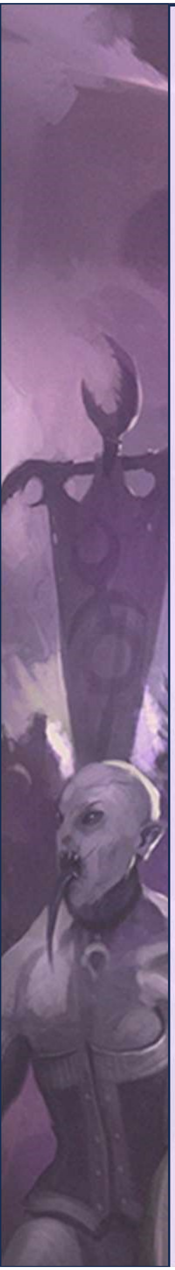
Competitive Warhammer Analysis:

https://github.com/jtspedding



Analysis Aims:

To optimise my future lists by identifying…

1. Units that are under-costed based on their statistics,

2. Units that are over-costed based on their statistics.

Unit Warscroll:

WARSCROLL

Orruk Brutes

MOVE 4"
WOUNDS 3
SAVE 4+
BRAVERY 6

Charging into battle with joyous bellows, Brutes seek out the largest enemies to batter into submission. Wearing the thickest armour and wielding huge weapons, they enjoy nothing more than dishing out a good and proper bashing.

MELEE WEAPONS	Range	Attacks	To Hit	To Wound	Rend	Damage
Brute Choppas	1"	4	3+	3+	-1	1
Jagged Gore-hacka	2"	3	3+	3+	-2	1
Gore-choppa	2"	3	4+	3+	-2	2
Boss Choppa	1"	3	3+	3+	-1	2
Boss Klaw and Brute Smasha	1"	4	4+	3+	-1	2

PITCHED BATTLE PROFILE
Unit Size: 5 **Points:** 140
Battlefield Role: Battleline
Base size: 40mm

Each model in an Orruk Brutes unit is armed with 1 of the following weapon options: Brute Choppas; or Jagged Gore-hacka. All models in the unit must be armed with the same weapon option. 1 in every 5 models can replace their weapon option with a Gore-choppa.

BATTALIONS: This warscroll can be used in the following warscroll battalions:
- Brutefist
- Ironfist
- Weirdfist
- Da Bossfist
- Dakkbad's Brawl
- Moggorz's Rekrootin' Krew

CHAMPION: 1 model in this unit can be a Brute Boss. Replace that model's weapon option with a Boss Choppa, or a Boss Klaw and Brute Smasha.

You Messin'?: *Most beings with half an ounce of common sense swiftly wither under the furious gaze of an orruk Brute who has marked his territory.*

Enemy models with a Wounds characteristic of 1 that are within 3" of this unit cannot contest objectives.

Duff Up da Big Thing: *The Brutes of the Ironjawz excel at fighting and killing the most powerful foes.*

Add 1 to hit rolls for attacks made by this unit that target a unit with a Wounds characteristic of 4 or more.

Unit Warscroll:

WARSCROLL

Blissbarb Archers

MOVE 6"
WOUNDS 1
SAVE 6+
BRAVERY 6

Blissbarb Archers are the lowest class of Sybarite, but no less deadly for it. Even when running pell-mell across the field they fire with deadly accuracy, laughing with glee as their sharp and toxin-laced projectiles strike home.

MISSILE WEAPONS	Range	Attacks	To Hit	To Wound	Rend	Damage
Blissbarb Bow	18"	2	3+	4+	-1	1

MELEE WEAPONS	Range	Attacks	To Hit	To Wound	Rend	Damage
Sybarite Blade	1"	1	3+	4+	-	1

PITCHED BATTLE PROFILE

Unit Size: 11 **Points:** 170
Battlefield Role: Battleline

MODEL	BASE SIZE
Blissbarb Archers	28.5mm
Blissbrew Homonculus	25mm

Each model in a Blissbarb Archers unit is armed with a Blissbarb Bow and Sybarite Blade.

BATTALIONS: This warscroll can be used in the following warscroll battalions:
- Depraved Carnival

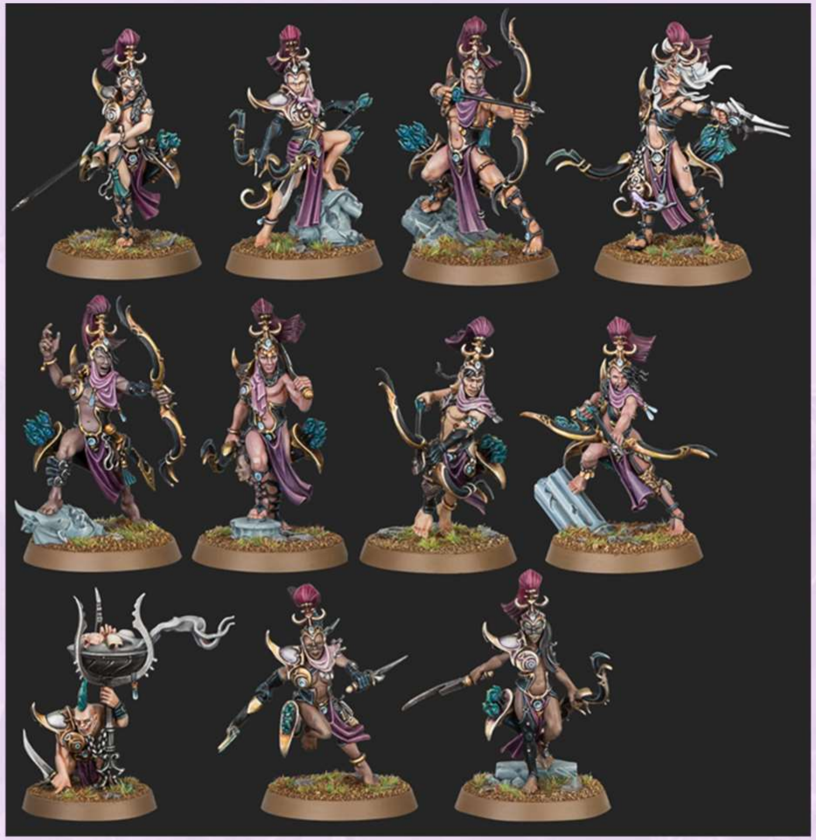
CHAMPION: 1 model in this unit can be a High Tempter. Add 1 to the Attacks characteristic of that model's Blissbarb Bow.

BLISSBREW HOMONCULUS: 1 in every 11 models in this unit must be a Blissbrew Homonculus. A Blissbrew Homonculus is armed with a Sybarite Blade. Add 1 to wound rolls for attacks made with missile weapons by this unit while it includes any Blissbrew Homonculi.

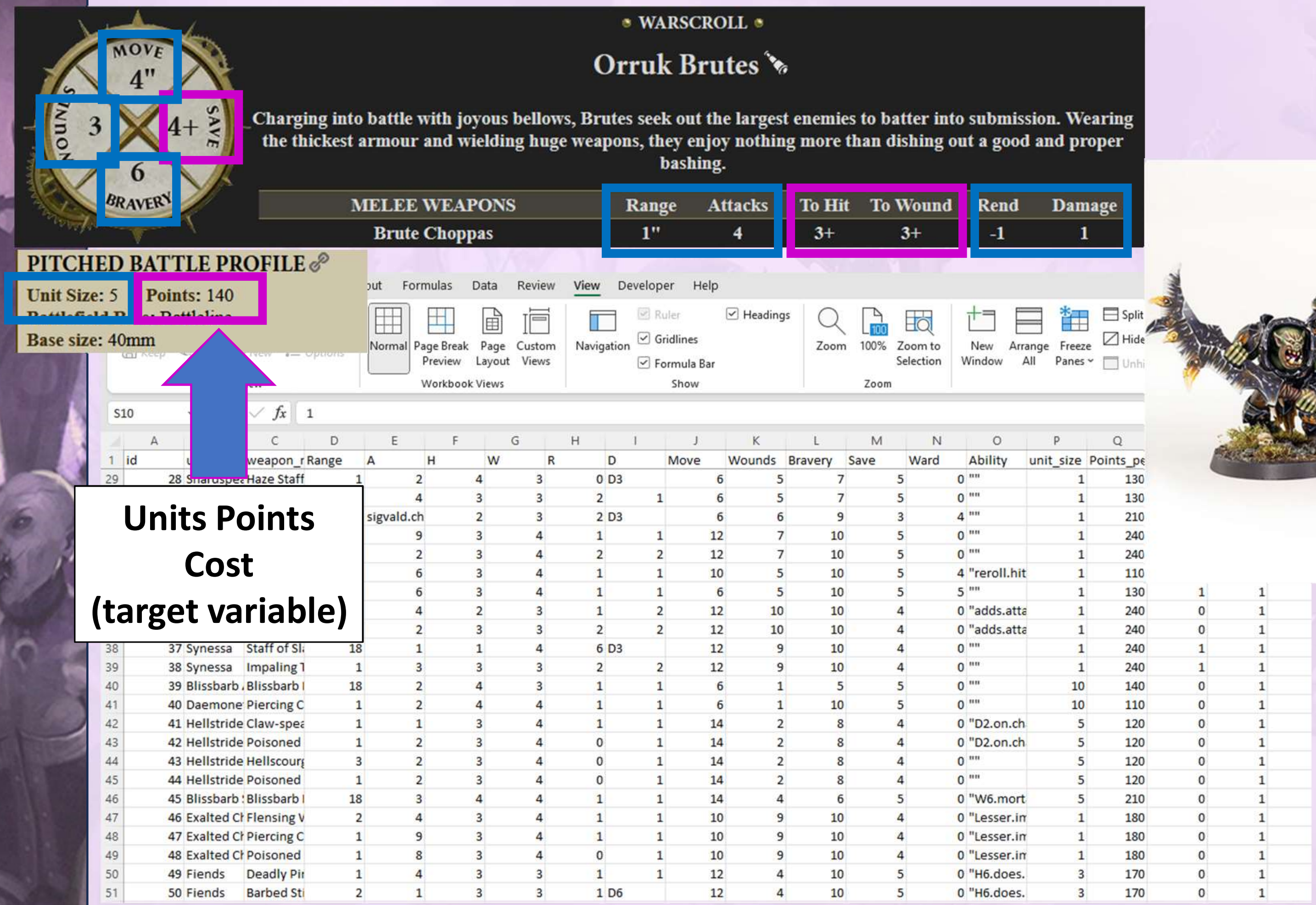
Light-footed Killers: *Blissbarb Archers can deliver pinpoint shots even while cavorting wildly across the battlefield.*

This unit can run and still shoot later in the turn.

