Turtle Games: Predicting Future Outcomes (Python and R)

Sujith Kumaar K C

16th Dec 2024

1. Problem Statement:

Turtle Games, a global game manufacturer aims to **improve overall sales performance** and seeks insights into their customer trends for marketing enhancements.

Stakeholders: Turtle Games – Sales and Marketing team, Product Management team, Technology partner

2. Business Objective Analysis:

Through a 5-why structured analysis approach, the core objective of Turtle could be to improve sales by comprehensively understanding customer behaviour and improving customer experience by:

- Improving Customer retention
- Enhancing marketing effectiveness
- Evaluating Customer Feedback
- Refine product offerings

Considering the data limitations, we will address these objectives by targeting to improve the Loyalty scheme to target these objectives.

3. Data Assumptions:

Data assumptions defined based on metadata and data exploration to avoid ambiguity:

- The dataset provided is a true sample subset representative of the entire customer base of Turtle Games and accounts for different demographics and all possible purchase platforms.
- Each row is assumed to represent a unique customer. Since Customer ID information is not available, it cannot be confirmed whether multiple rows correspond to the same customer making multiple purchases.
- Product codes provided can be mapped to actual products by Turtle Games internally.
- All reviews are from actual product purchases.

4. Analysis

4.1 Data Understanding and Preparation

The analysis uses the "turtle_reviews.csv" dataset, containing customer demographics, spending habits, loyalty points, and reviews. No duplicates or missing values were found, but future dataset should include Customer ID to avoid duplicates. Skewness in quantitative variables was observed and log transformation was identified as potential solution to be applied as required. Loyalty point outliers were retained to preserve dataset size. Age data showed potential misreporting in 2.5% of cases. Categorical variables like gender and education showed minimal impact on quantitative variables and were not further engineered. The cleaned dataset was exported for future use.

4.2 Linear Regression

To establish a mathematical relationship between customer attributes and loyalty points. Simple and multiple linear regressions were applied to assess the predictive power of spending score, remuneration, and age.

Method

- **[Python]:** `OLS() function [statsmodel]`
- **[R]:** `lm()function, Shapiro-Wilk test for normality and Breusch-Pagan test for heteroscedasticity`

Inputs:

- Predicted: Loyalty Points
- Predictors: spending score, remuneration, age

Process and Optimization:

1. Simple Linear Regression was applied between the predicted and predictor variables based on correlation results.
2. Multiple Linear Regression was applied using lm() in R.
3. Shapiro-Wilk test for normality and Breusch-Pagan test for heteroscedasticity were applied to evaluate and refine model performance.
4. Transformations (log and square root) were applied to loyalty points and predictor variables to improve model assumptions.
5. Log transformation of all variables led to a high adjusted R-squared (99.23%), indicating potential overfitting.
6. Square-root transformation of loyalty points improved adjusted R-squared to 90.68%, but normality failed.
7. Weighted Least Squares (WLS) regression was used to address heteroscedasticity and retained normality as well and was decided as the optimized model.

Model #	Dep Variable	Independent Variables / Factors	Model Accuracy (Adj R^2)	(p-Value)	Normality Test	Heteroscedasticity Test
1	LP	SS,REM	82.7%	<2.2e-16	Pass	Fail
2	LP	SS,REM,AGE	84.0%	<2.2e-16	Pass	Fail
3	Log (LP)	Log (SS, REM, AGE)	99.2% (Overfit)	<2.2e-16	Fail	Pass
4	Sqrt (LP)	SS,REM,AGE	90.7%	<2.2e-16	Fail	Pass
5	**LP**	**SS,REM,AGE + Weights from model 2**	**82.0%**	**<2.2e-16**	**Pass**	**Pass**

Refer Appendix 3 for implementation and stat test results of the WLS Regression model

4.3 Decision Tree Regression

Decision Tree was implemented to explore the structure of loyalty point drivers and identify decision rules for segmentation.

