



IMPLEMENTATIONS OF ALGORITHMS

A Report Submitted in Fulfilment of the Project Work Requirements of the Subject Business Analytics.

MBA Tech IT - Semester VIII

Academic Year 2022-23

GROUP 3

Rohit Bhatia
IO06

Rohini Bhattacharjee
IO08

Deepak Chaudhary
IO10

Osama Dalwai
IO12

Varad Gandhi
IO14

Aditya Huria
IO21

ABOUT THE DATASET

For the implementation of the project, 2 datasets have been selected, each contributing to the use of different methods in Analytics. Both the datasets are a mix of quantitative and qualitative information thus providing a stage to perform all the different Analysis procedures on them and gain further insights on the data.

The first dataset used is ‘FIFA 19 Players’ dataset wherein analysis methods such as Regression, Clustering, Discriminant Analysis, Factoring Analysis are applied in order to predict/understand the effect of the independent variables on the target variable/variables and get valuable insights from the data. In addition to it, the correlation between different variables is also analysed through the Correlation analysis.

Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	Preferred	Internatio	Weak Foo	Skill Move	Position	Height	Weight	Crossing	Finishing	HeadingA	ShortPass	Volleys	Dr
L. Messi	31	Argentina	94	94	FC Barcelona	€110.5M	€565K	2202	Left	5	4	4	RF	5'7	159lbs	84	95	70	90	86	
Cristiano Ronaldo	33	Portugal	94	94	Juventus	€77M	€405K	2228	Right	5	4	5	ST	6'2	183lbs	84	94	89	81	87	
Neymar Jr	26	Brazil	92	93	Paris Saint-Germain	€118.5M	€290K	2143	Right	5	5	5	LW	5'9	150lbs	79	87	62	84	84	
De Gea	27	Spain	91	93	Manchester United	€72M	€260K	1471	Right	4	3	1	GK	6'4	168lbs	17	13	21	50	13	
K. De Bruyne	27	Belgium	91	92	Manchester City	€102M	€355K	2281	Right	4	5	4	RCM	5'11	154lbs	93	82	55	92	82	
E. Hazard	27	Belgium	91	91	Chelsea	€93M	€340K	2142	Right	4	4	4	LF	5'8	163lbs	81	84	61	89	80	
L. Modrić	32	Croatia	91	91	Real Madrid	€67M	€420K	2280	Right	4	4	4	RCM	5'8	146lbs	86	72	55	93	76	
L. Suárez	31	Uruguay	91	91	FC Barcelona	€80M	€455K	2346	Right	5	4	3	RS	6'0	190lbs	77	93	77	82	88	
Sergio Ramos	32	Spain	91	91	Real Madrid	€51M	€380K	2201	Right	4	3	3	RCB	6'0	181lbs	66	60	91	78	66	
J. Oblak	25	Slovenia	90	93	Atlético Madrid	€68M	€94K	1331	Right	3	3	1	GK	6'2	192lbs	13	11	15	29	13	
R. Lewandowski	29	Poland	90	90	FC Bayern München	€77M	€205K	2152	Right	4	4	4	ST	6'0	176lbs	62	91	85	83	89	
T. Kroos		Germany	90	90	Real Madrid	€76.5M	€355K	2190	Right	4	5	3	LCM	6'0	168lbs	88	76	54	92	82	
D. Godin	32	Uruguay	90	90	Atlético Madrid	€44M	€125K	1946	Right	3	3	2	CB	6'2	172lbs	55	42	92	79	47	
David Silva	32	Spain	90	90	Manchester City	€60M	€285K	2115	Left	4	2	4	LCM	5'8	148lbs	84	76	54	93	82	
N. Kanté	27	France	89	90	Chelsea	€63M	€225K	2189	Right	3	3	2	LDM	5'6	159lbs	68	65	54	86	56	
P. Dybala	24	Argentina	89	94	Juventus	€89M	€205K	2092	Left	3	3	4	LF	5'10	165lbs	82	84	68	87	88	
H. Kane	24	England	89	91	Tottenham Hotspur	€83.5M	€205K	2165	Right	3	4	3	ST	6'2	196lbs	75	94	85	80	84	
A. Griezmann	27	France	89	90	Atlético Madrid	€78M	€145K	2246	Left	4	3	4	CAM	5'9	161lbs	82	90	84	83	87	
M. ter Stegen	26	Germany	89	92	FC Barcelona	€58M	€240K	1328	Right	3	4	1	GK	6'2	187lbs	15	14	11	36	14	
T. Courtois	26	Belgium	89	90	Real Madrid	€53.5M	€240K	1311	Left	4	2	1	GK	6'6	212lbs	14	14	13	33	12	
Sergio Busquets	29	Spain	89	89	FC Barcelona	€51.5M	€315K	2065	Right	4	3	3	CDM	6'2	168lbs	62	67	68	89	44	
E. Cavani	31	Uruguay	89	89	Paris Saint-Germain	€60M	€200K	2161	Right	4	4	3	LS	6'1	170lbs	70	89	89	78	90	
M. Neuer	32	Germany	89	89	FC Bayern München	€38M	€130K	1473	Right	5	4	1	GK	6'4	203lbs	15	13	25	55	11	
S. Agüero	30	Argentina	89	89	Manchester City	€64.5M	€300K	2107	Right	4	4	4	ST	5'8	154lbs	70	93	77	81	85	
G. Chiellini	33	Italy	89	89	Juventus	€27M	€215K	1841	Left	4	3	2	LCB	6'2	187lbs	58	33	83	59	45	
K. Mbappé	19	France	88	95	Paris Saint-Germain	€81M	€100K	2118	Right	3	4	5	RM	5'10	161lbs	77	88	77	82	78	
M. Salah	26	Egypt	88	89	Liverpool	€69.5M	€255K	2146	Left	3	3	4	RM	5'9	157lbs	78	90	59	82	73	
Casemiro	26	Brazil	88	90	Real Madrid	€59.5M	€285K	2170	Right	3	3	2	CDM	6'1	185lbs	52	59	76	85	53	

The second dataset used is ‘Google Play Store’ dataset wherein Classification techniques have been implemented in order to classify different apps as per their ratings. The target variable in this case is the Ratings parameter & all the other variables perform the trivial part to classify apps into different Rating tiers.

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	High	159 19M		10000	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
Coloring book moana	ART_AND_DESIGN	Medium	967 14M		500000	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
U Launcher Lite &™ FREE Live Cool Themes, Hide Apps	ART_AND_DESIGN	High	87510 8.7M		5000000	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
Sketch - Draw & Paint	ART_AND_DESIGN	High	215644 25M		50000000	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	High	967 2.8M		100000	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
Paper flowers instructions	ART_AND_DESIGN	High	167 5.6M		50000	Free	0	Everyone	Art & Design	March 26, 2017	1	2.3 and up
Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	Medium	178 19M		50000	Free	0	Everyone	Art & Design	April 26, 2018	1.1	4.0.3 and up
Infinite Painter	ART_AND_DESIGN	High	36815 29M		1000000	Free	0	Everyone	Art & Design	June 14, 2018	6.1.61.1	4.2 and up
Garden Coloring Book	ART_AND_DESIGN	High	13791 33M		1000000	Free	0	Everyone	Art & Design	September 20, 2017	2.9.2	3.0 and up
Kids Paint Free - Drawing Fun	ART_AND_DESIGN	High	121 3.1M		10000	Free	0	Everyone	Art & Design;Creativity	July 3, 2018	2.8	4.0.3 and up
Text on Photo - Fontee	ART_AND_DESIGN	High	13880 28M		1000000	Free	0	Everyone	Art & Design	October 27, 2017	1.0.4	4.1 and up
Name Art Photo Editor - Focus n Filters	ART_AND_DESIGN	High	8788 12M		1000000	Free	0	Everyone	Art & Design	July 31, 2018	1.0.15	4.0 and up
Tattoo Name On My Photo Editor	ART_AND_DESIGN	High	44829 20M		10000000	Free	0	Teen	Art & Design	April 2, 2018	3.8	4.1 and up
Mandala Coloring Book	ART_AND_DESIGN	High	4326 21M		100000	Free	0	Everyone	Art & Design	June 26, 2018	1.0.4	4.4 and up
3D Color Pixel by Number - Sandbox Art Coloring	ART_AND_DESIGN	High	1518 37M		100000	Free	0	Everyone	Art & Design	August 3, 2018	1.2.3	2.3 and up
Photo Designer - Write your name with shapes	ART_AND_DESIGN	High	3632 5.5M		500000	Free	0	Everyone	Art & Design	July 31, 2018	3.1	4.1 and up
350 Diy Room Decor Ideas	ART_AND_DESIGN	High	27 17M		10000	Free	0	Everyone	Art & Design	November 7, 2017	1	2.3 and up
FlipaClip - Cartoon animation	ART_AND_DESIGN	High	194216 39M		5000000	Free	0	Everyone	Art & Design	August 3, 2018	2.2.5	4.0.3 and up
ibis Paint X	ART_AND_DESIGN	High	224399 31M		10000000	Free	0	Everyone	Art & Design	July 30, 2018	5.5.4	4.1 and up
Logo Maker - Small Business	ART_AND_DESIGN	High	450 14M		100000	Free	0	Everyone	Art & Design	April 20, 2018	4	4.1 and up
Boys Photo Editor - Six Pack & Men's Suit	ART_AND_DESIGN	High	654 12M		100000	Free	0	Everyone	Art & Design	March 20, 2018	1.1	4.0.3 and up
Superheroes Wallpapers 4K Backgrounds	ART_AND_DESIGN	High	7699 4.2M		500000	Free	0	Everyone 10+	Art & Design	July 12, 2018	2.2.6.2	4.0.3 and up
HD Mickey Minnie Wallpapers	ART_AND_DESIGN	High	118 23M		50000	Free	0	Everyone	Art & Design	July 7, 2018	1.1.3	4.1 and up
Harley Quinn wallpapers HD	ART_AND_DESIGN	High	192 6.0M		10000	Free	0	Everyone	Art & Design	April 25, 2018	1.5	3.0 and up
ColorIt - Drawing & Coloring	ART_AND_DESIGN	High	7774 25M		500000	Free	0	Everyone	Art & Design;Creativity	October 11, 2017	1.0.8	4.0.3 and up
Animated Photo Editor	ART_AND_DESIGN	High	203 6.1M		100000	Free	0	Everyone	Art & Design	March 21, 2018	1.03	4.0.3 and up
Pencil Sketch Drawing	ART_AND_DESIGN	Medium	136 4.6M		10000	Free	0	Everyone	Art & Design	July 12, 2018	6	2.3 and up
Easy Realistic Drawing Tutorial	ART AND DESIGN	High	223 4.2M		100000	Free	0	Everyone	Art & Design	August 22, 2017	1	2.3 and up

CORRELATION

A statistical measure, represented as a number that reflects the extent and direction of a relationship between two or more variables. It denotes an association between two or more variables which describes the changes observed in the values of a variable if a changes are made to another variable.

