MVP: Evergreen or Ephemeral?

Background:

The goal of this project is to build a classification model for identifying evergreen vs ephemeral articles. The practical application of this model would be to help StumbleUpon - a user-curated web content discovery engine - improve on the quality of articles recommended by systematically determining if an article is valuable to its users or not.

Given the classes are balanced, the model will be optimized by looking at accuracy while keeping an eye on F1, Recall,Precision and ROC_AUC. It is particularly important to have high precision in order to avoid misclassifying articles as evergreen when they are not.

Data Gathering and Cleaning:

The data is gathered from Kaggle and consists of 7000+ rows, with one row corresponding to one URL that StumpleUpon recommended to their users. Each URL in the dataset is labeled as evergreen (1) or not (0) - this is the target.

Initial cleaning consisted of checking for duplicates, imputing null values where needed, renaming columns for clarity,

Preliminary Findings:

a. **Baseline Models KNN and Logistic Regression perform comparably and show that there's room for improvement. Precision is lower than recall meaning false positives are an issue.**

After initial feature selection the baseline model contains the following features: is_news, html_ratio,numwords_in_url, numberOfLinks, parametrizedLinkRatio, image_ratio, commonlinkratio_2, commonlinkratio_4, compression_ratio, commonlinkratio_3 spelling_errors_ratio, avglinksize, linkwordscore,non_markup_alphanum_characters. frameTagRatio.

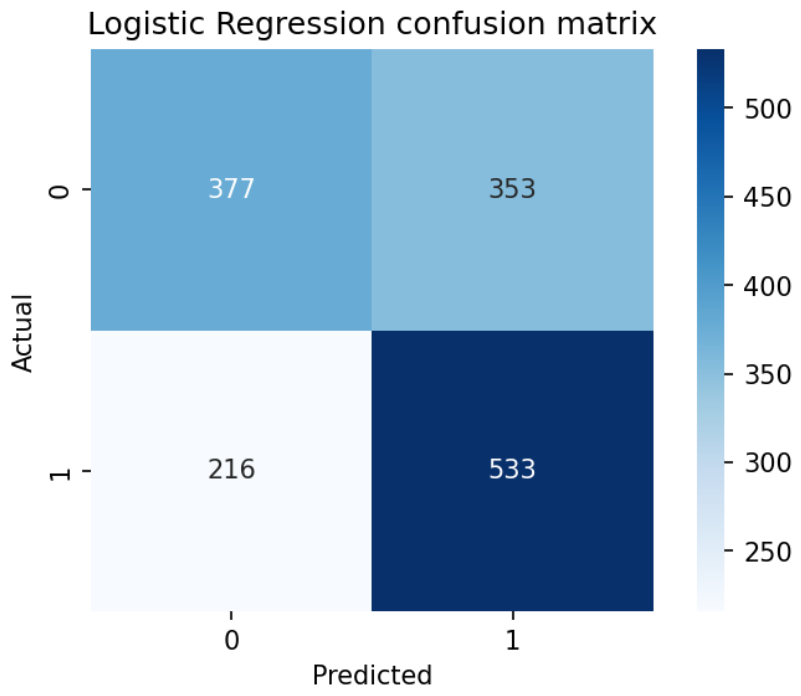
Both Logistic Regression and KNN were tested using grid search/ cross validation for parameters tuning. Both models show that there's room for improvement, with accuracy around 60%.

i. Logistic Regression

Baseline Logistic Regression with no Regularization yielded the following results:

	Accuracy	F1	Recall	Precision	ROC_AUC
Train	62.39%	66.14%	72.43%	61.44%	66.92%
Validation	62.31%	66.30%	72.01%	61.45%	66.57%