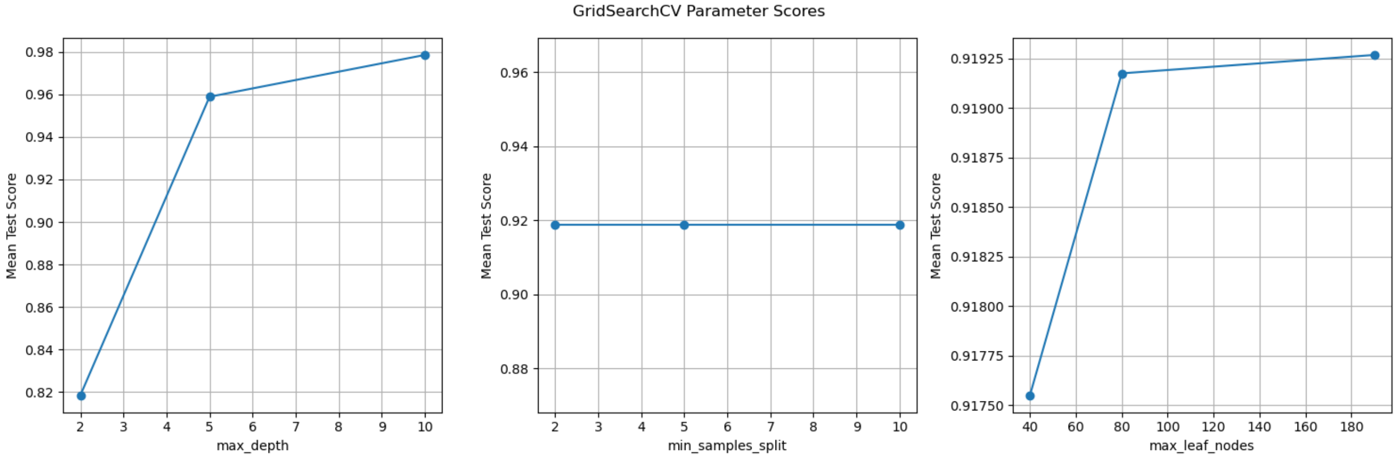
Method [Python]: `DecisionTreeRegressor [sklearn.tree]`

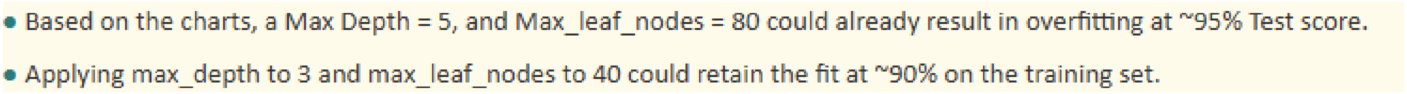
Inputs: One-hot encoded categorical features (gender, education) and all quantitative features - age, remuneration, spending score.

Process and Optimization:

1. The Initial tree was grown with default parameters and all features.
2. This was pruned from 10 features to 2 features in the second model
3. The final model was derived by pruning the max_depth and max_leaf_nodes parameters. The optimal depth and leaf node values were identified through **GridSearchCV Hyperparameter tuning** to retain **90% fit.**

Model	Test Set Metrics	Tree Structure Information
regressor [Full Initial Model]	**R-squared Score on Test Set: 99.53%** Mean Absolute Error: 31.95 Mean Squared Error: 7599.02 Root Mean Squared Error: 87.17234844452301	Max Depth: 23 Number of Leaves: 556 Feature Count: 10
regressor2 [Features pruned Model]	**R-squared Score on Test Set: 98.39%** Mean Absolute Error: 83.27 Mean Squared Error: 26097.98 Root Mean Squared Error: 161.54870369129105	Max Depth: 18 Number of Leaves: 196 Feature Count: **2**
regressor_opt **[GridSearchCV Optimized Model]**	**R-squared Score on Test Set: 98.71%** Mean Absolute Error: 75.04 Mean Squared Error: 20880.52 Root Mean Squared Error: 144.5009345710073	Max Depth: **3** Number of Leaves: **40** Feature Count: **2**