There are 3 main types of correlation. The first being Positive linear correlation, where if the variable on the x axis increases, the variable on the Y axis also increases. The second is Negative Linear Correlation, where as one variable increases in value, the other decreases. The third is Non-Linear Correlation, where there is a relationship between the variables but it is not linear.

Correlations

		Wage	Value
Wage	Pearson Correlation	1	.830**
	Sig. (2-tailed)		<.001
	N	6195	6195
Value	Pearson Correlation	.830**	1
	Sig. (2-tailed)	<.001	
	N	6195	6195

** . Correlation is significant at the 0.01 level (2-tailed).

➤ Nonparametric Correlations

		Wage	Value		
Spearman's rho	Wage	Correlation Coefficient	1.000	.688**	
		Sig. (2-tailed)		.	<.001
		N	6195	6195	
	Value	Correlation Coefficient	.688**	1.000	
		Sig. (2-tailed)	<.001		.
		N	6195	6195	

** . Correlation is significant at the 0.01 level (2-tailed).

Interpretation:

In parametric correlation, it is assumed that the variables are linearly correlated and the Pearson's Correlation values are determined on the basis of this assumption. The Pearson's Correlation value in this case, while testing the correlation between the variables 'Wages' which is the amount actually paid to the players and 'Value' which is calculated value of the player based on their capabilities, is 0.830. From this we can conclude that Wage shares 69% of its variability with Value.

Since significance (P value) is less than 0.001 which is less than the level of significance (0.01). We can say that the correlation is significant

Non-parametric Correlation is generally used where the data does not follow a normal distribution. Here, the Spearman's Rho is calculated and found to be 0.688 and since this value is close to 1, we can say that the two variables are significantly correlated.

Correlations				
Descriptive Statistics				
	Mean	Std. Deviation	N	
LongShots	56.09	18.572	6195	
ShotPower	63.92	15.819	6195	

Correlations				
		LongShots	ShotPower	
LongShots	Pearson Correlation	1	.874**	
	Sig. (2-tailed)		<.001	
	N	6195	6195	
ShotPower	Pearson Correlation	.874**	1	
	Sig. (2-tailed)	<.001		
	N	6195	6195	

** . Correlation is significant at the 0.01 level (2-tailed).

➔ Nonparametric Correlations

Correlations				
			LongShots	ShotPower
Spearman's rho	LongShots	Correlation Coefficient	1.000	.817**
		Sig. (2-tailed)	.	<.001
		N	6195	6195
	ShotPower	Correlation Coefficient	.817**	1.000
		Sig. (2-tailed)	<.001	.
		N	6195	6195

** . Correlation is significant at the 0.01 level (2-tailed).

Interpretation:

Correlation is shown between variables 'Longshots' and 'Shotpower'. Longshot refers to the distance of the shot in football and shotpower refers to the strength with which the ball is kicked. The aim of conducting this test was to verify if there's a correlation between the two.

The Pearson's correlation was calculated and found to be 0.874 and from this we can conclude that Longshots share 76.4% of their variability with shot power.

Since P value is less than 0.001 and is therefore less than the level of significance at 0.01, we can say that the data is statistically significant.

In the Non-parametric correlation table, it is found that Spearman's Rho is 0.817 and since this value is close to 1, we can say that the two variables are significantly correlated.

REGRESSION

It is a statistical method that identifies the relationship between a dependent variable and one or more independent variables. The regression equation obtained, can be used to predict the values of the dependent variables for any given values of the independent variable.

Based on the number of independent variables, regression is classified into Linear Regression and Multiple Linear Regression (MLR). If regression is done on one dependent variable and one Independent variable, it is termed as Linear Regression. If Regression is done one dependent and more than 1 independent variables, then it is termed as Multiple Linear Regression (MLR).

LINEAR REGRESSION

Descriptive Statistics			
	Mean	Std. Deviation	N
Penalties	54.55	15.381	6195
LongShots	56.09	18.572	6195

Model Summary ^b										
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	Change Statistics				Durbin-Watson
						F Change	df1	df2	Sig. F Change	
1	.774 ^a	.599	.599	9.739	.599	9257.104	1	6193	<.001	1.949
a. Predictors: (Constant), LongShots										
b. Dependent Variable: Penalties										

Interpretation:

Regression is done to find the relationship between the variables 'Penalties' which is the number of penalties shot by the player throughout their career and their ability to shoot further or 'Longshots'.

The mean number of penalties between all 6195 players is 54.55 and their mean Longshot is 56.09. Here the dependent variable is Penalties and the independent Variable is Longshots. Therefore, from the aim of this regression is to be able to predict the number of penalty shots taken by a player, if we knew his Longshot capacity.

From the Model Summary table above, we can see that the Durbin-Watson statistic is valued at 1.949, which falls under the acceptable range, so we can conclude that there isn't an autocorrelation problem with the data selected for regression analysis. The R sq. value from the table is 0.599, which can be interpreted as 59.9% of the changes in Y (Penalties) can be explained by a variation in X (Long Shots).

From this data, we can say that 59.9% of the penalty shots are dependent on the player's ability to shoot longer shots on the football field.

ANOVA ^a						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	878035.233	1	878035.233	9257.104	<.001 ^b
	Residual	587405.326	6193	94.850		
	Total	1465440.559	6194			

a. Dependent Variable: Penalties
b. Predictors: (Constant), LongShots

From the above ANOVA table, we can see that the P value is less than 0.001. Since a level of significance of 0.05 is maintained, which is greater than the P value, it can be concluded that the model is significant. Since the model is significant, we can say that there is a significant relationship between the variables.

Coefficients ^a											
Model		Unstandardized Coefficients		Standardized Coefficients			Correlations			Collinearity Statistics	
		B	Std. Error	Beta	t	Sig.	Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	18.587	.394		47.210	<.001					
	LongShots	.641	.007	.774	96.214	<.001	.774	.774	.774	1.000	1.000

a. Dependent Variable: Penalties

From the table above, we can formulate the regression equation.