Confusion Matrix on Test

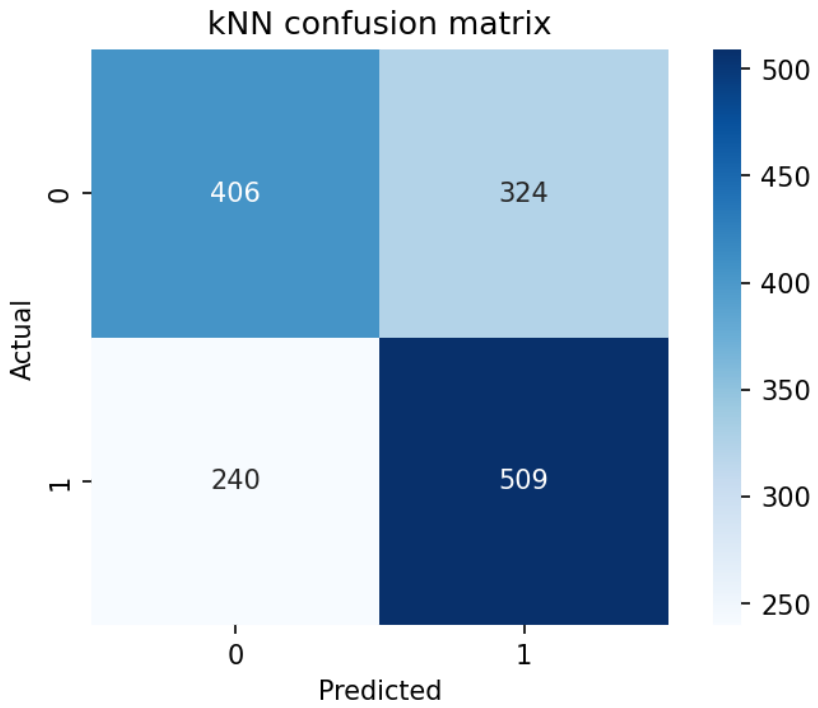


ii.KNN performs slightly better, however recall is lower and it seems to be slightly overfitting,meaning increasing the number of neighbors might be beneficial.

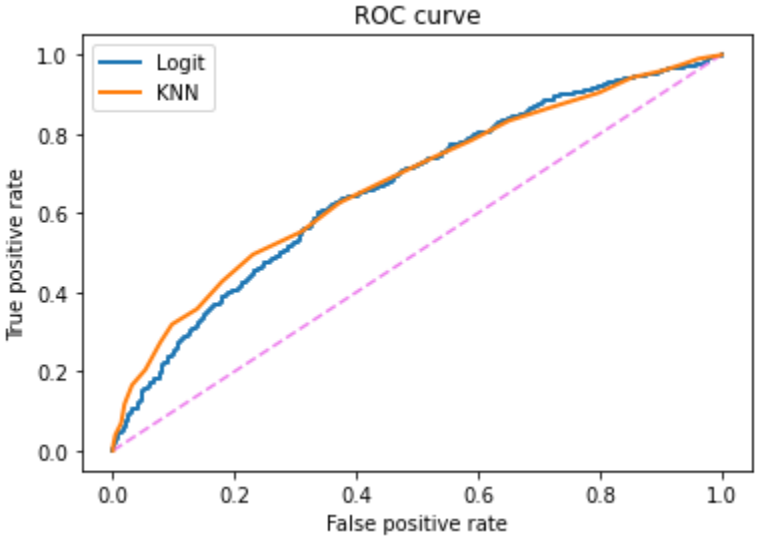
After running gridsearch to find the best parameters, the following results were gathered:

	Accuracy	F1	Recall	Precision	ROC_AUC
Train	66.73%	69.06%	72.10%	66.27%	73.5%
Validation	63.45%	66.16%	69.41%	63.22%	68.68%

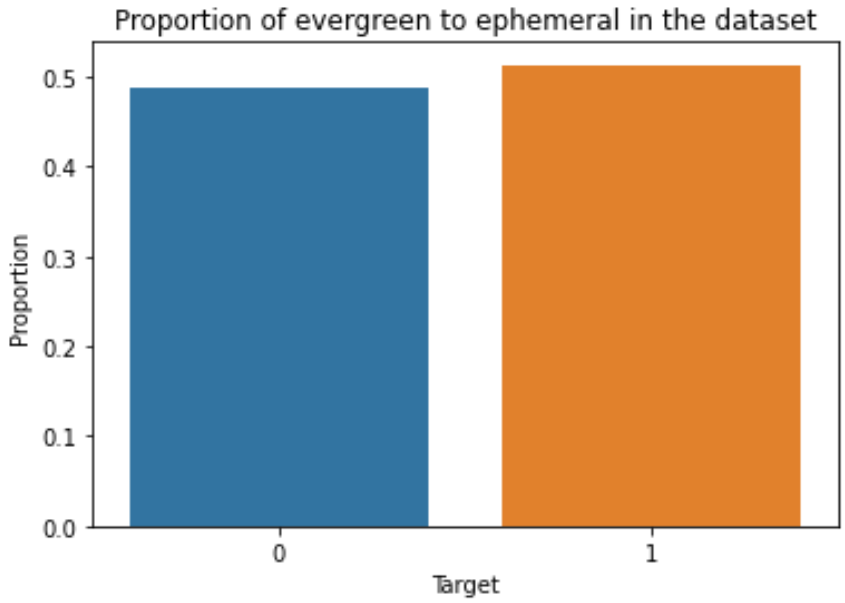
Confusion Matrix on Test



iii.<u>ROC AUC Comparison shows very similar performance for both baseline models</u>



b. **Classes are balanced, meaning there's no need to utilize imbalance techniques, however given negative is less frequent, would it be uself to optimize for precision?**



Next Steps:

a. **Feature Engineering:**

Three strategies of feature engineering will be tested to try improving the model:

- Layering in categoricals including but not limited to alchemy category

- Leveraging industry insights to engineer new features. For example, an analysis of 3.6 billion articles by Backlink showed that articles that are most likely to be evergreen are guides, how-to's and lists while articles that are less likely to be evergreen are reports and statistics that are likely to be updated.
- TF-IDF for text

b. <u>Increase Model Complexity:</u>

- The analysis can be expanded to more advanced models such as Decision Trees, Random Forest, XGBoost…