KEYWORDS CHAOS, HEDONITES OF SLAANESH, MORTAL, SLAANESH, BLISSBARB ARCHERS

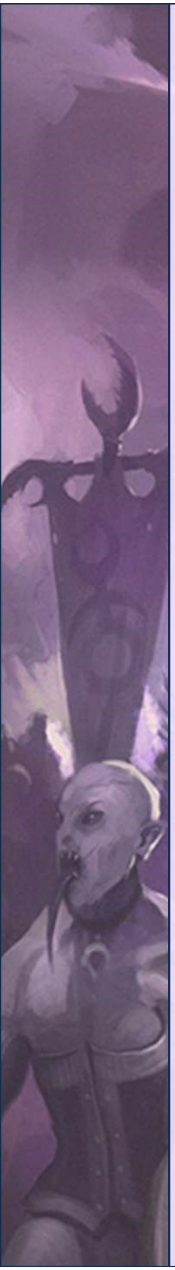


Unit Warscroll:



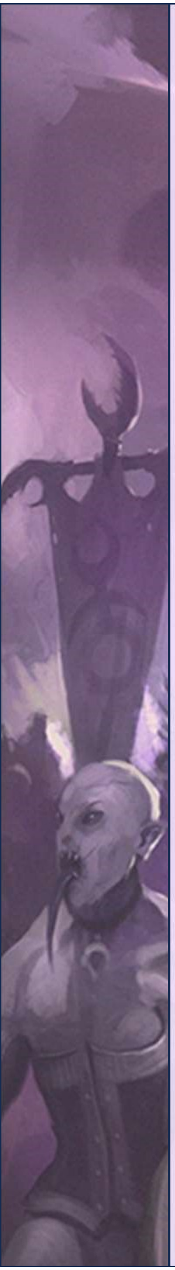
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	id	u	weapon_r	Range	A	H	W	R	D	Move	Wounds	Bravery	Save	Ward	Ability	unit_size	Points_pe		
29	28	Sharduspe	Haze Staff	1	2	4	3	0	D3	6	5	7	5	0	""	1	130		
					4	3	3	2	1	6	5	7	5	0	""	1	130		
					sigvald.ch	2	3	2	D3	6	6	9	3	4	""	1	210		
					9	3	4	1	1	12	7	10	5	0	""	1	240		
					2	3	4	2	2	12	7	10	5	0	""	1	240		
					6	3	4	1	1	10	5	10	5	4	"reroll.hit	1	110		
					6	3	4	1	1	6	5	10	5	5	""	1	130	1	1
					4	2	3	1	2	12	10	10	4	0	"adds.atta	1	240	0	1
					2	3	3	2	2	12	10	10	4	0	"adds.atta	1	240	0	1
38	37	Synessa	Staff of Sl	18	1	1	4	6	D3	12	9	10	4	0	""	1	240	1	1
39	38	Synessa	Impaling T	1	3	3	3	2	2	12	9	10	4	0	""	1	240	1	1
40	39	Blissbarb A	Blissbarb I	18	2	4	3	1	1	6	1	5	5	0	""	10	140	0	1
41	40	Daemone	Piercing C	1	2	4	4	1	1	6	1	10	5	0	""	10	110	0	1
42	41	Hellstride	Claw-spea	1	1	3	4	1	1	14	2	8	4	0	"D2.on.ch	5	120	0	1
43	42	Hellstride	Poisoned	1	2	3	4	0	1	14	2	8	4	0	"D2.on.ch	5	120	0	1
44	43	Hellstride	Hellscour	3	2	3	4	0	1	14	2	8	4	0	""	5	120	0	1
45	44	Hellstride	Poisoned	1	2	3	4	0	1	14	2	8	4	0	""	5	120	0	1
46	45	Blissbarb S	Blissbarb I	18	3	4	4	1	1	14	4	6	5	0	"W6.mort	5	210	0	1
47	46	Exalted Ch	Flensing V	2	4	3	4	1	1	10	9	10	4	0	"Lesser.im	1	180	0	1
48	47	Exalted Ch	Piercing C	1	9	3	4	1	1	10	9	10	4	0	"Lesser.im	1	180	0	1
49	48	Exalted Ch	Poisoned	1	8	3	4	0	1	10	9	10	4	0	"Lesser.im	1	180	0	1
50	49	Fiends	Deadly Pi	1	4	3	3	1	1	12	4	10	5	0	"H6.does.	3	170	0	1
51	50	Fiends	Barbed St	2	1	3	3	1	D6	12	4	10	5	0	"H6.does.	3	170	0	1

Warhammer Analysis:



Stats Time!

Aims: to optimise the units in my list.



Warhammer Analysis:

Data Collection:

1) Attempted to pull data from Websites but alas the html was too dense. Therefore so manual processing was required (n = 172).

Data Cleaning:

1) Impute string values as relevant integers (e.g., D6 == 3.5).
2) Invert dice roll values (not necessary but helpful to quickly interpret interactions).
3) Compute unit wounds by unit size features.

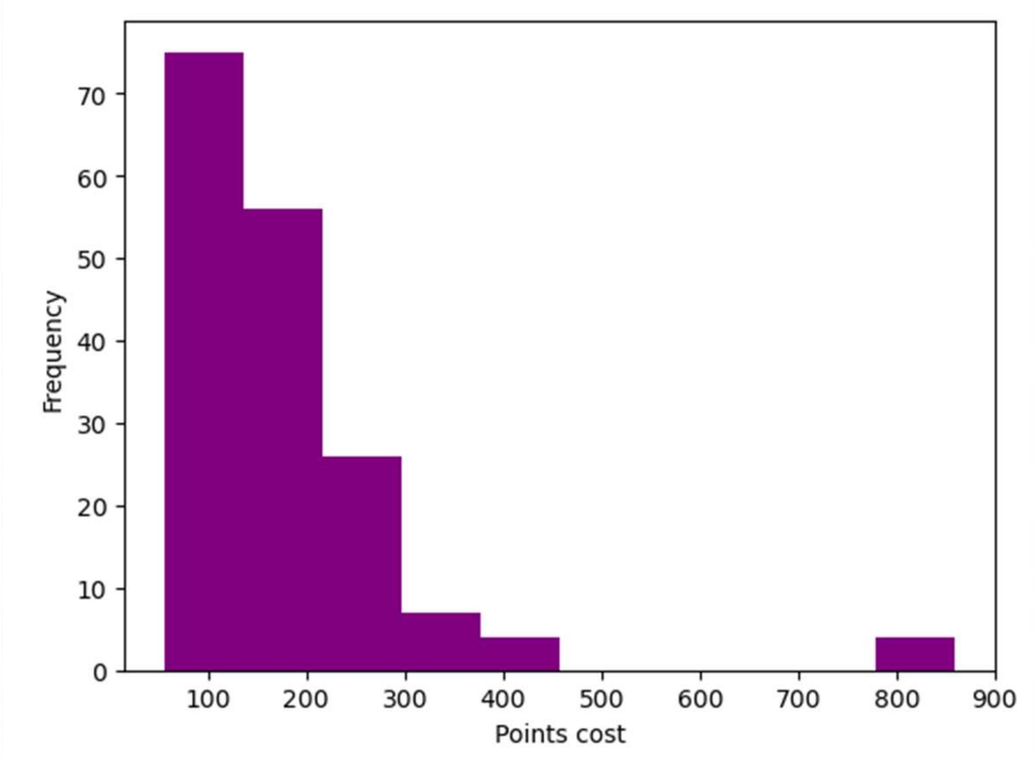
```python
#####################################################################################
# Data Cleaning:

df = df.drop("id", axis = 1)

# Need to replace the string/chars with numeric.
def string_replacer(x):
    x = x.replace("2D6", "7")
    x = x.replace("D3", "2")
    x = x.replace("3D6", "10.5")
    x = x.replace("D6", "3.5")
    x = x.replace("sigvald.charge", "7")
    x = x.astype(float)
    return(x)

# Need to the function to impute/fix string data points.
# df.dtypes
df["A"] = string_replacer(df["A"])
df["W"] = string_replacer(df["W"])
df["D"] = string_replacer(df["D"])
df["Move"] = string_replacer(df["Move"])

# Going to invert some of the features so that their coefs
# represent unit improvements (i.e., to-hit rolls on a 2+ are
# easier than to-hit rolls on a 4+ etc).
df["Save"] = 7 - df["Save"]
df["Ward"] = 7 - df["Ward"]
df["H"] = 7 - df["H"]
df["W"] = 7 - df["W"]
#####################################################################################
```

Warhammer Analysis:

EDA:

1) Examine feature histograms.