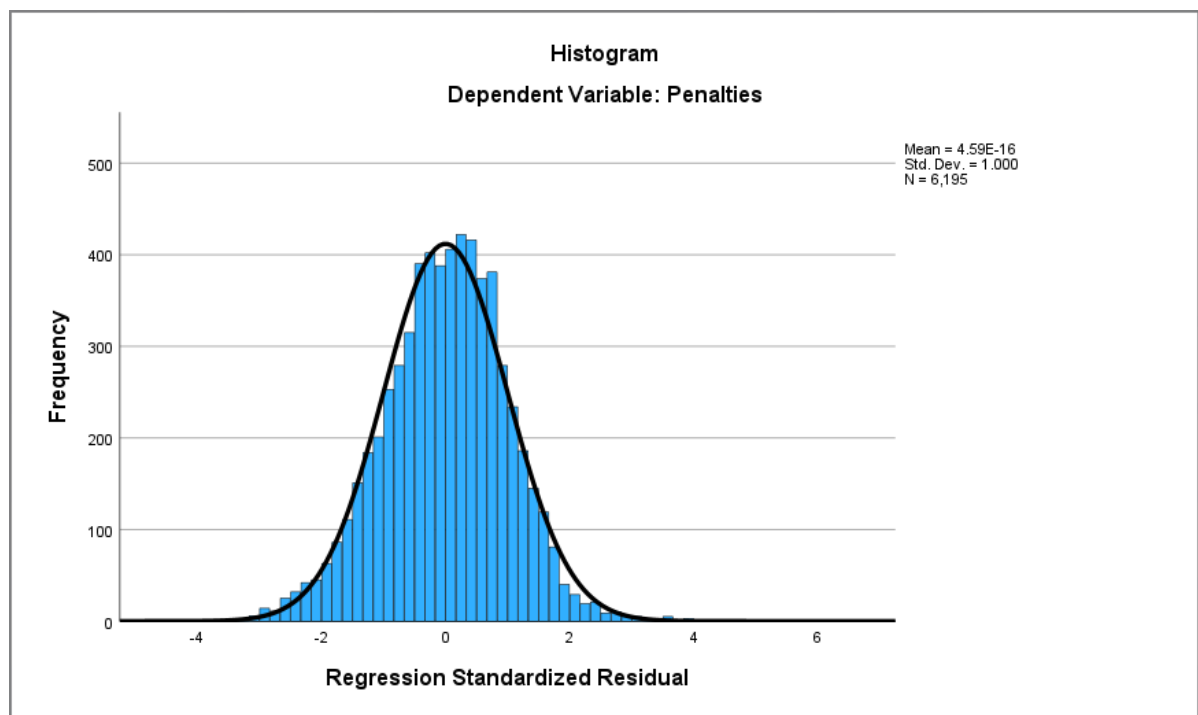
$$Y = 18.587 + 0.641 X$$

$$\text{or, (Penalties)} = 18.587 + 0.641 (\text{Longshot})$$

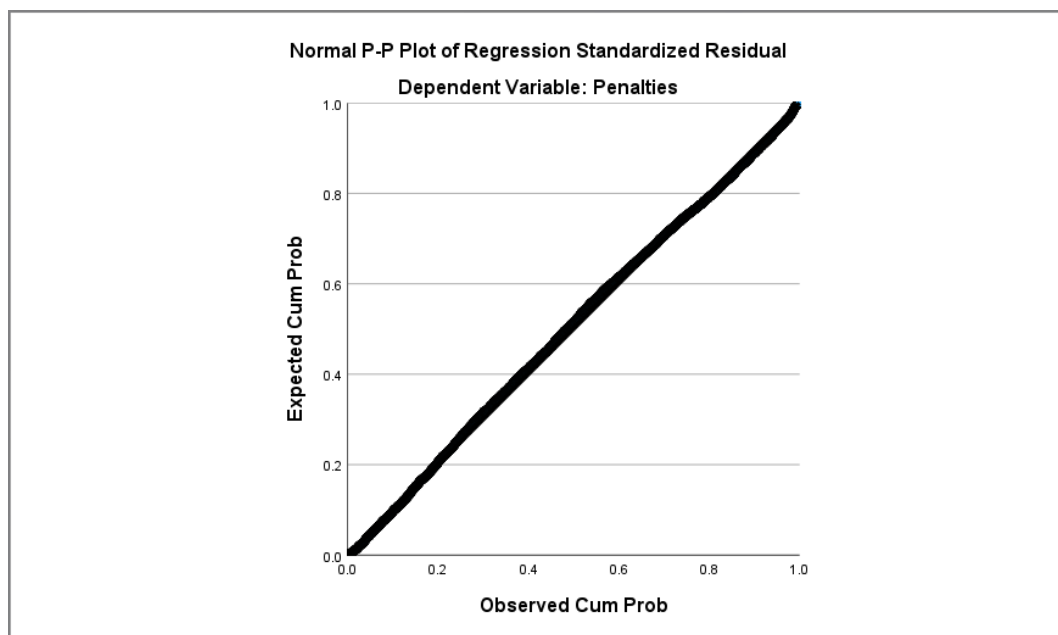
Here, 18.587 is the intercept value and 0.641 is the coefficient. So by plugging in values for the variables, we can use this equation to predict the number of penalty shots taken if we know their long shot capabilities.

For every 1 unit increase in the power of longshots, we can observe a 0.641 times increase in their penalty capabilities.

From this table, we can also see that the P value is less than 0.001. Since a level of significance of 0.05 is maintained, which is greater than the P value, it can be concluded that Longshots, as a variable, are significant to determining the number of penalties taken by a player.



Residuals are the difference between observed and predicted values. It can be seen that the graph is quite normally distributed, which proves the normality of the residuals.



A p-p plot is a plot to show the difference between the observed and predicted values. Since it can be observed to be quite linear, we can say that the model is quite accurate and can be used to predict the number of penalties of a player.

MULTIPLE LINEAR REGRESSION

Descriptive Statistics

	Mean	Std. Deviation	N
Agility	68.23	13.612	6195
Stamina	69.28	13.815	6195
Strength	68.51	11.932	6195

Interpretation:

Multiple Linear Regression is done to find the relationship between the variables 'Agility' of a player and their 'Strength' and 'Stamina'

The mean Agility between all 6195 players is 68.23, their mean Stamina is 69.28 and their mean Strength is 68.51. Here the dependent variable is Agility and the independent Variables are Strength and Stamina. Therefore, from the aim of this regression is to be able to predict the Agility of a player, if we knew his Stamina and Strength.

Model Summary^b

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	Change Statistics			Sig. F Change	Durbin-Watson
						F Change	df1	df2		
1	.720 ^a	.519	.519	9.441	.519	3342.091	2	6192	<.001	1.979

a. Predictors: (Constant), Strength, Stamina

b. Dependent Variable: Agility

From the Model Summary table above, we can see that the Durbin-Watson statistic is valued at 1.979, which falls under the acceptable range, so we can conclude that there isn't an autocorrelation problem with the data selected for regression analysis. The R sq. value from the table is 0.519, which can be interpreted as 51.9% of the changes in Y (Agility) can be explained by a variation in X1 (Stamina) and X2 (Strength).

ANOVA ^a						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	595732.570	2	297866.285	3342.091	<.001 ^b
	Residual	551866.418	6192	89.126		
	Total	1147598.987	6194			

a. Dependent Variable: Agility

b. Predictors: (Constant), Strength, Stamina

From the above ANOVA table, we can see that the P value is less than 0.001. Since a level of significance of 0.05 is maintained, which is greater than the P value, it can be concluded that the model is significant. Since the model is significant, we can say that there is a significant relationship between the variables.

Coefficients ^a											
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations			Collinearity Statistics	
		B	Std. Error	Beta			Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	75.849	.879		86.244	<.001					
	Stamina	.514	.009	.522	58.902	<.001	.465	.599	.519	.989	1.011
	Strength	-.631	.010	-.553	-62.453	<.001	-.500	-.622	-.550	.989	1.011

a. Dependent Variable: Agility

From the table above, we can formulate the following Regression Equation

$$Y = 75.849 + 0.514X_1 - 0.631X_2$$

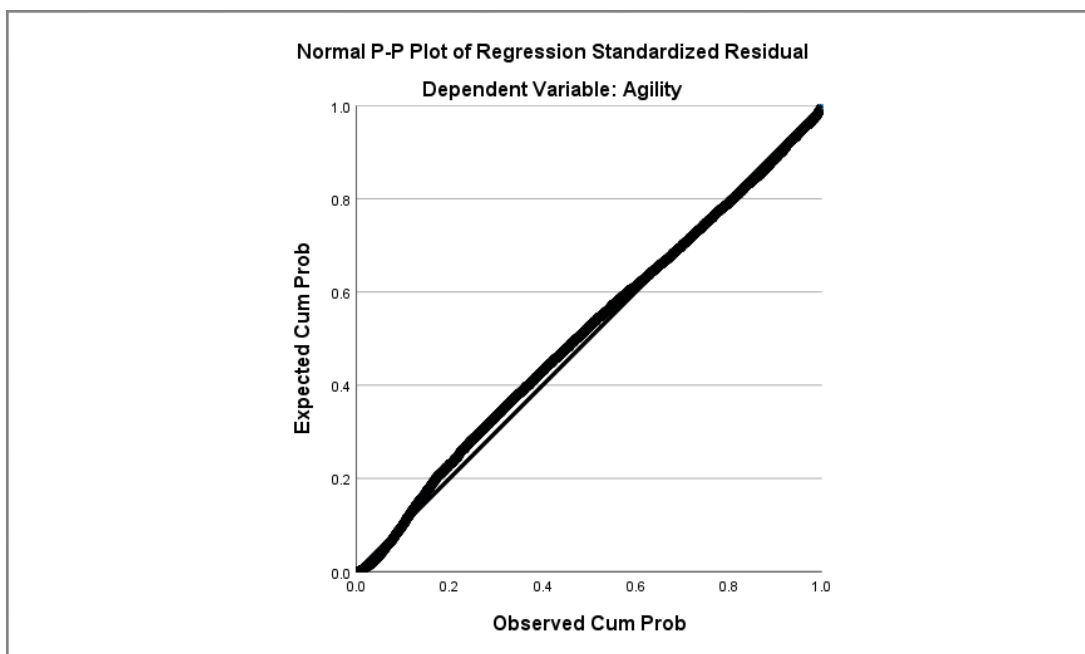
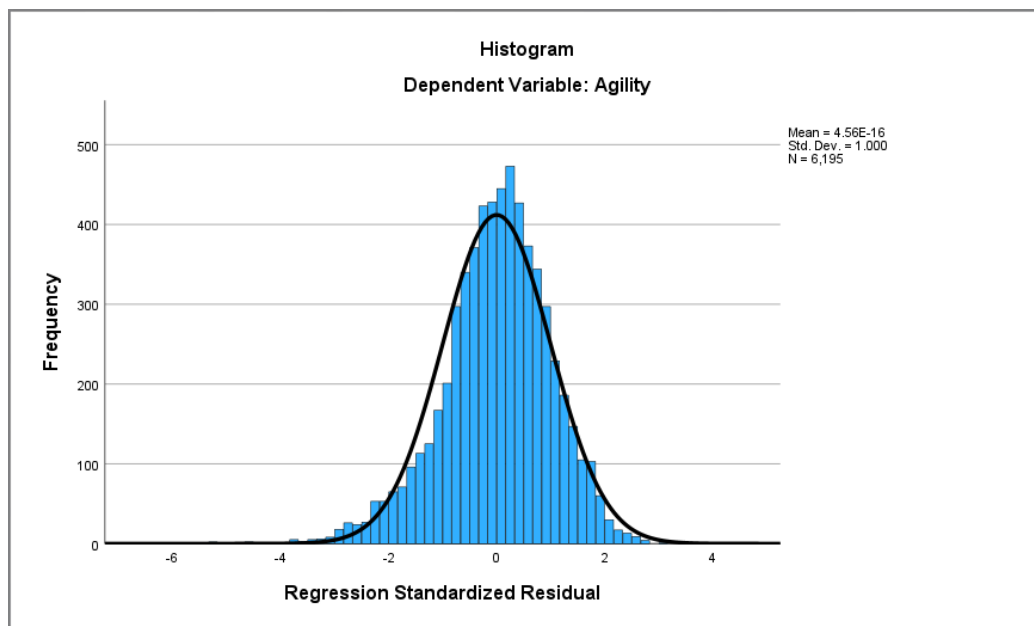
$$\text{or, (Agility)} = 75.849 + 0.514 (\text{Stamina}) - 0.631 (\text{Strength})$$

From the above equation, if we have the values of the Strength and Stamina of the player, we can predict their agility.