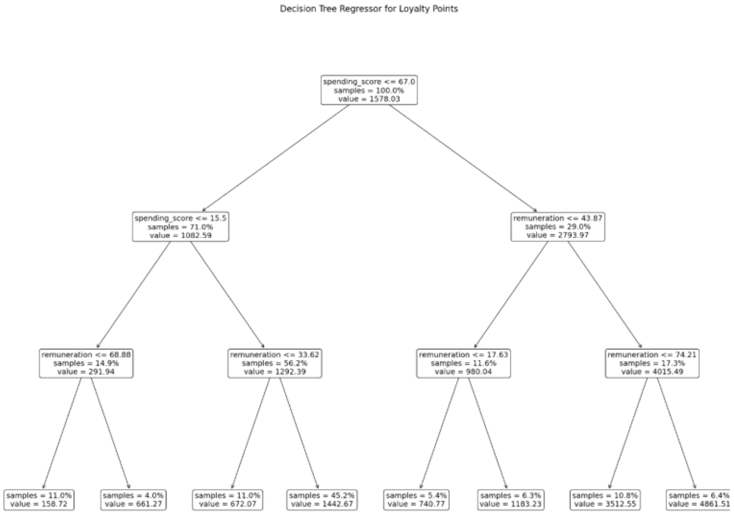


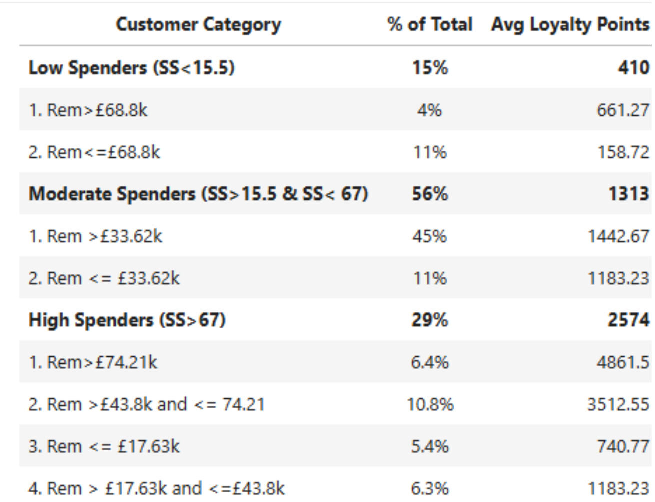


- Based on the charts, a Max Depth = 5, and Max_leaf_nodes = 80 could already result in overfitting at ~95% Test score.
- Applying max_depth to 3 and max_leaf_nodes to 40 could retain the fit at ~90% on the training set.

Fig: GridSearch CV Hyperparameter Tuning results plotted for different Decision Tree Parameters

4. Optimized Decision tree model achieved a high R-squared score on the full data set (91.04%).
5. Decision Tree results were interpreted as below, leading to identifying the possibility of applying k-means clustering to effectively classify customer segments.





Customer Category	% of Total	Avg Loyalty Points
Low Spenders (SS<15.5)	**15%**	**410**
1. Rem>£68.8k	4%	661.27
2. Rem<=£68.8k	11%	158.72
Moderate Spenders (SS>15.5 & SS< 67)	**56%**	**1313**
1. Rem >£33.62k	45%	1442.67
2. Rem <= £33.62k	11%	1183.23
High Spenders (SS>67)	**29%**	**2574**
1. Rem>£74.21k	6.4%	4861.5
2. Rem >£43.8k and <= 74.21	10.8%	3512.55
3. Rem <= £17.63k	5.4%	740.77
4. Rem > £17.63k and <=£43.8k	6.3%	1183.23

Fig: Optimized 3-Level Decision Tree and Results Interpretation Table

4.4 Clustering (k-Means):

Method [Python] : `KMeans [sklearn.cluster]`

Inputs: Remuneration, Spending score.

Process and Optimization:

1. Elbow method and Silhouette Score were used to identify the optimal range of clusters as 4,5,6.

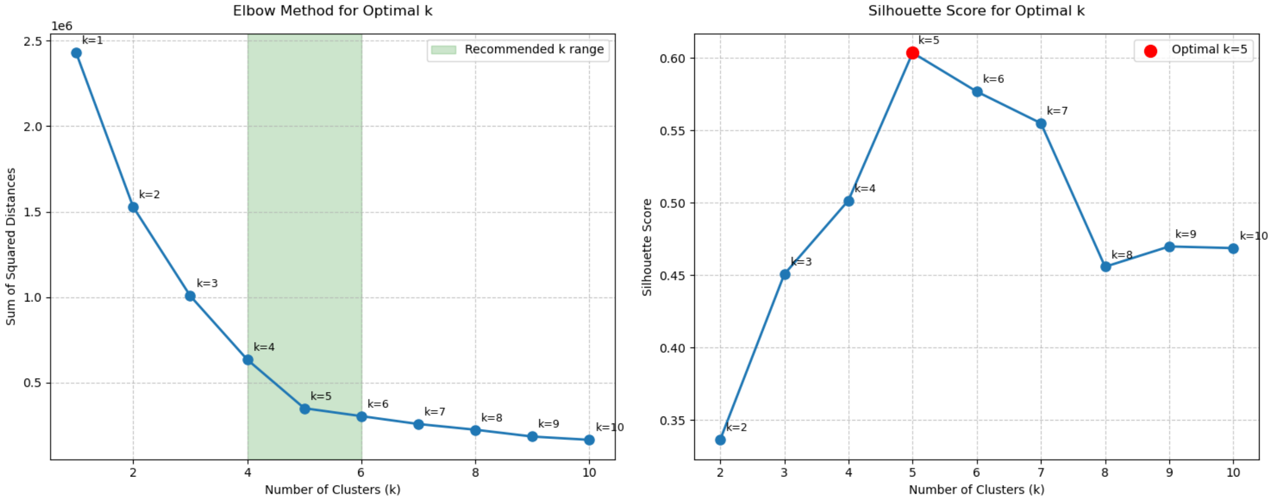
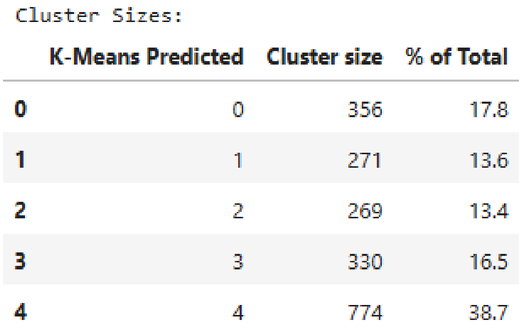


Fig: Elbow method and Silhouette Method visualized for Optimal cluster size.

