

|  |
| --- |
|  |
| Heilo sentiment analysys |
| Developing predictive model |



|  |  |
| --- | --- |
|  | Decorative |
| SUMMARY |
| The Helio Project manager required us to prioritize and speed up our sentiment analysis of the iPhone and the Galaxy over the other handsets in the shortlist. Helio must narrow down their device support to their customer’s most preferred device.  AWS has been used to collect large data and python scripts from filtering the data to create the large Matrix. The analytics team has provided the small Matrix manually to train our models.  Our task is to train our models with small Matrix and predict the sentiments for iPhone and Galaxy in large Matrix, |

|  |  |
| --- | --- |
|  | Decorative |
| iPhone Galaxy |
| Findings The raw data has been collected into small matrixes from 13000 web pages manually by the analytics team with mentions of pre-selected devices.  The analysis is based on identifying the sentiments related to the devices and their accessories. The following categorical ranking has been used for the analysis:  0: Unclear 1: Very negative 2: Somewhat Negative 3: Neutral 4: Somewhat Positive 5: Very Positive  The above histograms represent the raw web sentiments for each device. Both show high counts and positive sentiment, with a slight preference toward Samsung Galaxy devices over iPhones.  Model performances were evaluated via Kappa and accuracy, and the best performing model was used to predict sentiments on the large Matrix. |
| iPhone Galaxy |
| After applying the models to the large Matrix, we identified the majority of the sentiments are unclear. However, Galaxy won the majority of very positive sentiments. It is important to note that, Very positive and Unclear categories have the highest accuracy.  Decorative Confidence Three machine learning models have been used on the raw data for training and testing data sets: Random Forests Support Vector Machine(SVM) and Weighted K-Nearest Neighbors(KKNN). Additionally, three feature selection methods we evaluated to obtain a model with better performance: Near-Zero Variance(NZV), Correlation, and Recursive Feature Elimination(RFE).   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **iphone Model selelction** | | | **Galaxy Model selelction** | | | |  | Kappa | Accuracy |  | Kappa | Accuracy | | SVM | 41% | 70% | SVM | 35% | 69% | | KKNN | 30% | 15% | KKNN | 39% | 65% | | RF | 54% | 76% | RF | 48% | 75% | |  |  |  |  |  |  | |

Random Forest Has the highest accuracy and Kappa level in both iPhone and Galaxy Raw Data. So the Rf has been chosen as the model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **iPhone Data selection** | | | **Galaxy Data selection** | | |
|  | Kappa | Accuracy |  | Kappa | Accuracy |
| COR | 49% | 75% | COR | 48% | 75% |
| NZV | 53% | 76% | NZV | 50% | 76% |
| RFE | 56% | 77% | RFE | 48% | 75% |

Recursive Feature Elimination(RFE) has been selected for iPhone sentiment due to the highest accuracy and Kappa. The highest accuracy Kappa and accuracy have chosen near-zero variance(NZV) for galaxy data selection.

|  |  |  |
| --- | --- | --- |
|  |  | Decorative |
|  |  | Impact of findings |
| We had a good performance in the predictive models on the Large Matrix, over 75% for both iPhone and Galaxy except for negative categories. Although the user sentiments on the large Matrix were largely classified as ‘Unclear,’ both iPhone and Galaxy users are more likely to have positive sentiments than negative sentiments. Based on our findings, Galaxy is the winning device. Methodology EDA and Analytics have performed in R studio. Modeling has been the most time-consuming process. Eight different data frames, as mentioned above was evaluated. The most useful ones were RFE and RF for iPhone sentiments and NVZ and RF for Galaxy sentiments. The goal was to achieve the best Kappa and Accuracy with the smallest number of features. All the data frames were split 70/30 for evaluation. We also used parallel processing in R to accelerate model creation. The models used were RF, SVM, and KKNN; scores stored through the “resample” function in R |