Therefore, For every 1 unit increase in player stamina, there is a 0.514 times increase in their Agility and keeping strength constant .

Therefore, For every 1 unit increase in player strength, there is a 0.631 times decrease in their agility keeping stamina constant.

From this table, we can also see that the P value of both Strength and Stamina are less than 0.001. Since a level of significance of 0.05 is maintained, which is greater than the P values, it can be concluded that Strength and Stamina, as variables, are significant to determining the Agility of a player.



Residuals are the difference between observed and predicted values. It can be seen that the graph is quite normally distributed, which proves the normality of the residuals.

A p-p plot is a plot to show the difference between the observed and predicted values. Since it can be observed to be quite linear, with a slight variation, we can say that the model is quite accurate but may show some errors and can be used to predict the number of Agility of a player.

CLUSTERING

Clustering is a technique used in data analysis and machine learning to group similar objects or data points together based on their characteristics or features. The goal of clustering is to identify meaningful patterns or structures in the data that can help to better understand the underlying relationships among the objects.

There are many different types of clustering algorithms, each with its own strengths and weaknesses. Some common clustering algorithms include k-means clustering, hierarchical clustering, and density-based clustering.

HIERARCHICAL CLUSTERING

Hierarchical clustering is a method of clustering that creates a hierarchy of clusters by recursively splitting or merging smaller clusters based on their similarity. In other words, it builds a tree-like structure (dendrogram) that represents the relationships among the clusters at different levels of granularity.

There are two main types of hierarchical clustering: agglomerative and divisive. Agglomerative clustering starts with each data point as a separate cluster and iteratively merges the closest pairs of clusters until all data points are grouped into a single cluster. Divisive clustering, on the other hand, starts with all data points in a single cluster and recursively divides it into smaller sub-clusters until each data point is in its own cluster.

Cluster					
▶ [DataSet1]					
Case Processing Summary ^{a,b}					
Valid		Cases Missing		Total	
N	Percent	N	Percent	N	Percent
6195	100.0	0	.0	6195	100.0
a. Squared Euclidean Distance used					
b. Average Linkage (Between Groups)					

Interpretation:

ANOVA Effect Sizes ^a				
		Point Estimate	95% Confidence Interval	
			Lower	Upper
Age	Eta-squared	.218	.198	.232
	Epsilon-squared	.215	.194	.228
	Omega-squared Fixed-effect	.215	.194	.228
	Omega-squared Random-effect	.009	.008	.010
Potential	Eta-squared	.506	.488	.519
	Epsilon-squared	.504	.486	.517
	Omega-squared Fixed-effect	.504	.486	.517
	Omega-squared Random-effect	.034	.032	.036
Value	Eta-squared	.884	.879	.887
	Epsilon-squared	.883	.878	.887
	Omega-squared Fixed-effect	.883	.878	.887
	Omega-squared Random-effect	.207	.199	.212
Wage	Eta-squared	.749	.739	.756
	Epsilon-squared	.748	.738	.755
	Omega-squared Fixed-effect	.748	.738	.755
	Omega-squared Random-effect	.093	.088	.096
Special	Eta-squared	.127	.108	.138
	Epsilon-squared	.122	.104	.134
	Omega-squared Fixed-effect	.122	.104	.134
	Omega-squared Random-effect	.005	.004	.005

There were 6195 cases in total, with no missing cases. This means that all cases were included in the analysis and there was no need to exclude any cases due to missing data or other issues.

Each variable's effect size is calculated for each variable in the ANOVA Effect Sizes chart using a variety of measures, including Eta-squared, Epsilon-squared, Omega-squared Fixed-effect, and Omega-squared Random-effect.

The amount of overall variation in the dependent variable that can be attributed to the independent variable is measured by eta-squared. A modified version of Eta-squared called epsilon-squared takes into consideration the number of groups being compared. Omega-squared Although less biased for small sample sizes than Eta-squared, fixed-effect is another measure of impact size. Omega-squared A measure of impact size known as the random-effect takes the data's random variability into consideration.

Interpretation:

Based on the ANOVA Effect Sizes table, all five variables have high effect sizes according to all four measures of effect size. For example, the Value variable has a very high Eta-squared of .884, indicating that a large proportion of the total variation in the dependent variable can be explained by the independent variable. In conclusion, the ANOVA Effect Sizes table provides evidence that there are strong relationships between the independent variables and the dependent variable, suggesting that the differences between the groups are not only statistically significant, but also practically significant.

ANOVA						
		Sum of Squares	df	Mean Square	F	Sig.
Age	Between Groups	21062.827	29	726.304	59.385	<.001
	Within Groups	75400.741	6165	12.230		
	Total	96463.568	6194			
Potential	Between Groups	69550.216	29	2398.283	217.950	<.001
	Within Groups	67838.457	6165	11.004		
	Total	137388.673	6194			
Value	Between Groups	372928.862	29	12859.616	1616.692	<.001
	Within Groups	49038.123	6165	7.954		
	Total	421966.985	6194			
Wage	Between Groups	4965900.336	29	171237.943	634.908	<.001
	Within Groups	1662731.285	6165	269.705		
	Total	6628631.621	6194			
Special	Between Groups	40371703.760	29	1392127.716	30.803	<.001
	Within Groups	278625463.62	6165	45194.722		
	Total	318997167.38	6194			

The Sum of Squares represents the variation in the data that is attributable to a particular variable, and the Degrees of Freedom represent the number of observations minus the number of parameters estimated in the model. The Mean Square is the Sum of Squares divided by the Degrees of Freedom.

The F statistic in the table represents the ratio of the between-group variation to the within-group variation. A high F value indicates that the between-group variation is large compared to the within-group variation, which suggests that there are significant differences between the groups being compared.

Interpretation:

Based on the ANOVA table, all five variables have a low p-value (<0.001) and a high F value, indicating that the differences between the groups are statistically significant. For example, the Wage variable has a large F value of 634.908 and a very small p-value

Cluster Membership	
Case	3 Clusters
1	1
2	1
3	1
4	2
5	1
6	1
7	1
8	1
9	1
10	2
11	1
12	3
13	1
14	1
15	1
16	1
17	1

(<0.001), suggesting that there are significant differences in wages between the groups being compared.

Cluster membership refers to the assignment of each case or observation in a dataset to a particular cluster in a hierarchical clustering analysis. After performing the analysis and creating a dendrogram, researchers often choose to cut the tree at a certain level or distance threshold to create a set number of clusters. The resulting clusters will have a membership list of which cases belong to which cluster.

K-MEANS CLUSTERING

A clustering algorithm known as K-means attempts to divide a set of data points into K clusters. Iteratively updating the centroids based on the average of the data points in each cluster, the method first randomly initialises K cluster centroids before assigning each data point to the closest centroid.

Here is a step-by-step overview of the K-means algorithm:

1. Choose the number of clusters K that you want to create.
2. Initialise K centroids randomly. This can be done by selecting K random data points from the dataset.
3. Assign each data point to the nearest centroid based on the Euclidean distance between the data point and the centroid.
4. For each cluster, compute the mean of the data points assigned to it. This will be the new centroid for that cluster.
5. Repeat steps 3 and 4 until the centroids no longer change, or until a maximum number of iterations is reached.
6. Once the algorithm has converged, the data points are clustered based on the final positions of the centroids.

Quick Cluster

Initial Cluster Centers

	Cluster		
	1	2	3
Age	31	21	18
Overall	91	67	65
Potential	91	78	86
Value	80.0	1.1	1.0
Wage	455	1	3
Special	2346	1701	908

Iteration History^a

	Change in Cluster Centers		
Iteration	1	2	3
1	350.617	106.409	248.176
2	127.278	13.344	70.850
3	60.557	24.457	24.470
4	31.031	21.054	.000
5	16.882	15.151	13.326
6	10.722	10.390	8.908
7	5.768	6.038	5.834
8	4.219	4.963	6.867
9	3.582	3.989	3.756
10	2.726	3.226	3.311

a. Iterations stopped because the maximum number of iterations was performed. Iterations failed to converge. The maximum absolute coordinate change for any center is 3.310. The current iteration is 10. The minimum distance between initial centers is 793.051.