2. k-Means was applied for identified cluster sizes (k). Kde Pairplots were generated to visualize the cluster results. 4 and 6 clusters did not have fairly equal distribution and had overlapping clusters in the visualization. Hence 5 clusters were finalized – Silhouette score was 0.604 suggesting good cluster separation.

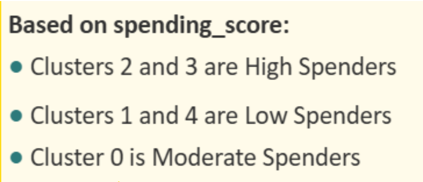


```
Cluster Sizes:
```

	K-Means Predicted	Cluster size	% of Total
0	0	356	17.8
1	1	271	13.6
2	2	269	13.4
3	3	330	16.5
4	4	774	38.7

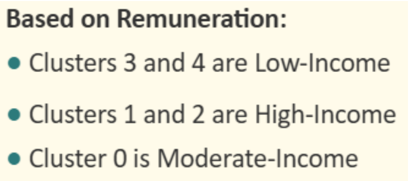
Fig: Cluster size check for equal distribution.

3. Persona identification was done using groupby on the dataset based on k-Means Predicted Clusters and aggregating **(min, max, mean, median)** the remuneration and spending score.



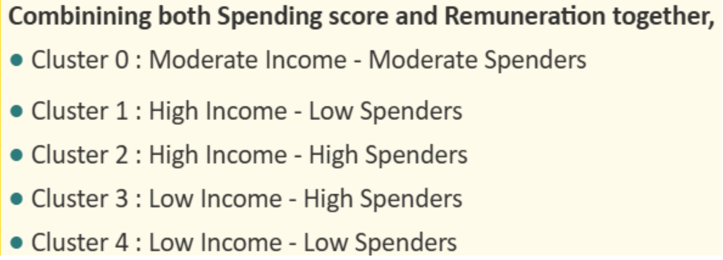
Based on spending_score:

- Clusters 2 and 3 are High Spenders
- Clusters 1 and 4 are Low Spenders
- Cluster 0 is Moderate Spenders



Based on Remuneration:

- Clusters 3 and 4 are Low-Income
- Clusters 1 and 2 are High-Income
- Cluster 0 is Moderate-Income



Combinining both Spending score and Remuneration together,

- Cluster 0 : Moderate Income - Moderate Spenders
- Cluster 1 : High Income - Low Spenders
- Cluster 2 : High Income - High Spenders
- Cluster 3 : Low Income - High Spenders
- Cluster 4 : Low Income - Low Spenders

4. Loyalty points distribution was analysed based on Customer Segment / Persona

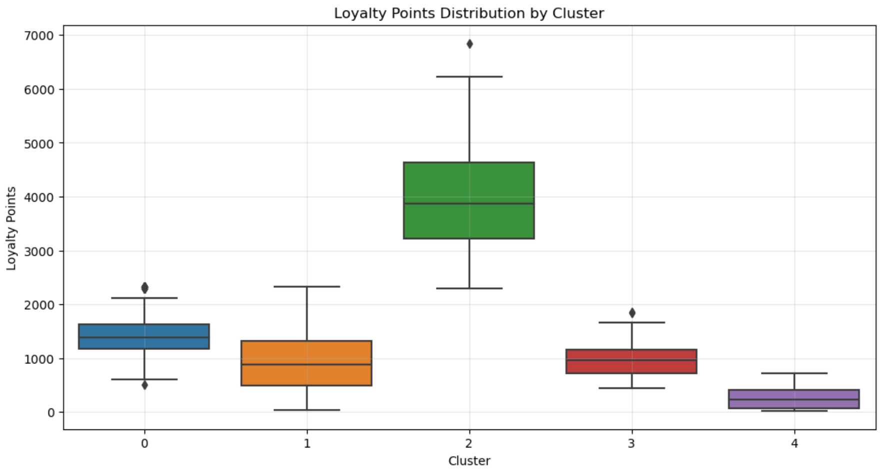


Fig: Boxplot of Cluser level distribution of Loyalty Points.

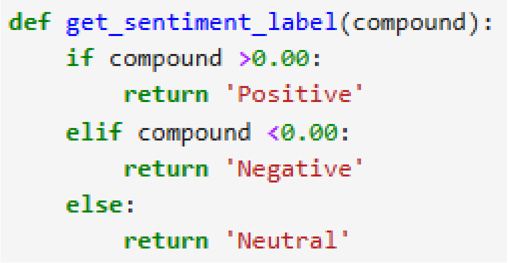
4.5 Sentiment Analysis (NLP):

Method [Python]: `nltk.sentiment.vader, TextBlob, WordCloud.`

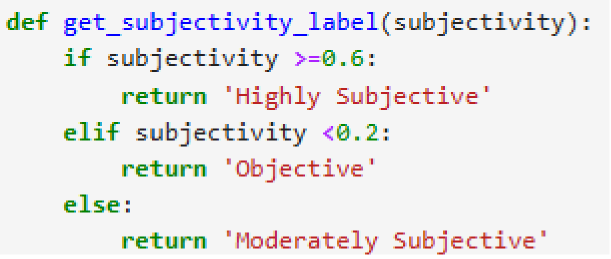
Inputs: Customer review text from the 'review' column

Process and Optimization:

1. Summary and Review columns were cleaned, checked for duplicates, and tokenized for sentiment analysis. No duplicates were removed as they were linked to different product IDs.
2. VADER SIA applied on Summary and Review texts, and sentiments labelled (Positive, Negative, Neutral) based on polarity compound scores.