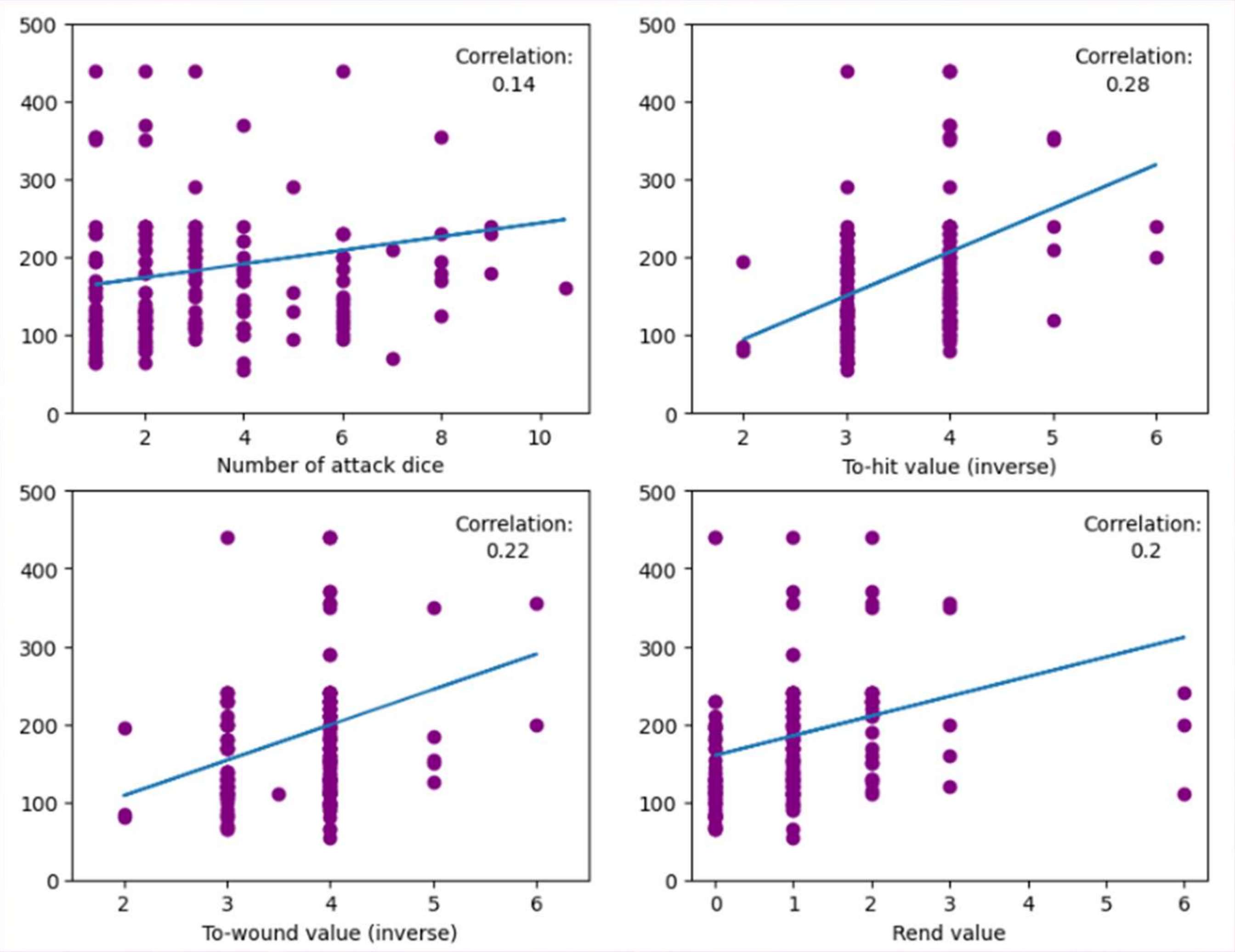
```
# Histogram of points costs.
plt.hist(df["Points_per_warscroll"], color="purple")
plt.xlabel("Points cost")
plt.ylabel("Frequency")
plt.show()
```



Warhammer Analysis:

EDA:

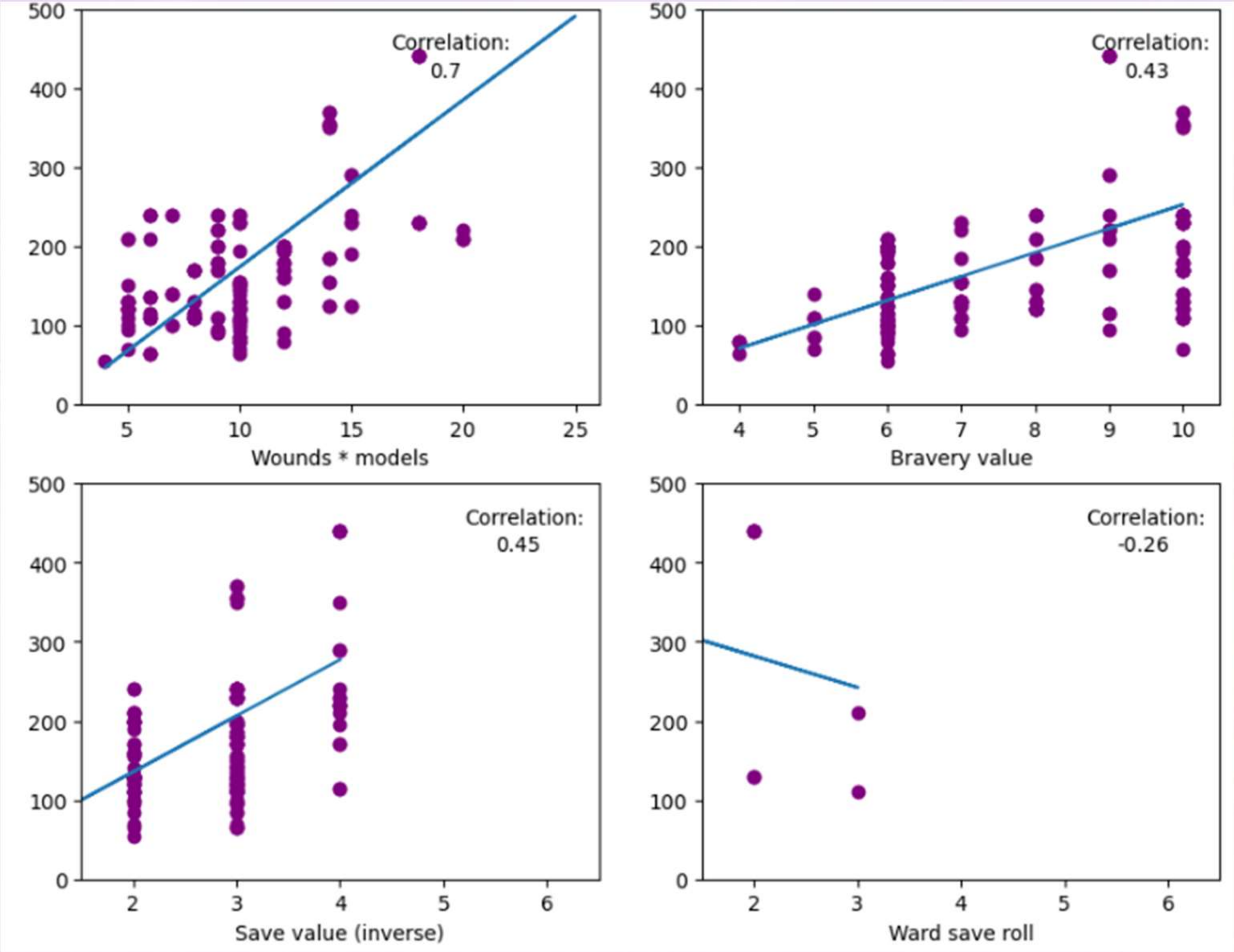
2) Examine scatterplots between the target and features.



Warhammer Analysis:

EDA:

2) Examine scatterplots between the target and features.



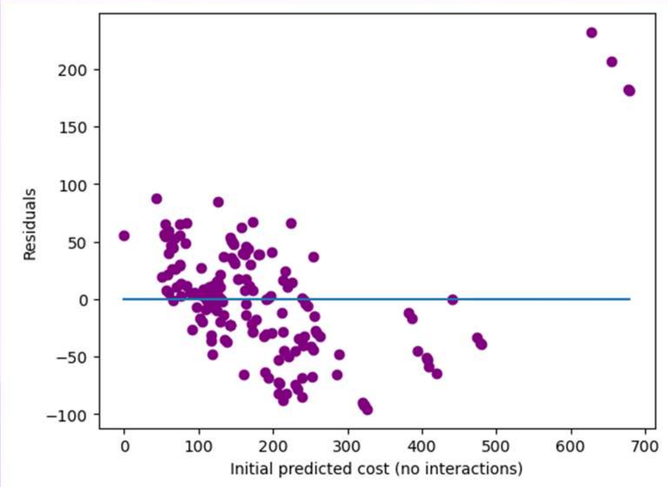
Warhammer Analysis:

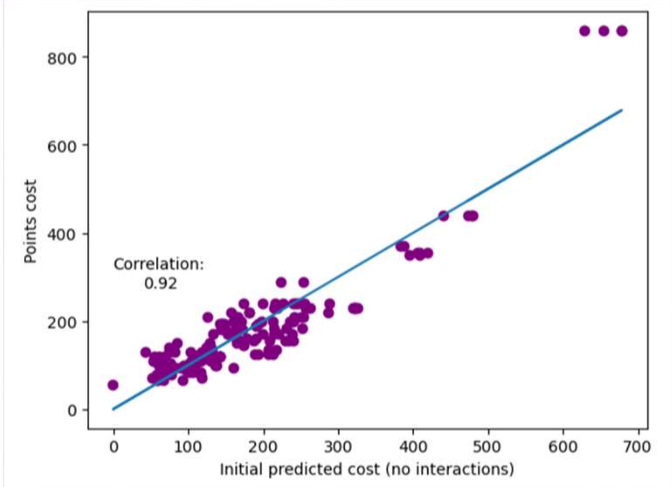
EDA:

3) Examine predicted values.