Interpretation:

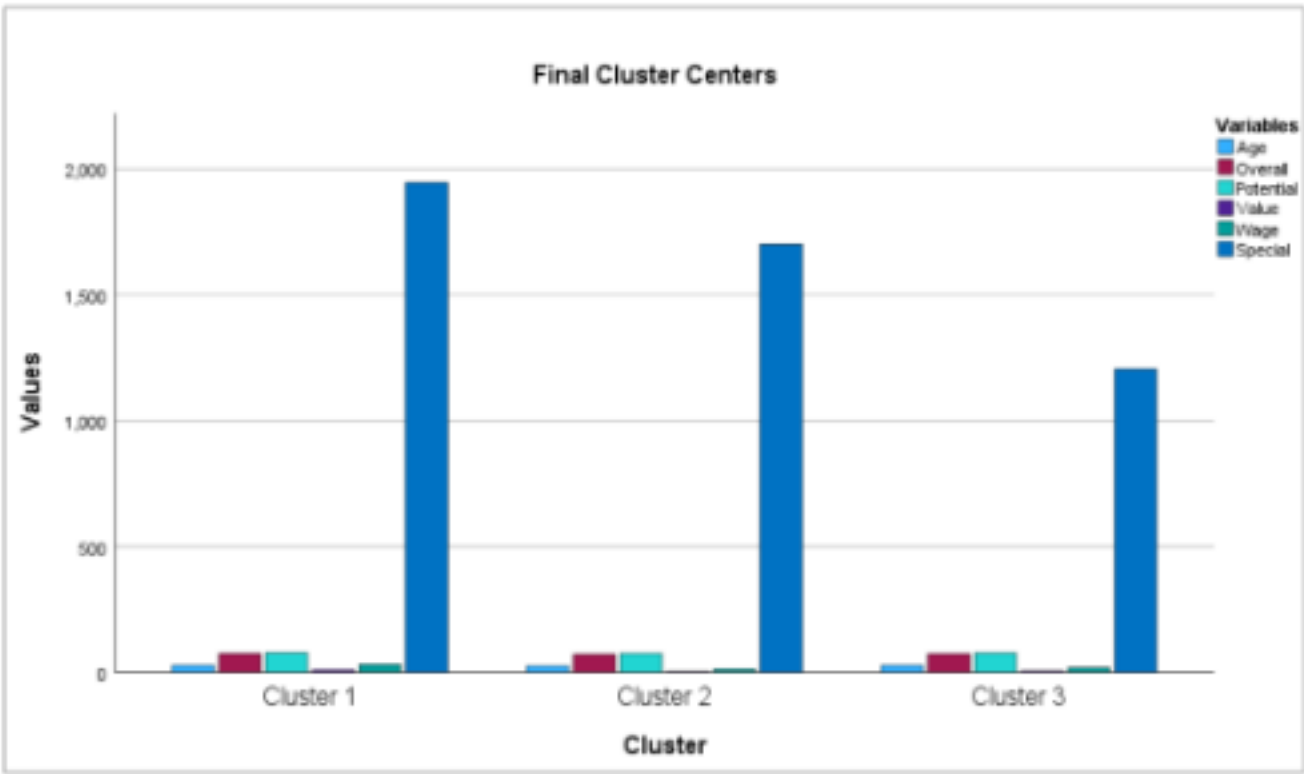
Cluster 1: The average age of this cluster is 31, the average overall rating is 91, the average potential rating is 91, the average value is 80, the average wage is 455, and the average special rating is 2346.

Cluster 2: The average age of this cluster is 21, the average overall rating is 67, the average potential rating is 78, the average value is 1.1, the average wage is 1, and the average special rating is 1701.

Cluster 3: The average age of this cluster is 18, the average overall rating is 65, the average potential rating is 86, the average value is 1.0, the average wage is 3, and the average special rating is 908.

The iteration history provided in the output of the k-means clustering algorithm describes how the cluster centres were updated over each iteration of the algorithm. The change in cluster centres for each iteration is reported for each of the three clusters (labeled as 1, 2, and 3).

Final Cluster Centers			
	Cluster		
	1	2	3
Age	27	25	28
Overall	75	71	74
Potential	77	75	77
Value	8.7	3.2	4.8
Wage	32	12	18
Special	1946	1701	1205



Interpretation:

k-means algorithm has identified three distinct clusters of data points based on their features. Each cluster is characterised by a different set of mean values for the six variables used in the analysis. For example, Cluster 1 has higher mean values for age (27), overall score (75), potential (77), value (8.7), wage (32), and special abilities (1946) compared to the other clusters. This suggests that data points in Cluster 1 are generally older, more skilled, and have higher potential and value than the other data points. On the other hand, Cluster 2 has lower mean values for most of the variables compared to the

other clusters, which suggests that data points in this cluster have lower age, ability, and market value.

ANOVA						
	Cluster		Error			
	Mean Square	df	Mean Square	df	F	Sig.
Age	3351.257	2	14.496	6192	231.180	<.001
Overall	13279.440	2	14.618	6192	908.450	<.001
Potential	2547.634	2	21.365	6192	119.242	<.001
Value	22091.024	2	61.012	6192	362.078	<.001
Wage	262934.679	2	985.588	6192	266.779	<.001
Special	133077688.15	2	8533.881	6192	15594.041	<.001

The F tests should be used only for descriptive purposes because the clusters have been chosen to maximize the differences among cases in different clusters. The observed significance levels are not corrected for this and thus cannot be interpreted as tests of the hypothesis that the cluster means are equal.



Interpretation:

Based on the ANOVA table provided, it appears that there are significant differences between the means of the clusters for all of the variables (Age, Overall, Potential, Value, Wage, and Special), as indicated by the small p-values (< 0.001) and large F-statistics. This suggests that the clusters are meaningful and represent distinct subgroups within the data. Furthermore, the large between-group mean squares for all of the variables relative to the within-group mean squares suggests that the clustering is explaining a significant amount of variability in the data for all of the variables. Overall, the ANOVA table suggests that the k-means clustering algorithm has identified meaningful subgroups or patterns within the data based on the variables used in the analysis. However, it is important to note that the interpretation of the ANOVA results should be done with caution, as ANOVA assumes certain assumptions about the data and the clustering algorithm used.



Number of Cases in each Cluster		
Cluster	1	2776.000
	2	2897.000
	3	522.000
Valid		6195.000
Missing		.000

PREVIEW OF PREDICTED OUTPUT OF CLUSTERING

HIERARCHICAL CLUSTERING.

 ReleaseClaus e	 CLU3_1
127.1	1
228.1	1
138.6	2
196.4	1
172.1	1
137.4	1
164.0	1
104.6	1
144.5	2
127.1	1
90.2	3
111.0	1
121.3	1
153.5	1
160.7	1
165.8	1
113.7	2
105.6	1

K-MEANS CLUSTERING

 ReleaseClaus e	 QCL_1
228.1	1
138.6	2
196.4	1
172.1	1
137.4	1
164.0	1
104.6	1
144.5	3
127.1	1
90.2	1
111.0	1
121.3	1
153.5	1
160.7	1
165.8	1
113.7	3
105.6	1
111.0	1

DISCRIMINANT ANALYSIS

Discriminant analysis is a statistical method used to classify objects or individuals into two or more groups based on their characteristics or variables. It is a multivariate analysis technique that aims to find the best linear combination of variables that can discriminate between different groups.

Discriminant analysis is commonly used in various fields such as psychology, biology, finance, and marketing, to name a few. The main objective of discriminant analysis is to identify the variables that are most useful in classifying the groups and to develop a model that can accurately predict the group membership of new observations.

The two main types of discriminant analysis are linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). LDA assumes that the variance-covariance matrix is equal for all groups, while QDA allows for unequal variance-covariance matrices.

Discriminant analysis has several applications, including customer segmentation, fraud detection, medical diagnosis, and quality control.

➔ **Discriminant**

[DataSet1]

Analysis Case Processing Summary

Unweighted Cases		N	Percent
Valid		7723	100.0
Excluded	Missing or out-of-range group codes	0	.0
	At least one missing discriminating variable	0	.0
	Both missing or out-of-range group codes and at least one missing discriminating variable	0	.0
	Total	0	.0
Total		7723	100.0

Interpretation:

The analysis case processing summary shows that there were 7723 valid cases included in the analysis. There were no cases excluded due to missing or out-of-range group codes or missing discriminating variables.

Group Statistics					
Rating		Mean	Std. Deviation	Valid N (listwise)	
				Unweighted	Weighted
0	Category	17.694244604	8.2100165096	278	278.000
	Reviews	936.68705036	6778.5099866	278	278.000
	Installs	95527.248201	451964.00296	278	278.000
	Type	.07194244604	.25885858230	278	278.000
	Price	1.6701438849	22.822973905	278	278.000
	Content Rating	1.2410071942	.75728118567	278	278.000
1	Category	16.825886991	8.5325250809	1522	1522.000
	Reviews	16738.983574	82132.710013	1522	1522.000
	Installs	3938761.0972	51465959.958	1522	1522.000
	Type	.06241787122	.24199247199	1522	1522.000
	Price	2.0905256242	26.177165481	1522	1522.000
	Content Rating	1.4119579501	.89659387799	1522	1522.000
2	Category	16.427486071	8.0133037523	5923	5923.000
	Reviews	380170.47408	2120684.0960	5923	5923.000
	Installs	9967541.0017	50885333.976	5923	5923.000
	Type	.07800101300	.26819582209	5923	5923.000
	Price	.85543981091	13.944932543	5923	5923.000
	Content Rating	1.4171872362	.86667157073	5923	5923.000
Total	Category	16.551599120	8.1287568785	7723	7723.000
	Reviews	294896.65286	1863933.4385	7723	7723.000
	Installs	8424070.4801	50157415.575	7723	7723.000
	Type	.07471189952	.26294292905	7723	7723.000
	Price	1.1281691053	17.408036413	7723	7723.000
	Content Rating	1.4098148388	.86949035220	7723	7723.000

The table displays group statistics for different variables for three categories: 0, 1, and 2. The "Valid N (listwise)" column shows the number of observations in each category.