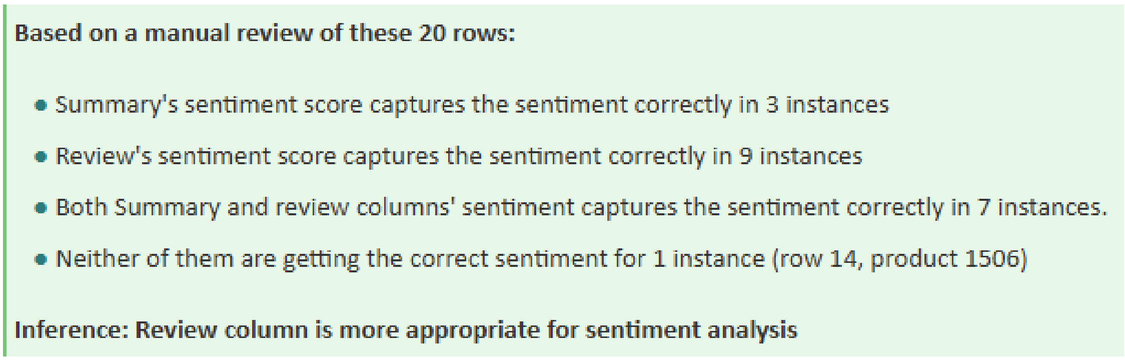


```
def get_sentiment_label(compound):
    if compound >0.00:
        return 'Positive'
    elif compound <0.00:
        return 'Negative'
    else:
        return 'Neutral'
```



```
def get_subjectivity_label(subjectivity):
    if subjectivity >=0.6:
        return 'Highly Subjective'
    elif subjectivity <0.2:
        return 'Objective'
    else:
        return 'Moderately Subjective'
```

3. Manual checks on 20 rows confirmed Review data as more accurate for sentiment identification.

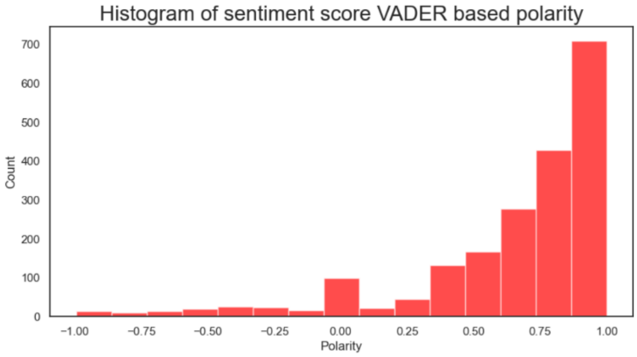


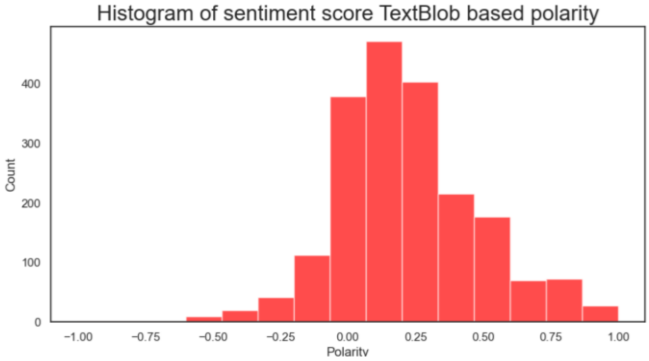
Based on a manual review of these 20 rows:

- Summary's sentiment score captures the sentiment correctly in 3 instances
- Review's sentiment score captures the sentiment correctly in 9 instances
- Both Summary and review columns' sentiment captures the sentiment correctly in 7 instances.
- Neither of them are getting the correct sentiment for 1 instance (row 14, product 1506)

Inference: Review column is more appropriate for sentiment analysis

4. TEXTBLOB's polarity and subjectivity analysis was applied to Review text, labelling sentiment based on polarity scores.
5. Histograms showed differing polarity distributions between TEXTBLOB and VADER.





6. A subset with mismatched VADER and TEXTBLOB sentiments was identified; 40 reviews were manually checked.
7. VADER correctly classified 29/40 reviews, while TEXTBLOB classified 9/40. With overall data having positive sentiment, VADER was deemed more effective, and its polarity scores were retained.
8. TOP 20 Positive, Negative Sentiments and 15 most frequent words were identified, with word clouds generated from cleaned Review texts.
9. Product- and Customer-level grouping of VADER polarity and TEXTBLOB subjectivity scores was performed.