4) Examine residual values.

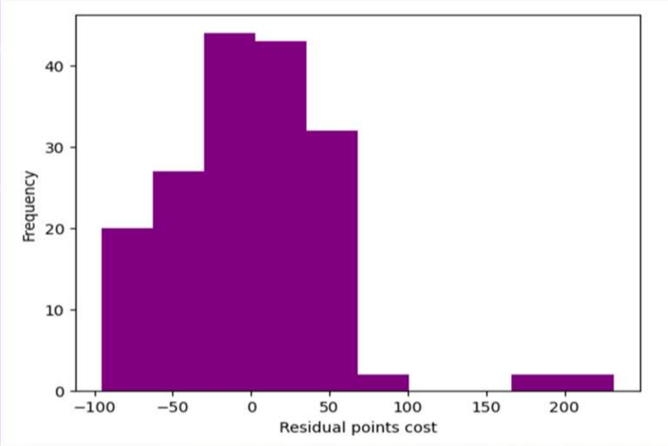
5) OLS assumptions etc.

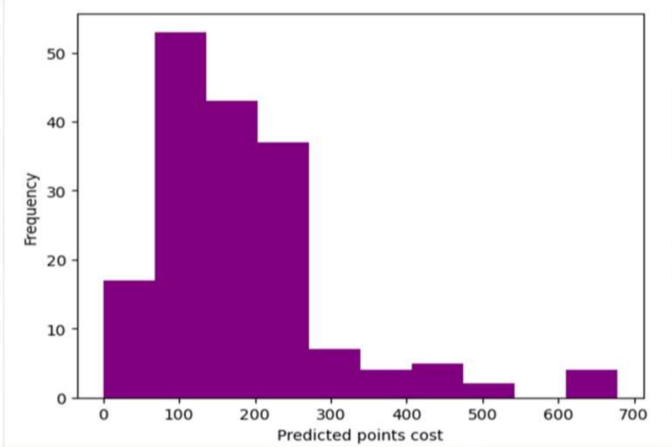





```
plt.hist(model.resid, color="purple");
plt.xlabel("Residual points cost");
plt.ylabel("Frequency");
```

```
plt.hist(model.fittedvalues, color="purple");
plt.xlabel("Predicted points cost");
plt.ylabel("Frequency");
```





Warhammer Analysis:

EDA:

6) Examine initial regression model coefficients.

```
#view model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Points_per_warscroll   R-squared:                       0.842
Model:                              OLS   Adj. R-squared:                  0.826
Method:                   Least Squares   F-statistic:                     55.28
Date:                  Mon, 13 Nov 2023   Prob (F-statistic):           1.78e-54
Time:                          12:00:26   Log-Likelihood:                -922.13
No. Observations:                   172   AIC:                             1876.
Df Residuals:                       156   BIC:                             1927.
Df Model:                            15
Covariance Type:              nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         -220.4718     55.006     -4.008      0.000    -329.125    -111.818
Range           -0.0268      0.925     -0.029      0.977      -1.853       1.800
A                1.8087      2.604      0.695      0.488      -3.334       6.951
H               23.8283      8.532      2.793      0.006       6.975      40.682
W              -12.2390      8.995     -1.361      0.176     -30.007       5.529
R                1.5733      5.609      0.280      0.779      -9.506      12.653
D                2.1816      6.696      0.326      0.745     -11.045      15.408
Move             1.2435      1.499      0.830      0.408      -1.717       4.204
Wounds           5.7168      1.975      2.894      0.004       1.815       9.619
Bravery          4.7949      3.743      1.281      0.202      -2.599      12.189
Save            21.7656      6.301      3.454      0.001       9.319      34.212
Ward             2.7065      3.545      0.764      0.446      -4.295       9.708
unit_size        3.8115      3.301      1.155      0.250      -2.708      10.331
Spells          70.8970      7.129      9.945      0.000      56.816      84.978
slaanesh_dummy -12.5517     13.205     -0.951      0.343     -38.636      13.532
Wounds_X_model  14.0543      1.883      7.464      0.000      10.335      17.774
==============================================================================
Omnibus:                       48.806   Durbin-Watson:                   0.862
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              138.088
Skew:                           1.144   Prob(JB):                     1.03e-30
Kurtosis:                       6.746   Cond. No.                         289.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Warhammer Analysis:

EDA:

6) Examine initial regression model coefficients.

7) Solved issue with high-low wounds and model counts.

```
#view model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Points_per_warscroll   R-squared:                       0.842
Model:                              OLS   Adj. R-squared:                  0.826
Method:                   Least Squares   F-statistic:                     55.28
Date:                  Mon, 13 Nov 2023   Prob (F-statistic):           1.78e-54
Time:                          12:00:26   Log-Likelihood:                -922.13
No. Observations:                   172   AIC:                             1876.
Df Residuals:                       156   BIC:                             1927.
Df Model:                            15
Covariance Type:              nonrobust
================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          -220.4718     55.006     -4.008      0.000    -329.125    -111.818
Range            -0.0268      0.925     -0.029      0.977      -1.853       1.800
A                 1.8087      2.604      0.695      0.488      -3.334       6.951
H                23.8283      8.532      2.793      0.006       6.975      40.682
W               -12.2390      8.995     -1.361      0.176     -30.007       5.529
R                 1.5733      5.609      0.280      0.779      -9.506      12.653
D                 2.1816      6.696      0.326      0.745     -11.045      15.408
Move              1.2435      1.499      0.830      0.408      -1.717       4.204
Wounds            5.7168      1.975      2.894      0.004       1.815       9.619
Bravery           4.7949      3.743      1.281      0.202      -2.599      12.189
Save             21.7656      6.301      3.454      0.001       9.319      34.212
Ward              2.7065      3.545      0.764      0.446      -4.295       9.708
unit_size         3.8115      3.301      1.155      0.250      -2.708      10.331
Spells           70.8970      7.129      9.945      0.000      56.816      84.978
slaanesh_dummy  -12.5517     13.205     -0.951      0.343     -38.636      13.532
Wounds_X_model   14.0543      1.883      7.464      0.000      10.335      17.774
==============================================================================
Omnibus:                       48.806   Durbin-Watson:                   0.862
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              138.088
Skew:                           1.144   Prob(JB):                     1.03e-30
Kurtosis:                       6.746   Cond. No.                         289.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

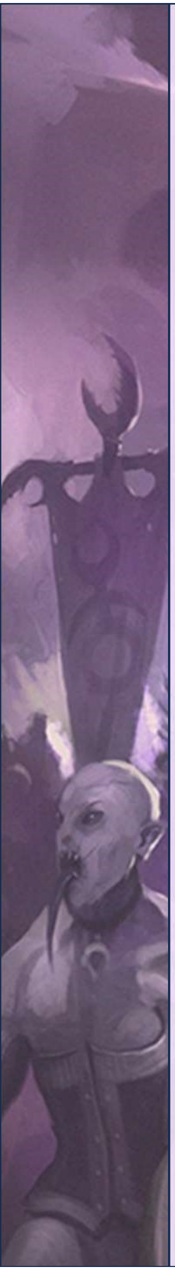
Warhammer Analysis:

Feature Engineering:

1) Standardise (or centre) variables and compute all 2-way interactions…

```
# Potential interaction effects to test.

# 2-way interactions.
df["A_X_H"] = preprocessing.scale(df["A"]) * preprocessing.scale(df["H"])
df["A_X_W"] = preprocessing.scale(df["A"]) * preprocessing.scale(df["W"])
df["A_X_D"] = preprocessing.scale(df["A"]) * preprocessing.scale(df["D"])

df["H_X_W"] = preprocessing.scale(df["H"]) * preprocessing.scale(df["W"])
df["H_X_D"] = preprocessing.scale(df["H"]) * preprocessing.scale(df["D"])

df["W_X_D"] = preprocessing.scale(df["W"]) * preprocessing.scale(df["D"])

# Join offensive interactions.
x = x.join([df["A_X_H"], df["A_X_W"], df["A_X_D"], df["H_X_W"], df["H_X_D"], df["W_X_D"]])

model = sm.OLS(dv, x).fit()
print(model.rsquared)
```

```
# More Interactions.
# Potential interaction effects to test.

# 2-way interactions.
df["Wounds_X_Save"] = preprocessing.scale(df["Wounds_X_model"]) * preprocessing.scale(df["Save"])
df["Wounds_X_Ward"] = preprocessing.scale(df["Wounds_X_model"]) * preprocessing.scale(df["Ward"])
df["Wounds_X_Bravery"] = preprocessing.scale(df["Wounds_X_model"]) * preprocessing.scale(df["Braver

df["Save_X_Bravery"] = preprocessing.scale(df["Save"]) * preprocessing.scale(df["Bravery"])
df["Save_X_Ward"] = preprocessing.scale(df["Save"]) * preprocessing.scale(df["Ward"])

df["Ward_X_Bravery"] = preprocessing.scale(df["Ward"]) * preprocessing.scale(df["Bravery"])

# Join offensive interactions.
x = x.join([df["Wounds_X_Save"], df["Wounds_X_Ward"], df["Wounds_X_Bravery"], df["Save_X_Bravery"],
            df["Save_X_Ward"], df["Ward_X_Bravery"]])

model = sm.OLS(dv, x).fit()
print(model.rsquared)
```



Warhammer Analysis:

Feature Engineering:

1) …and similarly for all 3-way and 4-way interaction effects

```python
# 3-way interactions.

df["A_X_H_X_W"] = preprocessing.scale(df["A"]) * preprocessing.scale(df["H"]) * preprocessing.scale
df["H_X_W_X_D"] = preprocessing.scale(df["H"]) * preprocessing.scale(df["W"]) * preprocessing.scale
df["H_X_A_X_D"] = preprocessing.scale(df["H"]) * preprocessing.scale(df["A"]) * preprocessing.scale


df["Wounds_X_Save_X_Ward"] = preprocessing.scale(df["Wounds_X_model"]) * preprocessing.scale(df["Sa
df["Save_X_Ward_X_Bravery"] = preprocessing.scale(df["Save"]) * preprocessing.scale(df["Ward"]) * p
df["Save_X_Wounds_X_Bravery"] = preprocessing.scale(df["Save"]) * preprocessing.scale(df["Wounds_X_


x = x.join([df["A_X_H_X_W"], df["H_X_W_X_D"], df["H_X_A_X_D"], df["Wounds_X_Save_X_Ward"],
            df["Save_X_Ward_X_Bravery"], df["Save_X_Wounds_X_Bravery"]])

model = sm.OLS(dv, x).fit()
print(model.rsquared)
```

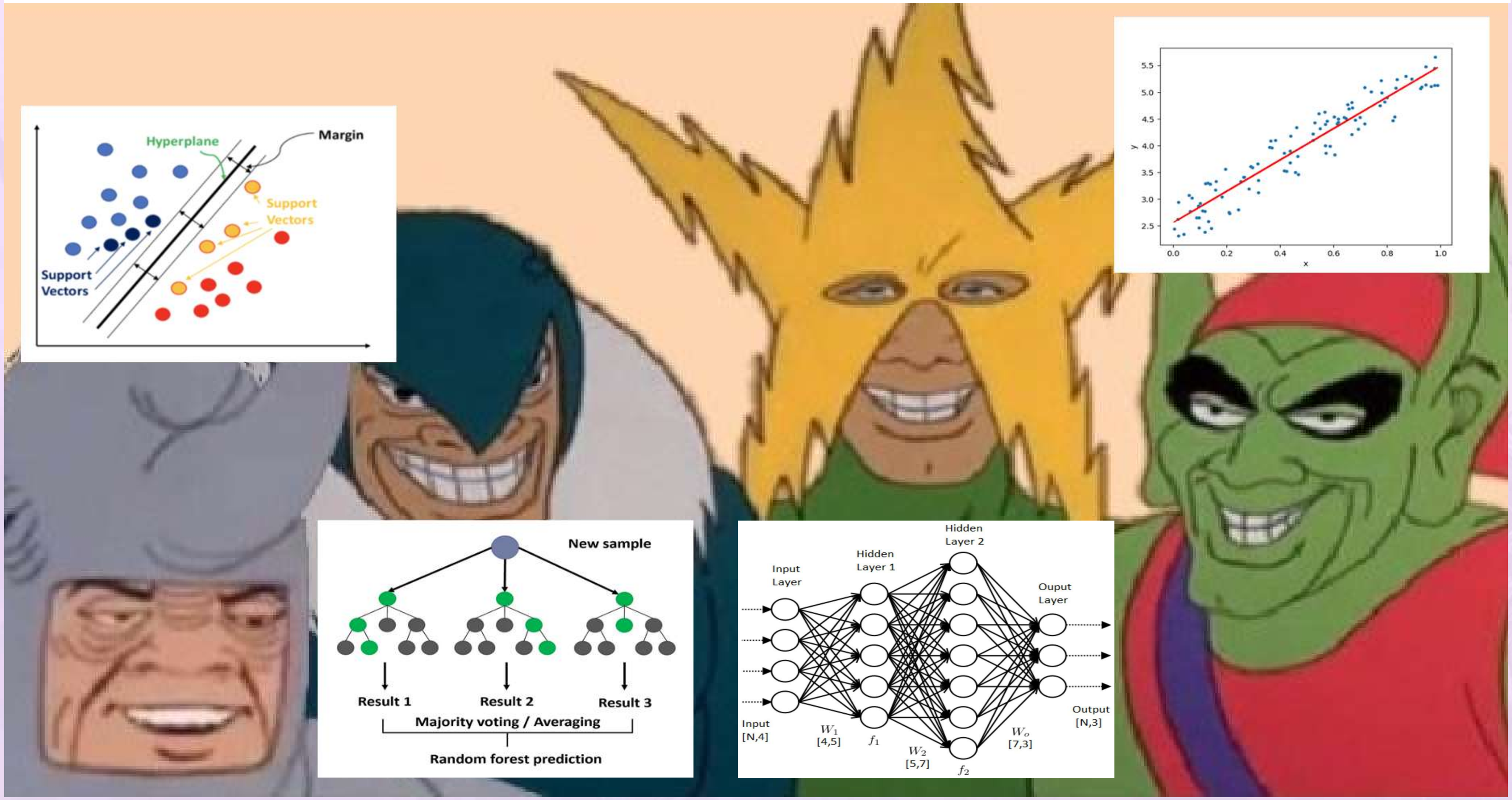
```python
# Four-way effect.
df["all_damage"] = preprocessing.scale(df["A"]) * preprocessing.scale(df["H"]) * preprocessing.scal

# Four-way effect.
df["all_defence"] = preprocessing.scale(df["Wounds_X_model"]) * preprocessing.scale(df["Save"]) * p

df["damage_X_defence"] = df["all_defence"] * df["all_damage"]

x = x.join([df["all_damage"], df["all_defence"], df["damage_X_defence"]])

model = sm.OLS(dv, x).fit()
print(model.rsquared)
```

Warhammer Analysis:

Data is ready to roll!!

Warhammer Analysis:

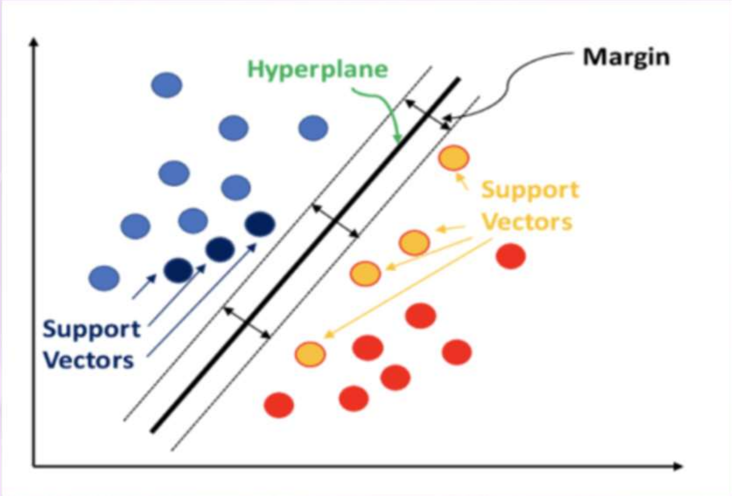
Me and the boys about to go find some insights

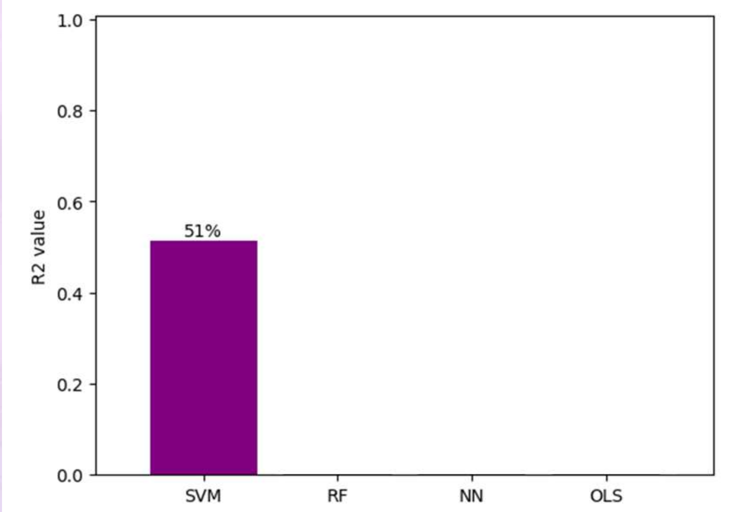


Warhammer Analysis:

Support Vector Machines:

- Common non-linear classifier (but also apparently does regression). Will find optimal (hyper)plane to separate data.
- Maximizes margins between classes based on nearby datum.
- Will predict new inputs based on there location within margins.

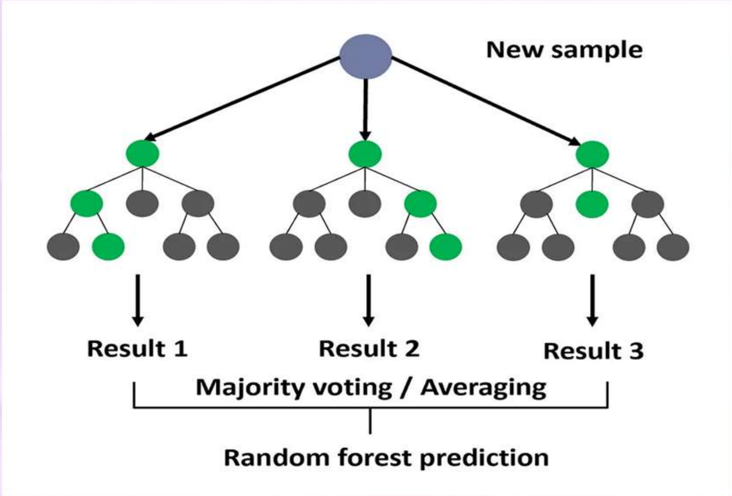


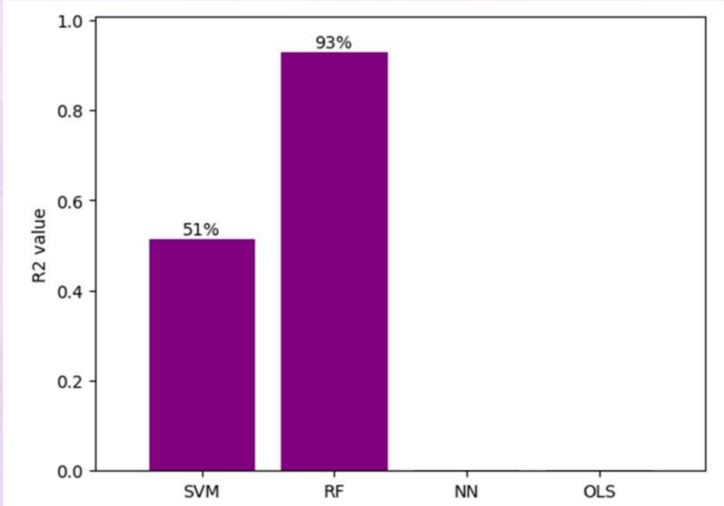


Warhammer Analysis:

Random Forest Boi ™:

- Uses decision tree methods to classify cases.
- Each individual tree is a poor predictor, however when aggregated (bagging/boosted) results are more accurate.
- Uses random selection of predictor variables.