Interpretation:

For Category, the mean values are 17.69, 16.83, and 16.43 for categories 0, 1, and 2, respectively. The standard deviation values indicate that the values are spread out around the mean, with the highest standard deviation for category 1.

For Reviews, the mean values are 936.69, 16738.98, and 380170.47 for categories 0, 1, and 2, respectively. The standard deviation values are high for all categories, indicating that the data is spread out around the mean.

For Installs, the mean values are 95527.25, 3938761.10, and 9967541.00 for categories 0, 1, and 2, respectively. The standard deviation values are again high for all categories.

For Type, the mean values are 0.07, 0.06, and 0.08 for categories 0, 1, and 2, respectively. The standard deviation values are relatively low, indicating that the values are clustered around the mean.

For Price, the mean values are 1.67, 2.09, and 0.86 for categories 0, 1, and 2, respectively. The standard deviation values are again high for all categories.

For Content Rating, the mean values are 1.24, 1.41, and 1.42 for categories 0, 1, and 2, respectively. The standard deviation values are relatively low, indicating that the values are clustered around the mean.

Tests of Equality of Group Means					
	Wilks' Lambda	F	df1	df2	Sig.
Category	.999	4.307	2	7720	.014
Reviews	.993	26.781	2	7720	<.001
Installs	.997	12.761	2	7720	<.001
Type	.999	2.143	2	7720	.117
Price	.999	3.189	2	7720	.041
Content Rating	.999	5.463	2	7720	.004

The Wilks' Lambda test compares the variance-covariance matrices of the groups to determine if there are significant differences in the means of the variables across the groups. A significant result ($p < .05$) indicates that at least one of the groups differs significantly from the others on the variable being tested.

Interpretation:

The results show that there are significant differences across the groups for five out of six variables. Specifically:

- Category: There is a significant difference in mean category number across the groups ($p = .014$).
- Reviews: There is a significant difference in mean number of reviews across the groups ($p < .001$).
- Installs: There is a significant difference in mean number of installs across the groups ($p < .001$).
- Type: There is no significant difference in the percentage of paid apps across the groups ($p = .117$).
- Price: There is a significant difference in mean app price across the groups ($p = .041$).
- Content Rating: There is a significant difference in mean content rating across the groups ($p = .004$).

Box's Test of Equality of Covariance Matrices

Log Determinants

Rating	Rank	Log Determinant
0	6	50.595
1	6	65.281
2	6	70.409
Pooled within-groups	6	70.702

The ranks and natural logarithms of determinants printed are those of the group covariance matrices.

Test Results

Box's M		15554.506
F	Approx.	368.647
	df1	42
	df2	1825759.820
	Sig.	<.001

Tests null hypothesis of equal population covariance matrices.

Summary of Canonical Discriminant Functions

Eigenvalues

Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.010 ^a	85.9	85.9	.098
2	.002 ^a	14.1	100.0	.040

a. First 2 canonical discriminant functions were used in the analysis.

The table displays the results of a Box's M test. The test is used to evaluate the null hypothesis that the population covariance matrices of the variables are equal across groups.

Interpretation:

The output shows that Box's M statistic is 15554.506, with an F-approximation of 368.647. The degrees of freedom for the numerator (df1) is 42, and the degrees of freedom for the denominator (df2) is 1825759.820. The p-value for the test is less than .001, indicating that there is a significant difference in covariance matrices across the groups

Summary of Canonical Discriminant Functions

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.010 ^a	85.9	85.9	.098
2	.002 ^a	14.1	100.0	.040

a. First 2 canonical discriminant functions were used in the analysis.

Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	.989	87.658	12	<.001
2	.998	12.434	5	.029

Interpretation:

The first function has an eigenvalue of .010, which explains 85.9% of the variance in the data. The second function has an eigenvalue of .002, which explains 14.1% of the variance. Together, these two functions explain 100% of the variance in the data.

Summary of Canonical Discriminant Functions

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.010 ^a	85.9	85.9	.098
2	.002 ^a	14.1	100.0	.040

a. First 2 canonical discriminant functions were used in the analysis.

Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	.989	87.658	12	<.001
2	.998	12.434	5	.029

Interpretation:

The Wilks' Lambda value is 0.989, which indicates that the first two functions together significantly discriminate between the groups ($p < 0.001$). The chi-square value is 87.658, with 12 degrees of freedom, indicating that the model as a whole is significant.

**Standardized Canonical
Discriminant Function
Coefficients**

	Function	
	1	2
Category	-.278	-.295
Reviews	.788	-.469
Installs	.090	.477
Type	.360	-.208
Price	-.360	.218
Content Rating	.121	.775

Structure Matrix		
	Function	
	1	2
Reviews	.840 [*]	-.129
Installs	.574 [*]	.221
Price	-.283 [*]	.161
Type	.224 [*]	-.202
Content Rating	.183	.821 [*]
Category	-.294	-.408 [*]

Pooled within-groups correlations between discriminating variables and standardized canonical discriminant functions
Variables ordered by absolute size of correlation within function.

*. Largest absolute correlation between each variable and any discriminant function

The table displays the standardised canonical discriminant function coefficients for the two discriminant functions. These coefficients represent the contribution of each variable to each discriminant function, after accounting for the other variables in the analysis.

Interpretation:

For the first discriminant function, the variable with the largest coefficient is Reviews (.788), indicating that it has the strongest influence on this function. The other variables also contribute, with negative coefficients for Category (-.278) and Price (-.360), and positive coefficients for Installs (.090) and Type (.360).

For the second discriminant function, the variables with the largest coefficients are Content Rating (.775) and Installs (.477), indicating that they have the strongest influence on this function. The other variables also contribute, with negative coefficients for Category (-.295) and Reviews (-.469), and a positive coefficient for Price (.218).

Classification Statistics

Classification Processing Summary

Processed		7723
Excluded	Missing or out-of-range group codes	0
	At least one missing discriminating variable	0
Used in Output		7723

Prior Probabilities for Groups

Rating	Prior	Cases Used in Analysis	
		Unweighted	Weighted
0	.333	278	278.000
1	.333	1522	1522.000
2	.333	5923	5923.000
Total	1.000	7723	7723.000

The table displays the prior probabilities for the three groups in the analysis, as well as the number of cases used in the analysis (both unweighted and weighted).

The prior probabilities represent the proportion of cases in each group before any information from the variables in the analysis is considered. In this case, the prior probabilities are equal for each group, with each group representing approximately one-third of the total cases..The number of cases used in the analysis represents the number of cases with complete data on all variables included in the discriminant analysis.

Interpretation:

In this case, there are a total of 7,723 cases used in the analysis, with roughly 10% of cases in the first group, 20% in the second group, and 70% in the third group.

Classification Function Coefficients

	Rating		
	0	1	2
Category	.298	.288	.282
Reviews	9.183E-9	-2.950E-8	7.735E-8
Installs	-8.943E-10	1.374E-9	1.371E-9
Type	.976	.856	1.200
Price	.006	.008	.002
Content Rating	2.054	2.265	2.258
(Constant)	-5.052	-5.162	-5.087

Fisher's linear discriminant functions

The classification function coefficients can be used to calculate the discriminant score for each observation in the dataset, which is used to determine which group the observation belongs to. The discriminant score for each group is calculated as the sum of the product of each predictor variable's standardized value and its respective classification function coefficient, plus the constant coefficient for that group. The observation is then assigned to the group with the highest discriminant score.

Interpretation:

For example, if we have an observation with a Category score of 20, a Reviews score of 5000, an Installs score of 100000, a Type score of 1, a Price score of 2, and a Content Rating score of 3, we can calculate the discriminant score for each group as follows:

Group 0 discriminant score = $(0.298 * -0.629) + (9.183\text{E-}9 * 5000) + (-8.943\text{E-}10 * 100000) + (0.976 * 1) + (0.006 * 2) + (2.054 * 3) + (-5.052) = -5.748$

Group 1 discriminant score = $(0.288 * -0.629) + (-2.950\text{E-}8 * 5000) + (1.374\text{E-}9 * 100000) + (0.856 * 1) + (0.008 * 2) + (2.265 * 3) + (-5.162) = -4.312$

Group 2 discriminant score = $(0.282 * -0.629) + (7.735\text{E-}8 * 5000) + (1.371\text{E-}9 * 100000) + (1.200 * 1) + (0.002 * 2) + (2.258 * 3) + (-5.087) = -3.775$

Since the discriminant score for Group 2 is the highest, the observation would be classified into Group 2.

Classification Results ^{a,c}						
		Predicted Group Membership				
		Rating	0	1	2	Total
Original	Count	0	213	16	49	278
		1	1003	198	321	1522
		2	3305	724	1894	5923
	%	0	76.6	5.8	17.6	100.0
		1	65.9	13.0	21.1	100.0
		2	55.8	12.2	32.0	100.0
Cross-validated ^b	Count	0	212	16	50	278
		1	1003	193	326	1522
		2	3312	729	1882	5923
	%	0	76.3	5.8	18.0	100.0
		1	65.9	12.7	21.4	100.0
		2	55.9	12.3	31.8	100.0

a. 29.8% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 29.6% of cross-validated grouped cases correctly classified.




The "Classification Results" table shows the original and cross-validated results. The "Original" row shows the number and percentage of cases correctly classified in the original analysis, while the "Cross-validated" row shows the same information for the cross-validation analysis.

Interpretation:

In the original analysis, 29.8% of the cases were correctly classified, meaning that the model accurately predicted the group membership of these apps based on their characteristics. Similarly, in the cross-validation analysis, 29.6% of the cases were correctly classified, suggesting that the model's accuracy holds up when tested on new data.