5. Insights and Recommendations:

Loyalty Drivers and Statistical Significance

Insights:

1. Spending Score and Remuneration are the strongest predictors of customer loyalty. While Age shows a weak negative correlation, it remains statistically significant in the regression model.
2. The WLS model is statistically robust and can be effectively used for customer relationship management.

Recommendations:

- Investigate lower engagement among younger customers and develop tailored strategies.
- Use the WLS model (Spending Score, Remuneration, and Age) to predict loyalty points for new customers, supporting CRM and resource allocation.
- Validate the WLS model with external or future data and monitor periodically for performance degradation.
- Refine the prediction model with broader datasets.

Customer Segmentation

Insights:

1. The Decision Tree highlights spending score and remuneration as the most critical features.
2. K-Means Clustering identifies clear customer segments based on spending score and remuneration.

Recommendations:

- Implement tailored loyalty programs and personalize marketing, product recommendations, and benefits for each segment.
- Introduce tiered loyalty benefits, including exclusive products/services, with clear upgrade paths and targeted incentives. A gamified approach for tiers and upgrades could enhance branding.
- Continuously monitor segment migration to assess strategy effectiveness.
- Analyse segment-specific purchase patterns to identify churn risks and design proactive re-engagement strategies.

Sentiment Analysis

Insights:

1. Most customer reviews express positive sentiment towards Turtle Games products.
2. Comments are moderately subjective, indicating a good balance of fact and opinion.
3. Positive and negative keyword analysis highlights gameplay, age appropriateness, product quality, material quality, and ease of use as key features customers review.
4. Average product sentiment scores show overall positive sentiment across all products.
5. Customer persona-based sentiment analysis reveals overwhelmingly positive attitudes across clusters, with Basic Buyers at 86% positive sentiment and Premium Buyers at 90% (5% negative).
6. The VADER sentiment analyser performed better than TEXTBLOB for the provided review set.

Recommendations:

- Implement a rating system for features like product quality, shopping experience, and game experience alongside the review section for better insights, as reviews were often summarized with ratings (e.g., five stars).
- Invest in advanced NLP techniques fine-tuned for Turtle Games' review data. VADER's superior performance over TEXTBLOB highlights the need for model tuning to extract actionable insights.
- Deeper product-level and customer-segment sentiment analysis can uncover unique traits and interests, guiding product development and improving customer satisfaction.

6. Data Limitations:

- Lack of Customer ID in the dataset increases the possibility of duplicate customer profiles being analysed.
- Loyalty points mechanism has not been clarified – do they have expiration dates, the redeeming structure.
- Reviews / purchase history with a timestamp can help identify trends related to sentiment analysis.
- Family composition can be a useful customer detail in identifying purchase patterns in time, spend amount and product preferences.
- Data quality issues observed in Age and Education restricts their potential usage in analysis.
- Customer feedback forms can be enhanced with metrics like ratings on different aspects of the product to improve feedback analysis.
- Product IDs to Product and Product Category mapping can be included to enhance product and purchase pattern analysis and improve the robustness in customer segmentation analysis and customer sentiment analysis.

Appendix 1 - References:

https://www.geeksforgeeks.org/how-to-test-for-multicollinearity-in-r/

Weighted Least Squares Regression in Python - GeeksforGeeks

https://www.statology.org/weighted-least-squares-in-r/

https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

https://youtu.be/QpzMWQvxXWk?si=H8Dv7vcPGHKNQxNp

https://www.analyticsvidhya.com/blog/2021/10/sentiment-analysis-with-textblob-and-vader/

https://stats.stackexchange.com/questions/798/calculating-optimal-number-of-bins-in-a-histogram

https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

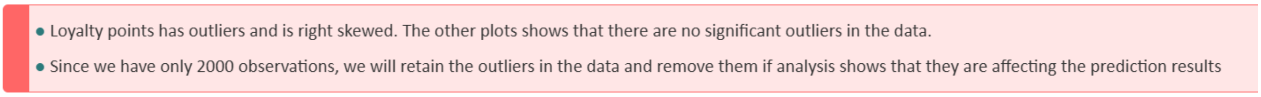
https://datagy.io/sklearn-decision-tree-classifier/

https://www.kaggle.com/code/alejopaullier/make-your-notebooks-look-better

Appendix 2 – Notebooks How to Use:

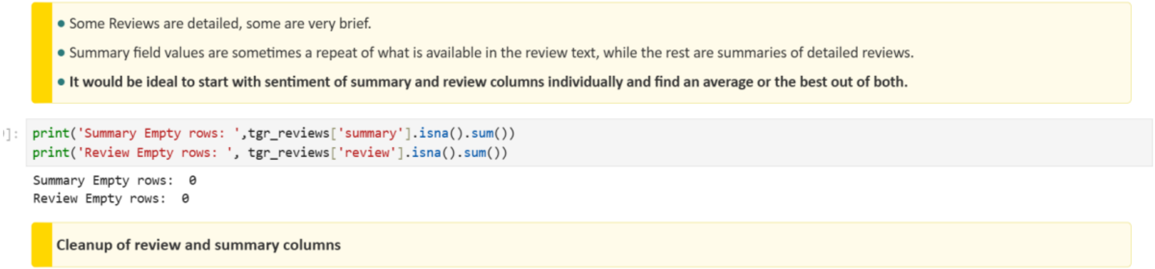
Notebook readability and decision-making steps within the data analysis process was improved by adding colour coded HTML – customized Markdown cells within both Python and R notebooks.

Red cells indicating potential issues



- Loyalty points has outliers and is right skewed. The other plots shows that there are no significant outliers in the data.
- Since we have only 2000 observations, we will retain the outliers in the data and remove them if analysis shows that they are affecting the prediction results

Yellow cells indicating observations and decisions



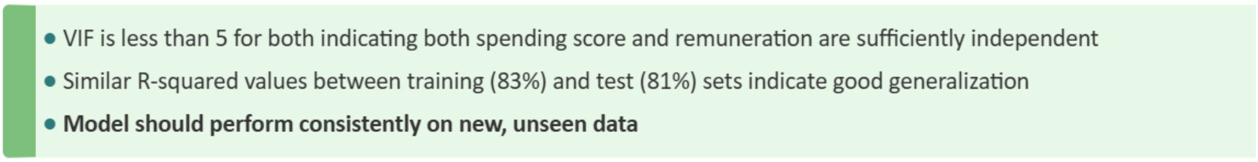
- Some Reviews are detailed, some are very brief.
- Summary field values are sometimes a repeat of what is available in the review text, while the rest are summaries of detailed reviews.
- **It would be ideal to start with sentiment of summary and review columns individually and find an average or the best out of both.**

```
print('Summary Empty rows: ',tgr_reviews['summary'].isna().sum())
print('Review Empty rows: ', tgr_reviews['review'].isna().sum())
```

```
Summary Empty rows:  0
Review Empty rows:  0
```