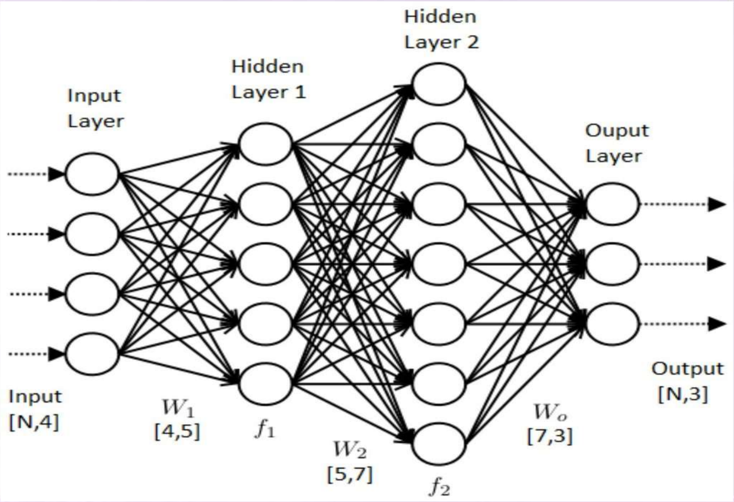


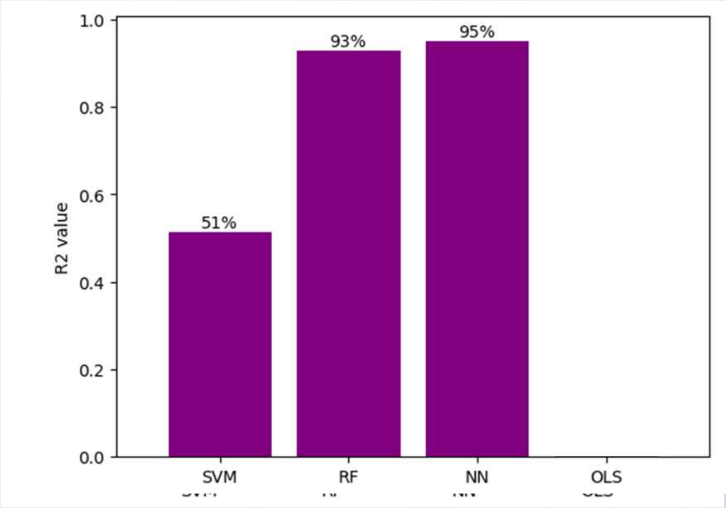


Warhammer Analysis:

Neural Network:

- Designed to mimic human synaptic processes.
- Networks contain input, hidden, and output layers.
- Each node adds weights and biases to the function. These forms an activation function.

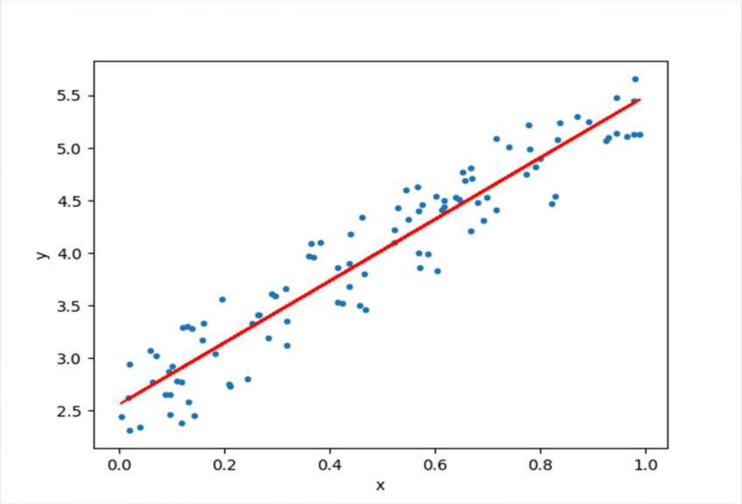


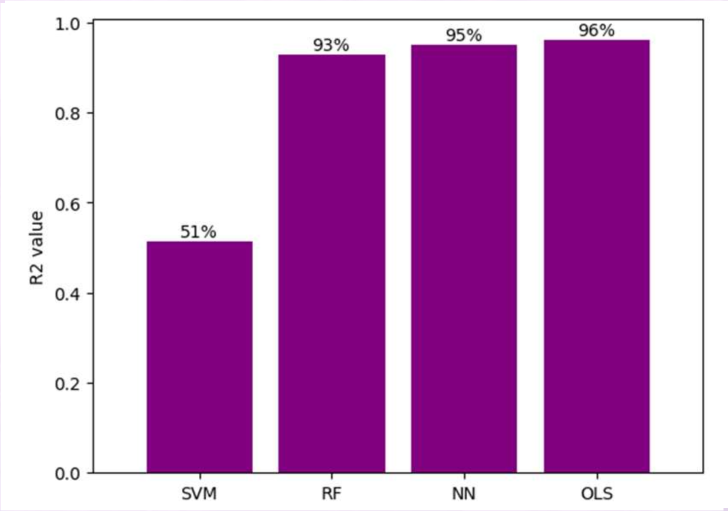


Warhammer Analysis:

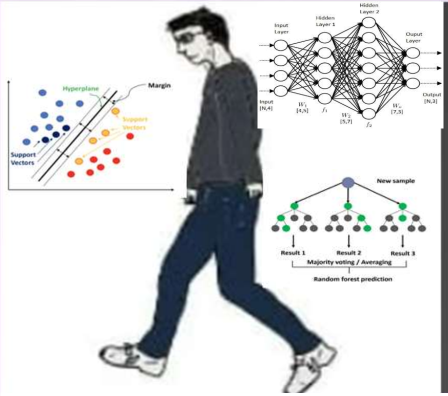
Linear Regression:

- Fits a line to the data using ordinary least squares.
- Minimises residuals.
- Can be used to model continues target variables.

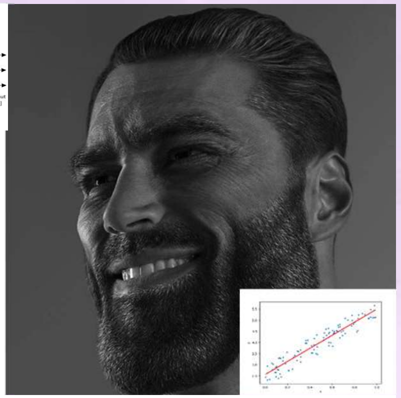




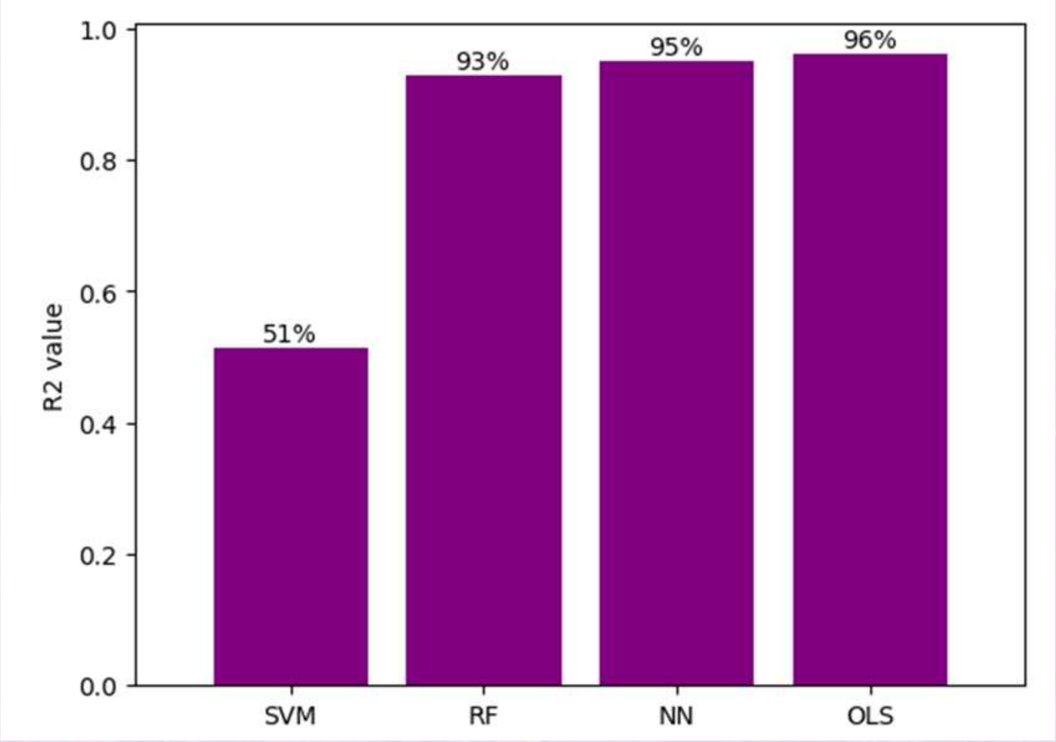
Warhammer Analysis:



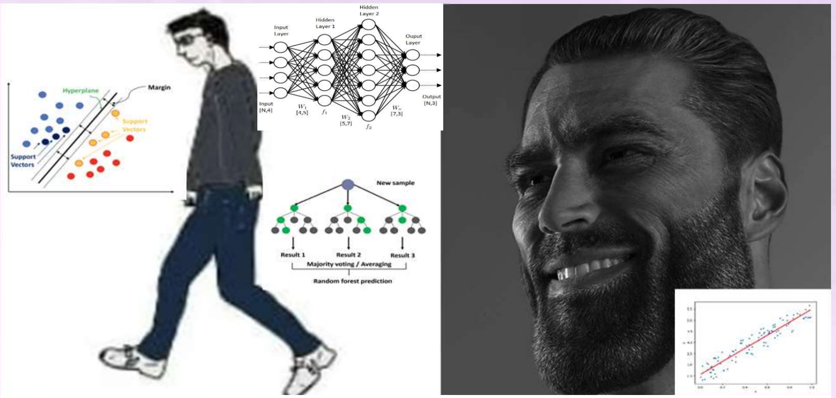
- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data



- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.

