
240	21	0.473595	45.820475	hp,attack,defense,height,egg_cycles,special_at...

AIC **BIC**

```
[(8849.219616783905, 8951.686088677574),
 (8813.967520977454, 8916.433992871123),
 (8904.736821003014, 9007.203292896684),
 (8845.965091484824, 8948.431563378494),
 (8778.906935935702, 8881.373407829371),
 (8780.838808865523, 8883.305280759192),
 (8810.2440714981, 8912.71054339177),
 (8759.414401526914, 8861.880873420583),
 (8894.425265033933, 8996.891736927602),
 (8792.36364846636, 8894.83012036003),
 (8830.60252937075, 8933.069001264419),
```

Summary of Findings

In summary, we initially fit a model with variables selected through exploratory data analysis and our knowledge of pokemon. These variables are shown in Figure8. We remove special_defense, gen_introduced, female_rate, and legendary since we fail to reject the null hypothesis that the reduced model is true for these variables.

	sum_sq	df	F	PR(>F)
C(shape)	43303.483979	13.0	7.195421	1.588017e-13
C(legendary)	444.334345	1.0	0.959813	3.274847e-01
C(can_evolve)	48841.112242	1.0	105.502390	1.550913e-23
weight	6614.882291	1.0	14.288902	1.666169e-04
hp	27658.603049	1.0	59.745747	2.754205e-14
attack	17121.611049	1.0	36.984639	1.726612e-09
defense	48160.806477	1.0	104.032852	3.023661e-23
height	4974.559017	1.0	10.745617	1.083390e-03
egg_cycles	12428.321764	1.0	26.846597	2.692813e-07
female_rate	163.957278	1.0	0.354166	5.519065e-01
capture_rate	10008.459392	1.0	21.619417	3.797301e-06
gen_introduced	819.021210	1.0	1.769180	1.838046e-01
special_attack	8476.517230	1.0	18.310247	2.068090e-05
special_defense	104.752181	1.0	0.226277	6.344084e-01
Residual	437476.830210	945.0	NaN	NaN

Figure 8: Partial ANOVA table for initial model

We perform best subset model selection by fitting 255 models from all possible combinations of the 10 remaining variables. We report their Mallow's Cp, Adjusted R-squared, and AIC/BIC. The three best models are shown in Figure 9.

	Predictors	Adjusted_R_Squared	Mallows CP	PredictorName
254	23	0.492968	11.000000	weight,hp,attack,defense,height,egg_cycles,capture_rate,special_attack,C(shape),C(can_evolve)
249	22	0.486728	21.852570	weight,hp,attack,defense,egg_cycles,capture_rate,special_attack, C(shape),C(can_evolve)
253	22	0.486248	22.761918	hp,attack,defense,height,egg_cycles,capture_rate,special_attack, C(shape),C(can_evolve)

Figure 9: Top 3 models from Best Subsets

Because it has marginally higher Adjusted R-Squared and Mallow's Cp that is marginally closer to k = 22, we select candidate model 249. The variable height is removed and we are left with this model:

```
Speed ~ weight + hp + attack + defense + egg_cycles + capture_rate +
special_attack + C(shape) + C(can_evolve)
```

We fit this model with OLS regression and find that every single variable is significant. The only variable that is borderline significance is weight with a p = 0.044. If we use Bonferroni correction to adjust for the multiple hypothesis tests, then we can reject weight as a significant predictor of speed. The full regression summary is in Appendix A1. However, there are 62 influential points in the model as measured by Cook's Distance. This can greatly affect the slope of our regression line and the estimates of our coefficients.

Influential Points

Figure 10: Scatter Plot of Studentized Residuals

To get a better idea of what the difference is, we fit the same model but with all the influential points removed. The full regression summary excluding influential points is in Appendix A1. The Adjusted R-squared increases from about 49% to about 57% after removing the influential points. Furthermore, the coefficient estimates for Legs-shaped pokemon, Blob-shaped pokemon and Bug-Wings-shaped pokemon change dramatically. The largest change is with Bug-Wings shaped pokemon whose slope coefficient with influential points is 15.7, but only 6.3 without. The top 3 differences are listed in Figure 11.

	With influential	Without influential	Difference
C(shape)[T.Bug-Wings]	15.702089	6.298182	9.403906
C(shape)[T.Blob]	-4.897789	-9.219037	4.321248
C(shape)[T.Legs]	1.662223	-2.335118	3.997341

Figure 11: Difference between model coefficients

In conclusion, our best model finds every single predictor chosen by best subset model selection to be significant except for weight. However, that changes if we decide to exclude influential points, which appear to disproportionately affect the estimates of speed of certain shapes of pokemon.

Conclusion

Potential Problems and Suggestions:

Data:

The data has many outliers like in height , weight which reduces no. of pokemons from 1118 to ~700. We can probably capture more correct data.

Analysis:

In our final model, we can still see heteroscedasticity. Also, we are not achieving very high goodness of fit which could be because of speed not being linearly related to most of the predictors. We might need to transform some of our features to make relations more linear. Also, the heteroscedasticity still persists in our final model. For which we can run corrected studentized t-test to see if actually some variables mayn't alter inference and affect model results.

Impact

Best Model:

The best model for speed is its relationship with variables

```
weight + hp + attack + defense + egg_cycles + capture_rate +
special_attack + C(shape) + C(can_evolve)
```

where we can see the highest positive impacts of the shape of pokemon being quadrupled, fish or upright. We see speed is hindered if pokemon has tentacles or blobs. Also, evolution possibility(can_envolve) slows down pokemons.