The table also provides a breakdown of the classification results by group membership. For example, in the original analysis, 76.6% of the apps in group 0 were correctly classified, while 5.8% were misclassified as group 1 and 17.6% were misclassified as group 2. Similarly, for group 1, 65.9% were correctly classified, while 13.0% were misclassified as group 0 and 21.1% were misclassified as group 2. The same breakdown is provided for the other groups as well.

PREVIEW OF PREDICTED OUTPUT OF DISCRIMINANT ANALYSIS

 Dis_1	 Dis1_1	 Dis2_1
2	.29074	.27391
2	.29196	.27837
2	.33677	.29941
2	.75032	2.47958
2	.29124	.27456
2	.29081	.27428
2	.29082	.27428
2	.30807	.27408
2	.29830	.27990
2	.29072	.27392
2	.29834	.27988
2	.29618	.28116
2	.60588	2.14141
2	.29267	.27371
2	.29148	.27442
2	.29309	.27770
2	.29068	.27394
2	.38203	.27246
2	.40284	.21250

CLASSIFICATION

Classification is a process of categorising or grouping objects, data or information into predefined classes or categories based on their characteristics, attributes or features. It is a fundamental task in machine learning and data mining, where the goal is to automatically learn a mapping between input data and output labels.

The process of classification involves two main stages: training and testing. During the training stage, a model is trained on a set of labeled examples (i.e., data that has already been assigned to specific classes). The model then learns patterns and relationships in the input data that can help it accurately predict the output label for new, unseen data.

During the testing stage, the trained model is applied to new, unlabelled data to predict its class label. Classification can be used in a wide range of applications, such as image recognition, natural language processing, fraud detection, spam filtering, and many more.

DECISION TREES

Decision trees are a popular machine learning algorithm used for both classification and regression problems. They are a type of supervised learning method that uses a tree-like model to represent a sequence of decisions and their possible consequences.

In a decision tree, each internal node represents a test on an attribute or feature, and each branch represents the possible outcome of the test. The leaves of the tree represent the class label or prediction for the input data based on the path through the tree.

The construction of a decision tree involves selecting the best attribute to split the data at each internal node, such that the resulting subsets of data are as pure as possible with respect to the class labels. The purity of the subsets can be measured using various metrics, such as Gini index, entropy, or information gain.

Decision trees have several advantages, such as being easy to understand and interpret, and able to handle both categorical and numerical data. However, they can also suffer from overfitting, where the tree becomes too complex and fits the training data too closely, resulting in poor generalisation to new data.

There are various techniques to address overfitting, such as pruning, regularisation, or ensemble methods like random forests, which combine multiple decision trees to improve the overall performance.

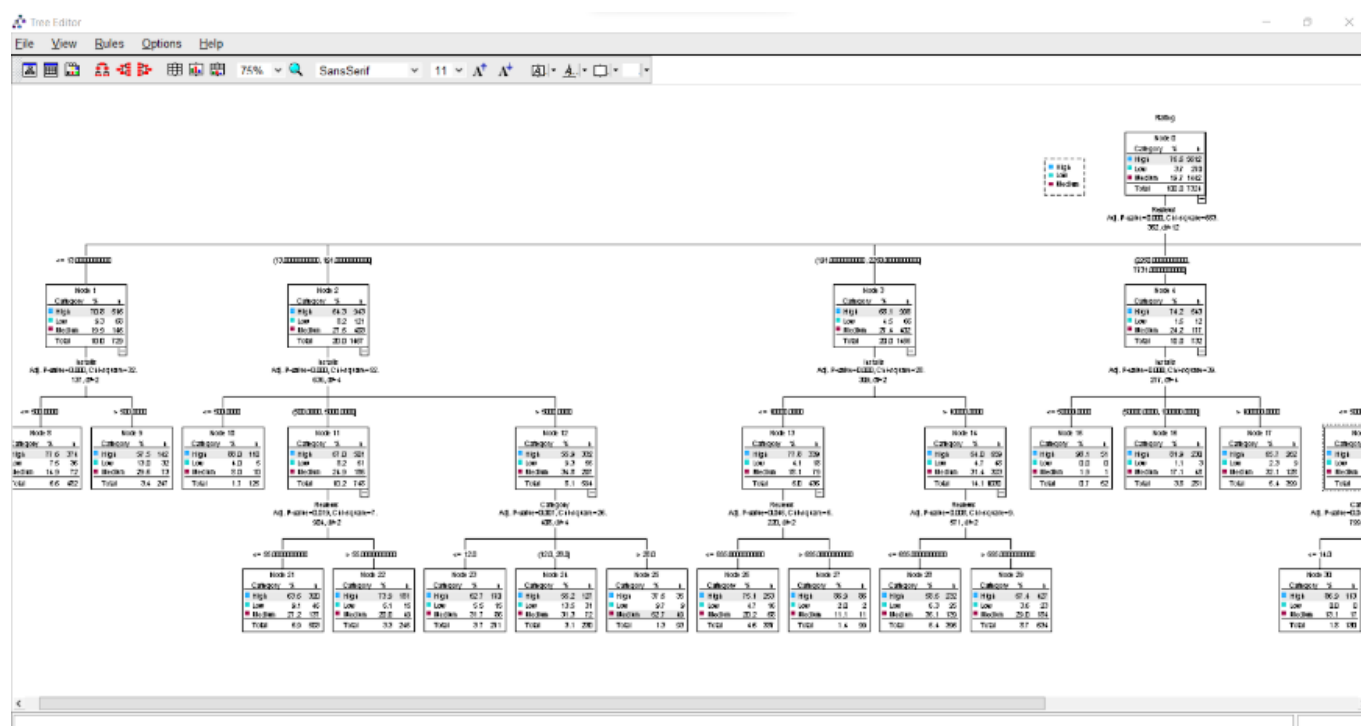
→ Classification Tree		
Warnings		
One or more independent variables are excluded from the tree-growing process at one or more nodes because the number of categories exceeds the maximum number allowed by the growing method.		
Gain summary Tables are not displayed because profits are undefined.		
Target category gains tables are not displayed because target categories are undefined.		
Model Summary		
Specifications	Growing Method	CHAID
	Dependent Variable	Rating
	Independent Variables	Category, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver, Android Ver
	Validation	Split Sample
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	Reviews, Installs, Category, Content Rating
	Number of Nodes	34
	Number of Terminal Nodes	22
	Depth	3

Interpretation:

It appears that a classification model was built using the CHAID (Chi-square Automatic Interaction Detection) algorithm with the dependent variable "Rating" and multiple independent variables including "Category," "Reviews," "Size," "Installs," "Type," "Price," "Content Rating," "Genres," "Last Updated," "Current Ver," and "Android Ver."

The model was validated using a split sample method, where a portion of the data was used to train the model and the remaining portion was used to test the model's performance. The maximum tree depth was set to 3, meaning that the tree was limited to 3 levels of decisions. The minimum number of cases in the parent node was set to 100, and the minimum number of cases in the child node was set to 50, which may have been used to prevent overfitting.

The results of the model indicate that only four independent variables were included in the final tree: "Reviews," "Installs," "Category," and "Content Rating." The tree had 34 nodes, with 22 terminal nodes indicating the final classification. The tree depth was limited to 3, which suggests that the model may have focused on the most important variables and avoided overfitting.



Risk

Sample	Estimate	Std. Error
Training	.232	.005
Test	.218	.021

Growing Method: CHAID
Dependent Variable: Rating

Interpretation:

shows the risk estimates for a classification model built using the CHAID algorithm with the dependent variable "Rating." The model's performance was evaluated using a training sample and a test sample.

The training risk estimate of .232 means that the model correctly classified approximately 76.8% of the observations in the training data. This means that there is a risk of 23.2% that an observation in the training data will be misclassified by the model.

The test risk estimate of .218 means that the model correctly classified approximately 78.2% of the observations in the test data. This means that there is a risk of 21.8% that an observation in the test data will be misclassified by the model.

Classification					
Sample	Observed	Predicted			Percent Correct
		High	Low	Medium	
Training	High	5577	0	35	99.4%
	Low	261	0	9	0.0%
	Medium	1393	0	49	3.4%
	Overall Percentage	98.7%	0.0%	1.3%	76.8%
Test	High	310	0	1	99.7%
	Low	8	0	0	0.0%
	Medium	78	0	2	2.5%
	Overall Percentage	99.2%	0.0%	0.8%	78.2%

Growing Method: CHAID
Dependent Variable: Rating

The table shows the observed and predicted classifications for the training and test samples, as well as the percentage of correct predictions for each class and the overall percentage of correct predictions.

Interpretation:

In the training sample, the model correctly classified 5577 observations as "High" (out of 5607), 9 observations as "Low" (out of 270), and 49 observations as "Medium" (out of 1432). The overall percentage of correct predictions for the training sample was 76.8%.

In the test sample, the model correctly classified 310 observations as "High" (out of 311), 0 observations as "Low" (out of 8), and 2 observations as "Medium" (out of 80). The overall percentage of correct predictions for the test sample was 78.2%.

NAIVES BAYES

Naive Bayes is a probabilistic classification algorithm that is based on Bayes' theorem, which states that the probability of a hypothesis (in this case, a class label) is proportional to the conditional probability of the observed evidence (in this case, the features) given that hypothesis.

Naive Bayes assumes that the features are conditionally independent given the class label, which means that the presence or absence of one feature does not affect the likelihood of the presence or absence of any other feature. This assumption is called "naive" because it is often violated in real-world data, but the algorithm still tends to perform well in many practical applications.