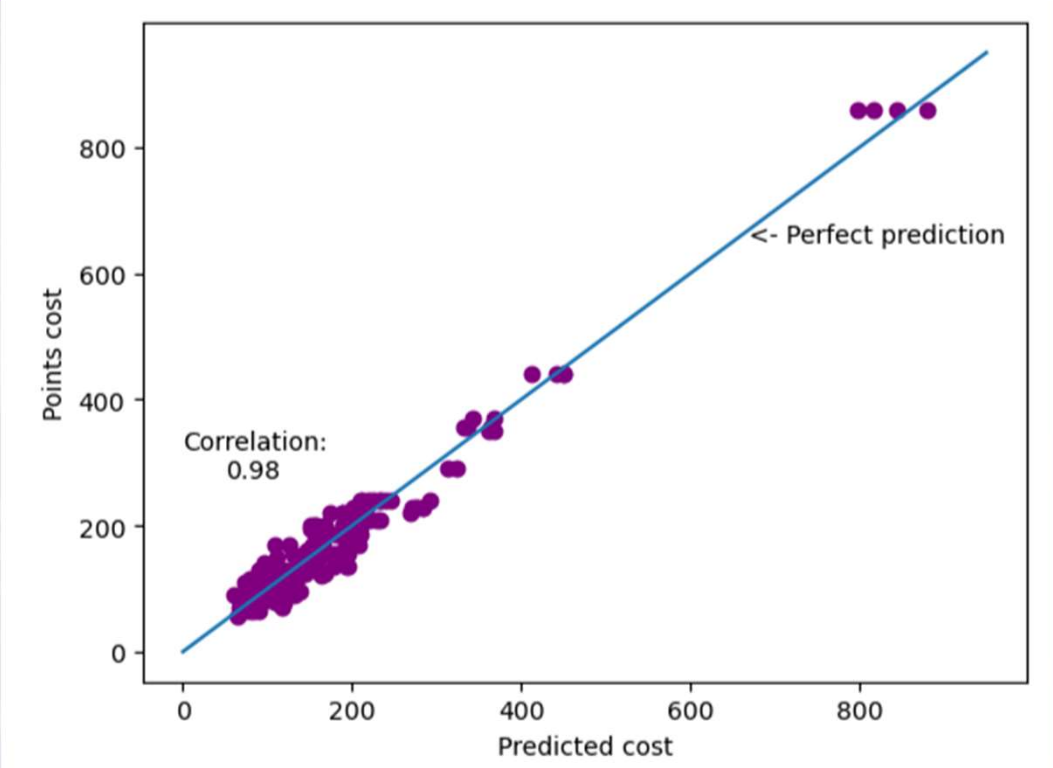


Warhammer Analysis:

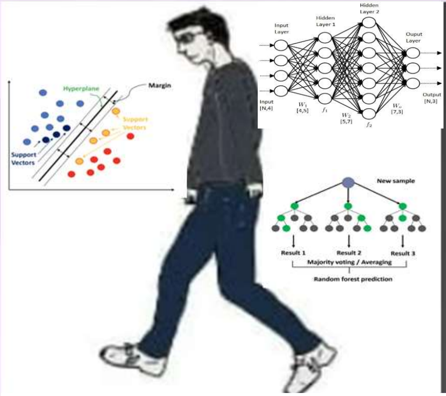


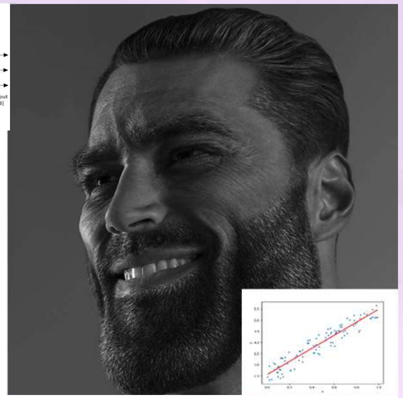
- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data

- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.



Warhammer Analysis:





- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data

- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.

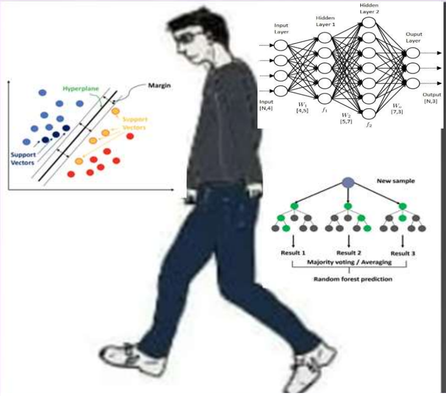
OLS Regression Results

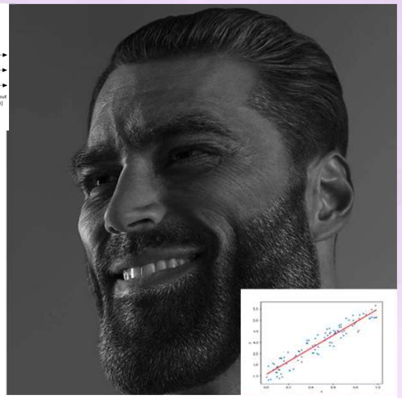
Dep. Variable:	Points_per_warscroll	R-squared:	0.961
Model:	OLS	Adj. R-squared:	0.950
Method:	Least Squares	F-statistic:	91.40
Date:	Sun, 12 Nov 2023	Prob (F-statistic):	2.33e-78
Time:	19:22:05	Log-Likelihood:	-802.52
No. Observations:	172	AIC:	1679.
Df Residuals:	135	BIC:	1795.
Df Model:	36		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-257.7503	79.904	-3.226	0.002	-415.775	-99.725
Range	-0.5074	0.516	-0.983	0.327	-1.528	0.513
A	-0.2141	1.767	-0.121	0.904	-3.708	3.280
H	1.9158	8.275	0.232	0.817	-14.450	18.282
W	12.1233	6.723	1.803	0.074	-1.173	25.419
R	1.7515	3.250	0.539	0.591	-4.676	8.179
D	8.6807	5.300	1.638	0.104	-1.802	19.163
Move	4.7176	0.940	5.021	0.000	2.859	6.576
Wounds	3.4505	1.366	2.527	0.013	0.750	6.151
Bravery	6.3408	2.589	2.449	0.016	1.220	11.462
Save	42.4442	7.310	5.807	0.000	27.988	56.901
Ward	4.0172	6.808	0.590	0.556	-9.447	17.481
unit_size	-0.0204	2.035	-0.010	0.992	-4.044	4.004
Spells	39.1308	4.786	8.176	0.000	29.666	48.596
slaanesh_dummy	13.3755	8.542	1.566	0.120	-3.519	30.270
Wounds_X_model	6.6395	1.249	5.318	0.000	4.170	9.109
A_X_H	1.0632	5.202	0.204	0.838	-9.226	11.352
A_X_W	0.8068	3.761	0.215	0.830	-6.630	8.244
A_X_D	-1.2522	4.158	-0.301	0.764	-9.476	6.971
H_X_W	8.4850	7.496	1.132	0.260	-6.341	23.311
H_X_D	0.8230	7.046	0.117	0.907	-13.111	14.757
W_X_D	-9.3248	4.338	-2.150	0.033	-17.904	-0.746
Wounds_X_Save	30.6198	4.762	6.429	0.000	21.201	40.039
Wounds_X_Ward	-5.4914	5.676	-0.968	0.335	-16.716	5.733
Wounds_X_Bravery	-12.9485	4.342	-2.982	0.003	-21.536	-4.361
Save_X_Bravery	12.6823	5.519	2.298	0.023	1.768	23.597
Save_X_Ward	-32.3678	18.296	-1.769	0.079	-68.551	3.816
Ward_X_Bravery	-9.1746	8.070	-1.137	0.258	-25.134	6.785
A_X_H_X_W	5.3967	7.183	0.751	0.454	-8.809	19.602
H_X_W_X_D	7.3550	11.418	0.644	0.521	-15.226	29.936
H_X_A_X_D	-6.2425	7.235	-0.863	0.390	-20.551	8.066
Wounds_X_Save_X_Ward	10.8077	13.314	0.812	0.418	-15.524	37.139
Save_X_Ward_X_Bravery	36.8268	16.152	2.280	0.024	4.883	68.771
Save_X_Wounds_X_Bravery	27.8105	5.503	5.054	0.000	16.928	38.693
all_damage	7.9642	10.762	0.740	0.461	-13.319	29.248
all_defence	7.6687	14.195	0.540	0.590	-20.405	35.743
damage_X_defence	0.3828	2.361	0.162	0.871	-4.287	5.053

Warhammer Analysis:

Negative Residuals = Under-costed Units





- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data

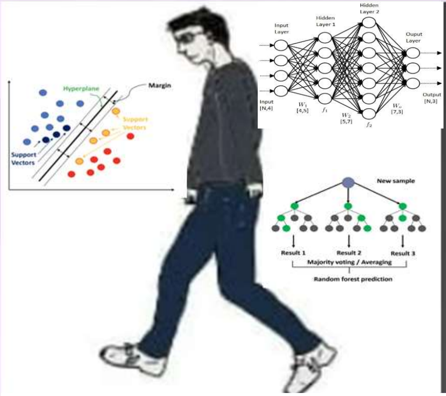
- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.