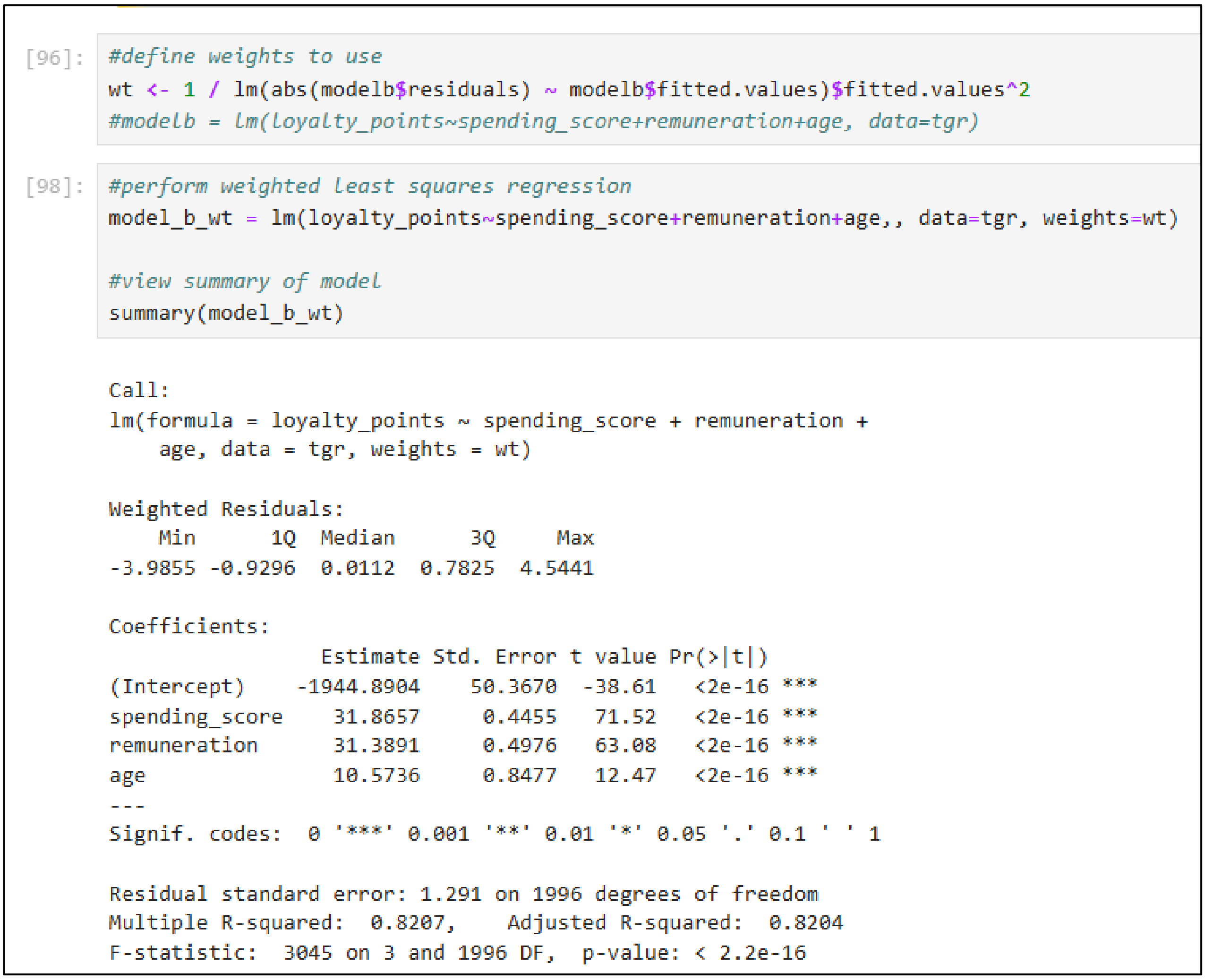
Cleanup of review and summary columns

Green cells indicating conclusions arrived at.

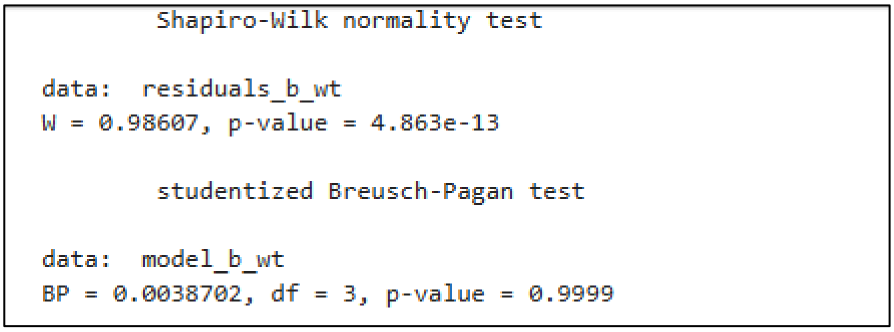


- VIF is less than 5 for both indicating both spending score and remuneration are sufficiently independent
- Similar R-squared values between training (83%) and test (81%) sets indicate good generalization
- **Model should perform consistently on new, unseen data**

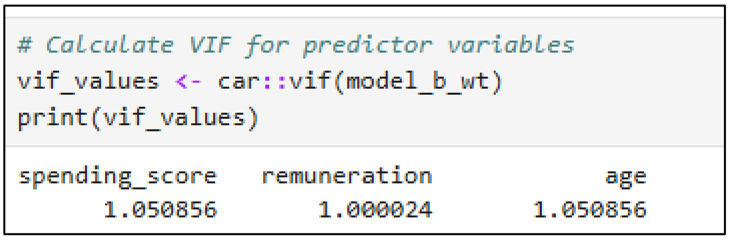
Appendix 3 – R Linear Regression Results & Stat tests(Weighted Linear Regression):



```
[96]: #define weights to use
      wt <- 1 / lm(abs(modelb$residuals) ~ modelb$fitted.values)$fitted.values^2
      #modelb = lm(loyalty_points~spending_score+remuneration+age, data=tgr)

[98]: #perform weighted least squares regression
      model_b_wt = lm(loyalty_points~spending_score+remuneration+age,, data=tgr, weights=wt)

      #view summary of model
      summary(model_b_wt)


      Call:
      lm(formula = loyalty_points ~ spending_score + remuneration +
          age, data = tgr, weights = wt)

      Weighted Residuals:
          Min      1Q  Median      3Q     Max
      -3.9855 -0.9296  0.0112  0.7825  4.5441

      Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
      (Intercept)    -1944.8904    50.3670  -38.61   <2e-16 ***
      spending_score    31.8657     0.4455   71.52   <2e-16 ***
      remuneration      31.3891     0.4976   63.08   <2e-16 ***
      age               10.5736     0.8477   12.47   <2e-16 ***
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      Residual standard error: 1.291 on 1996 degrees of freedom
      Multiple R-squared:  0.8207,    Adjusted R-squared:  0.8204
      F-statistic:  3045 on 3 and 1996 DF,  p-value: < 2.2e-16
```



```
        Shapiro-Wilk normality test

data:  residuals_b_wt
W = 0.98607, p-value = 4.863e-13

        studentized Breusch-Pagan test

data:  model_b_wt
BP = 0.0038702, df = 3, p-value = 0.9999
```



```
# Calculate VIF for predictor variables
vif_values <- car::vif(model_b_wt)
print(vif_values)

spending_score   remuneration            age
      1.050856       1.000024       1.050856
```