Future Analysis

By looking at our final model, I would like to see how a Pokemon's speed varies with log transformation of some of the features and how model performance is impacted thereafter.

Appendix

Dep. Variable:	speed	**R-squared:**	0.498
Model:	OLS	**Adj. R-squared:**	0.487
Method:	Least Squares	**F-statistic:**	44.85
Date:	Wed, 13 Oct 2021	**Prob (F-statistic):**	2.85e-126
Time:	00:01:50	**Log-Likelihood:**	-4356.6
No. Observations:	972	**AIC:**	8757.
Df Residuals:	950	**BIC:**	8865.
Df Model:	21		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	92.2890	6.796	13.580	0.000	78.952	105.626
C(shape)[T.Arms]	-1.2643	4.638	-0.273	0.785	-10.367	7.839
C(shape)[T.Ball]	2.8493	4.450	0.640	0.522	-5.884	11.582
C(shape)[T.Blob]	-4.8978	4.770	-1.027	0.305	-14.259	4.463
C(shape)[T.Bug-Wings]	15.7021	5.824	2.696	0.007	4.272	27.132
C(shape)[T.Fish]	8.0777	4.953	1.631	0.103	-1.643	17.799

C(shape)[T.Heads]	7.9510	6.466	1.230	0.219	-4.739	20.641
C(shape)[T.Humanoid]	2.7242	3.986	0.684	0.494	-5.097	10.546
C(shape)[T.Legs]	1.6622	5.856	0.284	0.777	-9.830	13.155
C(shape)[T.Quadruped]	14.8112	3.845	3.852	0.000	7.265	22.357
C(shape)[T.Squiggle]	-7.3909	5.277	-1.401	0.162	-17.746	2.964
C(shape)[T.Tentacles]	-8.9020	5.967	-1.492	0.136	-20.612	2.808
C(shape)[T.Upright]	9.3254	3.864	2.414	0.016	1.743	16.908
C(shape)[T.Wings]	16.7946	4.257	3.945	0.000	8.440	25.149
C(can_evolve)[T.True]	-24.6836	2.168	-11.387	0.000	-28.938	-20.430
hp	-0.2998	0.039	-7.691	0.000	-0.376	-0.223
attack	0.2150	0.031	6.957	0.000	0.154	0.276
defense	-0.3582	0.033	-10.911	0.000	-0.423	-0.294
weight	-0.0047	0.002	-2.017	0.044	-0.009	-0.000
egg_cycles	0.2378	0.033	7.202	0.000	0.173	0.303
capture_rate	-0.0683	0.014	-5.003	0.000	-0.095	-0.042
special_attack	0.1590	0.029	5.433	0.000	0.102	0.216

A1: Regression Summary

Dep. Variable:	speed	**R-squared:**	0.582
Model:	OLS	**Adj. R-squared:**	0.573
Method:	Least Squares	**F-statistic:**	58.99
Date:	Wed, 13 Oct 2021	**Prob (F-statistic):**	3.22e-152
Time:	00:01:53	**Log-Likelihood:**	-3939.5
No. Observations:	910	**AIC:**	7923.
Df Residuals:	888	**BIC:**	8029.
Df Model:	21		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	97.5595	6.222	15.680	0.000	85.348	109.771
C(shape)[T.Arms]	-2.6338	4.190	-0.629	0.530	-10.856	5.589
C(shape)[T.Ball]	2.0724	3.982	0.520	0.603	-5.743	9.888
C(shape)[T.Blob]	-9.2190	4.430	-2.081	0.038	-17.913	-0.525
C(shape)[T.Bug-Wings]	6.2982	6.122	1.029	0.304	-5.717	18.313
C(shape)[T.Fish]	8.1060	4.474	1.812	0.070	-0.675	16.887
C(shape)[T.Heads]	5.8816	5.620	1.047	0.296	-5.148	16.912
C(shape)[T.Humanoid]	0.1050	3.546	0.030	0.976	-6.855	7.065

C(shape)[T.Legs]	-2.3351	5.276	-0.443	0.658	-12.689	8.019
C(shape)[T.Quadruped]	14.3530	3.419	4.198	0.000	7.643	21.063
C(shape)[T.Squiggle]	-8.3902	4.695	-1.787	0.074	-17.604	0.824
C(shape)[T.Tentacles]	-10.7094	5.734	-1.868	0.062	-21.963	0.544
C(shape)[T.Upright]	8.6166	3.438	2.506	0.012	1.870	15.364
C(shape)[T.Wings]	15.7430	3.778	4.167	0.000	8.328	23.158
C(can_evolve)[T.True]	-26.1068	1.964	-13.296	0.000	-29.960	-22.253
hp	-0.3613	0.040	-9.040	0.000	-0.440	-0.283
attack	0.2266	0.028	7.981	0.000	0.171	0.282
defense	-0.3431	0.030	-11.521	0.000	-0.402	-0.285
weight	-0.0065	0.002	-3.141	0.002	-0.011	-0.002
egg_cycles	0.2258	0.031	7.355	0.000	0.166	0.286
capture_rate	-0.0766	0.012	-6.346	0.000	-0.100	-0.053
special_attack	0.1612	0.027	5.959	0.000	0.108	0.214

A2: Regression Summary Excluding Influential Points