Out[21]:

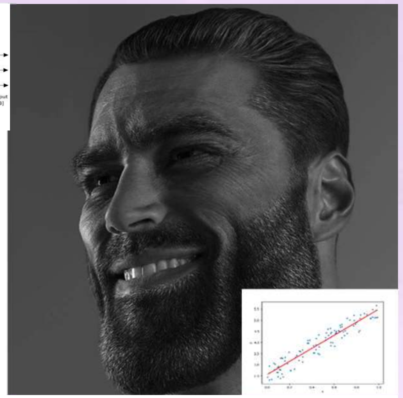
	unit_name	fittedvalues	Points_per_warscroll	residuals	resid_%
107	Chaos Spawn	117.82	70	-47.82	-68.31
106	Raptoryx	118.38	80	-38.38	-47.98
144	Chaos Warhounds	118.23	80	-38.23	-47.79
100	Furies	130.88	90	-40.88	-45.42
104	Mindstealer Sphiranx	137.24	95	-42.24	-44.46
117	Beasts of Chaos Tzaangor Shaman	194.91	135	-59.91	-44.38
151	Tuskgor Chariots	89.23	65	-24.23	-37.28
155	Ungor Raiders	108.95	80	-28.95	-36.19
24	Infernal Enrapturess	163.37	120	-43.37	-36.14
148	Dragon Ogors	167.17	125	-42.17	-33.74
99	Fomoroid Crusher	125.74	100	-25.74	-25.74
143	Centigors	106.59	85	-21.59	-25.40
122	Dragon Ogor Shaggoth	194.30	155	-39.30	-25.35
114	Soul Grinder	283.04	230	-53.04	-23.06
94	Chaos Warriors	268.29	220	-48.29	-21.95
129	Ungors	79.24	65	-14.24	-21.91
8	Syll'Esske	207.12	170	-37.12	-21.84
97	Chaos Chosen	291.17	240	-51.17	-21.32
54	Seeker Chariots	130.41	110	-20.41	-18.55
150	Razorgors	64.52	55	-9.52	-17.31

Warhammer Analysis:

Positive Residuals = Over-costed Units



- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data

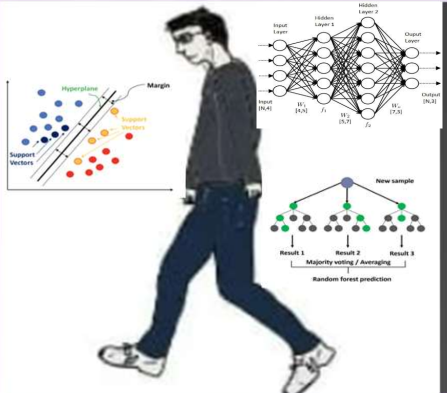


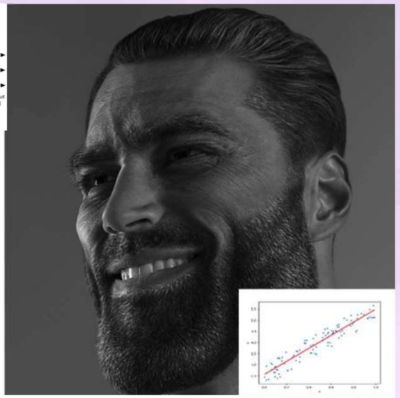
- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.

Out[22]:

	unit_name	fittedvalues	Points_per_warscroll	residuals	resid_%
48	Fiends	107.64	170	62.36	36.68
39	Daemonettes	72.20	110	37.80	34.36
96	Untamed Beasts	60.27	90	29.73	33.03
141	Bullgors	88.68	130	41.32	31.78
70	Chaos Lord	79.10	115	35.90	31.22
38	Blissbarb Archers	96.35	140	43.65	31.18
139	Bestigors	91.27	125	33.73	26.98
127	Grashrak Fellhoof	110.71	150	39.29	26.19
163	Chimera	151.42	200	48.58	24.29
136	Beasts of Chaos Tzaangor Skyfires	151.75	195	43.25	22.18
58	Seekers	101.76	130	28.24	21.72
126	Beastlord	74.58	95	20.42	21.49
75	Chaos Lord on Karkadrak	174.61	220	45.39	20.63
101	Gorebeast Chariot	91.75	115	23.25	20.22
26	Lord of Pain	103.95	130	26.05	20.04
167	Ghorgon	128.66	155	26.34	16.99
105	Ogroid Theridons	158.59	190	31.41	16.53
120	Doombull	92.99	110	17.01	15.46
93	Marauders	72.78	85	12.22	14.38
91	Marauder Horsemen	90.34	105	14.66	13.96

Warhammer Analysis:





- Uses massive computing
- Is difficult to interpret
- Tuning is time-consuming
- Is cringe
- Algos are still being optimised
- Requires massive data

- Can literally be calculated by hand
- coefs tell you how to interpret
- What are compute costs?
- Perfected by Gauss 1795
- Requires small samples
- Is the best linear unbiased est.

Positive Residuals = Over-costed Units

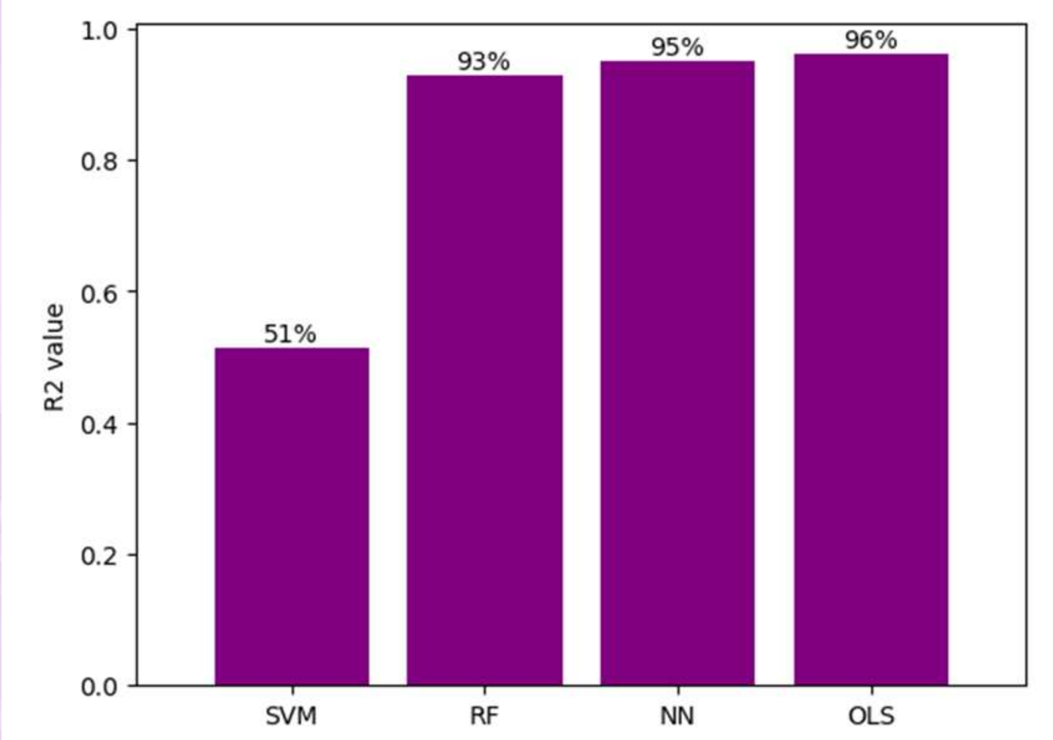
Out[22]:

	unit_name	fittedvalues	Points_per_warscroll	residuals	resid_%
48	Fiends	107.64	170	62.36	36.68
39	Daemonettes	72.20	110	37.80	34.36
96	Untamed Beasts	60.27	90	29.73	33.03
141	Bullgors	88.68	130	41.32	31.78
70	Chaos Lord	79.10	115	35.90	31.22
38	Blissbarb Archers	96.35	140	43.65	31.18
139	Bestigors	91.27	125	33.73	26.98
127	Grashrak Fellhoof	110.71	150	39.29	26.19
163	Chimera	151.42	200	48.58	24.29
136	Beasts of Chaos Tzaangor Skyfires	151.75	195	43.25	22.18
58	Seekers	101.76	130	28.24	21.72
126	Beastlord	74.58	95	20.42	21.49
75	Chaos Lord on Karkadrak	174.61	220	45.39	20.63
101	Gorebeast Chariot	91.75	115	23.25	20.22
26	Lord of Pain	103.95	130	26.05	20.04
167	Ghorgon	128.66	155	26.34	16.99
105	Ogroid Theridons	158.59	190	31.41	16.53
120	Doombull	92.99	110	17.01	15.46
93	Marauders	72.78	85	12.22	14.38
91	Marauder Horsemen	90.34	105	14.66	13.96

Warhammer Analysis:

Question:

Why did the OLS model outperform the more advanced ML approaches?



Questions:

Cloudy ⛅ (estrogen angel)
@oncloud_e

i have never kissed a boy and at this point i'm afraid to. what if he tries to teach me about Warhammer 40k

5:16 AM · May 31, 2022

843 Reposts **156** Quotes **12.9K** Likes **171** Bookmarks