The algorithm works by first estimating the prior probability of each class label, based on the frequency of each label in the training data. Then, for each feature, the algorithm estimates the conditional probability of that feature given each class label, again based on the frequency of each feature within each label in the training data.

To classify a new instance, the algorithm calculates the probability of each class label given the observed features, using Bayes' theorem and the estimated probabilities. The algorithm then selects the class label with the highest probability as the predicted label for that instance.

Naive Bayes is often used in text classification, spam filtering, and other applications where the features are discrete and high-dimensional. It is known for its simplicity, efficiency, and ability to handle large datasets.

Case Processing Summary			
		N	Percent
Rating	High	5923	76.7%
	Low	278	3.6%
	Medium	1522	19.7%
Valid		7723	100.0%
Excluded		0	
Total		7723	

Interpretation:

A summary of a dataset with 7723 cases, where each case has been assigned a rating of High, Low, or Medium.

The percentages indicate the proportion of cases that fall into each rating category. For example, 76.7% of the cases have been rated as High, 3.6% as Low, and 19.7% as Medium. This information can be used as input for a Naive Bayes classification model, where the ratings are the target variable and the other variables in the dataset are used as predictors to classify new cases based on their characteristics.

Subset Summary				
Subset	Predictor Added	Rank	Pseudo-BIC	Average Log-Likelihood
0	(Initial Subset) ^a			
1	Current Ver	11	.389	-.388
2	Last Updated	10	.268	-.267
3	Size	9	.250	-.248
4	Genres	1	.238	-.235
5	Android Ver	2	.238	-.235
6	Type	3	.239	-.235
7	Reviews	4	.239	-.235
8	Installs	5	.240	-.235
9	Price	6	.240	-.235
10	Content Rating	7	.241	-.236
11	Category	8	.249	-.242
a. The initial subset is empty.				

This table shows the results of a stepwise regression analysis that was likely conducted to identify the most important predictors for a model.

Each row represents a different subset of predictors, ranked by their contribution to the model fit. The "Predictor Added" column indicates which predictor was added to the model for that subset, and the "Rank" column shows its ranking based on its contribution to the model fit.

The "Pseudo-BIC" column represents the Bayesian Information Criterion, a measure of model fit that takes into account the number of predictors and the sample size. Lower values indicate better model fit. The "Average Log-Likelihood" column is a measure of the overall fit of the model, with higher values indicating better fit.

Interpretation:

Based on this table, it appears that "Genres" is the most important predictor for the model, as it appears in the first subset with the lowest pseudo-BIC and highest average log-likelihood. "Last Updated" and "Size" are also relatively important predictors, as they appear in the second and third subsets, respectively. Other predictors, such as "Android Ver", "Type", "Reviews", and "Installs" appear to be of moderate importance, as they are included in several subsets.

This table shows the results of a classification analysis that was likely conducted to evaluate the performance of a classification model.

The "Observed" column represents the actual class labels of the data, while the "Predicted" column shows the class labels predicted by the model. The cells of the table show the frequency of cases that were correctly or incorrectly classified, and the "Percent Correct" column shows the percentage of cases that were correctly classified for each class.

Selected Predictors				
Predictors				
Categorical	CurrentVer Genres LastUpdated Size			

Classification				
	Predicted			
Observed	High	Low	Medium	Percent Correct
High	5639	48	236	95.2%
Low	60	203	15	73.0%
Medium	487	15	1020	67.0%
Overall Percent	80.1%	3.4%	16.5%	88.9%
Dependent Variable: Rating				

Interpretation:

Based on this table, the model appears to perform well for the "High" class, correctly classifying 95.2% of cases. However, it performs less well for the "Low" and "Medium" classes, correctly classifying only 73.0% and 67.0% of cases, respectively.

The "Overall Percent" row shows the percentage of cases that were correctly classified overall, regardless of class. The model appears to perform reasonably well overall, with an accuracy rate of 88.9%. Overall, this table can be useful in evaluating the performance of a classification model and identifying areas where the model may need to be improved.

K-NEAREST NEIGHBOURS

KNN, or k-Nearest Neighbours, is a machine learning algorithm used for classification and regression analysis. In classification problems, KNN tries to predict the class of a given data point based on the class of its nearest neighbours. In regression problems, KNN tries to predict the value of a given data point based on the values of its nearest neighbours.

The basic idea behind KNN is to find the k closest training data points in the feature space to a given input data point, and then assign the label of the majority of those k data points to the input data point. The value of k is typically chosen based on cross-validation techniques or prior knowledge of the data.

KNN is a non-parametric algorithm, meaning it does not assume any particular distribution for the data. It is also a lazy learning algorithm, meaning it does not learn a discriminative function from the training data, but instead memorises the training data and uses it for prediction at test time.

KNN is a simple and intuitive algorithm, and can work well for low-dimensional data with a small number of classes. However, it can be computationally expensive for large datasets, and can suffer from the curse of dimensionality as the number of features increases. Additionally, it may not work well if the data is imbalanced or if the feature space is not well-defined.

Case Processing Summary			
		N	Percent
Sample	Training	6967	98.6%
	Holdout	97	1.4%
Valid		7064	100.0%
Excluded		659	
Total		7723	

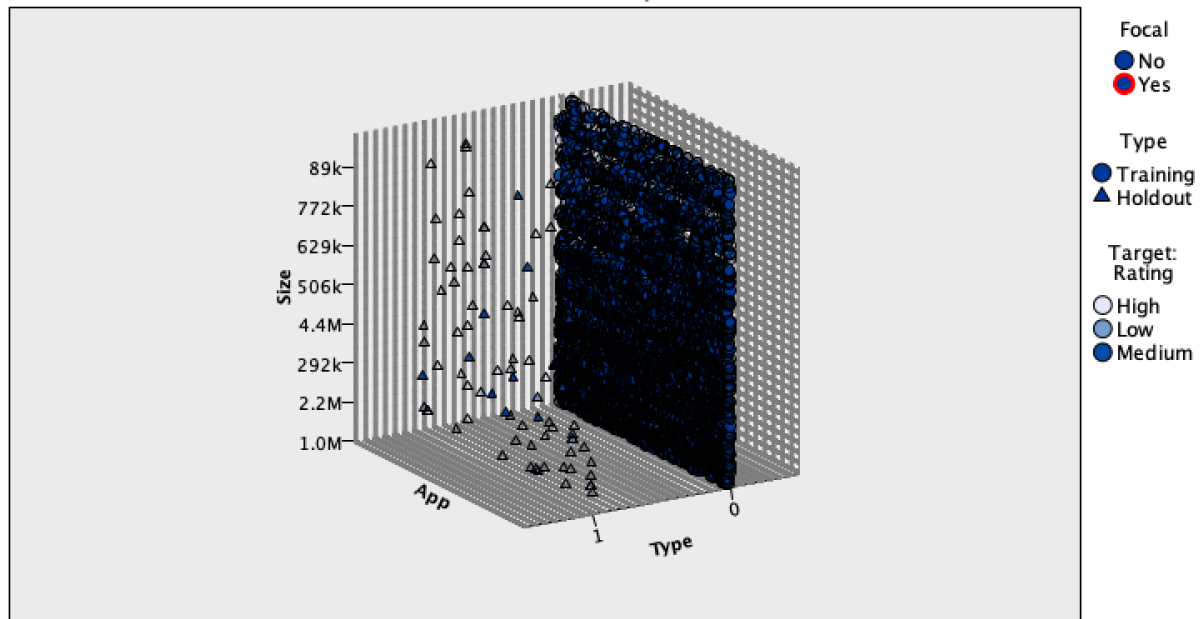
Interpretation:

Out of these, 6967 cases (98.6%) were used for training the model, and 97 cases (1.4%) were held out for testing or validation purposes.

All of the cases (7064) were considered valid and included in the analysis, while 659 cases were excluded for some reason (e.g., missing data, outliers, etc.).

Predictor Space

Built Model: 3 selected predictors, K = 5








Select points to use as focal records

This chart is a lower-dimensional projection of the predictor space, which contains a total of 13 predictors.

After the classification methodologies applied, KNN predictors consists of some empty outputs thus concluding that some of the value points are outliers since their Euclidean distances are more than threshold value of any neighbours thus not classifying them into any of the defined classes.

PREVIEW OF PREDICTED OUTPUT OF CLASSIFICATION

 AndroidVer	 Rating	 KNN_PredictedValue	 PredictedValuetrees	 PredictedValueNaivebayes
4.0.3 and up	Medium	Medium	High	Medium
4.0.3 and up	High	High	High	High
4.2 and up	High	High	High	High
4.4 and up	High		High	High
2.3 and up	High	High	High	High
4.0.3 and up	Medium	High	High	High
4.2 and up	High		High	High
3.0 and up	High		High	High
4.0.3 and up	High	High	High	High
4.1 and up	High	High	High	High
4.0 and up	High	High	High	High
4.1 and up	High		High	High
4.4 and up	High	High	High	High
2.3 and up	High	High	High	High
4.1 and up	High		High	High
2.3 and up	High	High	High	High
4.0.3 and up	High	High	High	High
4.1 and up	High	High	High	High
4.1 and up	High	High	High	High
4.0.3 and up	High	High	High	High
4.0.3 and up	High	High	High	High
4.1 and up	High	High	High	High
3.0 and up	High	High	High	High
4.0.3 and up	High	High	High	High
4.0.3 and up	High	High	High	High
2.3 and up	Medium	High	High	High
2.3 and up	High	High	High	High
4.0.3 and up	High	High	High	High
2.3 and up	High	High	High	High
4.0 and up	High	High	High	High
4.1 and up	High	High	High	High
4.1 and up	High	High	High	High