**Sentiment analysis for marketing**

Project Title: Sentiment analysis for marketing

Phase 4: Devolopement Part 2

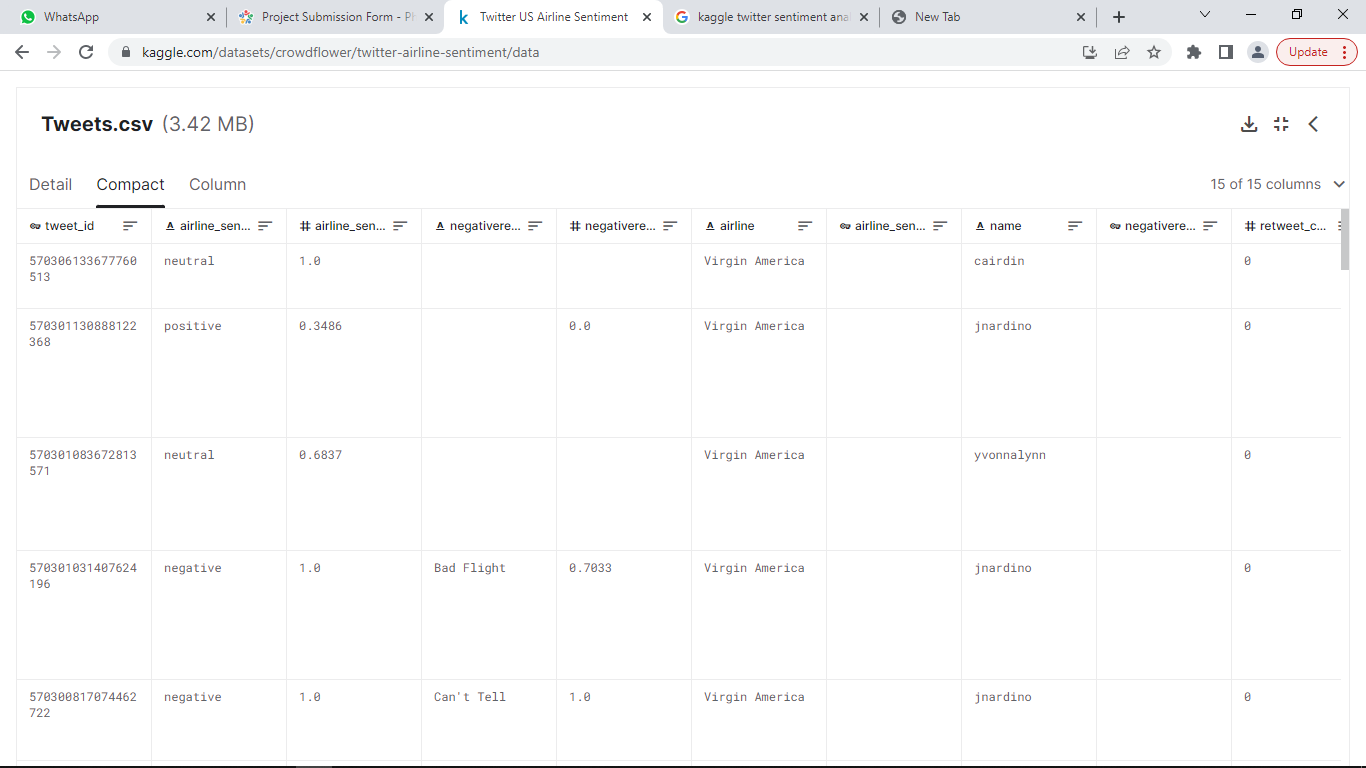
Topic**: To begin building the process of sentiment analysis for marketing**

Sentiment analysis for marketing

Introduction:

1. **Parsing online feedback can be a challenging task for any business with a strong social presence. In marketing, sentiment analysis can be useful for teams that want to examine commentary about their brands from a qualitative angle. By looking at the tone and content of social sentiments, you can develop enhanced metrics that may offer more valuable insight.**
2. **In this article, we define sentiment analysis in marketing, describe its benefits and offer tips to help you incorporate it into your team’s marketing plan.**
3. **Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online. This method is more comprehensive than traditional online marketing tracking, which measures the number of online interactions that customers have with a brand, like comments and shares. Using sentiment analysis, you can label individual interactions as positive, negative or neutral. Once you've figured out how to determine and track these labels, you can use this new data set for a variety of marketing purposes, including your online strategy.**

**Given data set:**

****

Overview of the process:

The following is an overview of the process of building a sentiment analysis for marketing by feature selection, model training, and evaluation:

1. Prepare the data: This includes cleaning the data, removing outliers, and handling missing values.
2. Perform feature selection: This can be done using a variety of methods, such as correlation analysis, information gain, and recursive feature elimination.
3. Train the model: There are many different machine learning algorithms that can be used for house price prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.
4. Evaluate the model: This can be done by calculating the mean squared error (MSE) or the root mean squared error (RMSE) of the model's predictions on the held-out test set.
5. Deploy the model: Once the model has been evaluated and found to be performing well, it can be deployed to production so that it can be used to predict the house prices of new houses.

PROCEDURE:

Feature selection:

1. Identify the target variable. This is the variable that you want to predict, such as house price.
2. Explore the data. This will help you to understand the

relationships between the different features and the target variable. You can use data visualization and correlation analysis to identify features that are highly correlated with the target variable.

1. Remove redundant features. If two features are highly correlated with each other, then you can remove one of the features, as they are likely to contain redundant information.
2. Remove irrelevant features. If a feature is not correlated with the target variable, then you can remove it, as it is unlikely to be useful for prediction.

Feature Selection:

We are selecting numerical features which have more

than 0.50 or less than -0.50 correlation rate based on Pearson Correlation Method—which is the default value of parameter "method" in corr() function. As for selecting categorical features, I selected the categorical values which I believe have significant effect on the target variable such as Heating and MSZoning.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

In [1]:

|  |  |  |  |
| --- | --- | --- | --- |
| important\_num\_cols = list(df.corr()["SalePrice"][(df.corr()["SalePrice"]>0.5 | | | |
| 0)|(df.corr()["SalePrice<-0.50)].index) | |  | |
| cat\_cols=["MSZoning","Utilities","BldgType","Heating","KitchenQual"," | | |
| SaleCondition","LandSlop"] |  | |
|  |  |  |  |

important\_cols = important\_num\_cols + cat\_cols

df = df[important\_cols]

*in# DataFrame*

import pandas as pd

*# Matplot*

import matplotlib.pyplot as plt

%matplotlib inline

*# Scikit-learn*

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score

from sklearn.manifold import TSNE

from sklearn.feature\_extraction.text import TfidfVectorizer

*# Keras*

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad\_sequences

from keras.models import Sequential

from keras.layers import Activation, Dense, Dropout, Embedding, Flatten, Conv1D, MaxPooling1D, LSTM

from keras import utils

from keras.callbacks import ReduceLROnPlateau, EarlyStopping

*# nltk*

import nltk

from nltk.corpus import stopwords

from nltk.stem import SnowballStemmer

*# Word2vec*

import gensim

*# Utility*

import re

import numpy as np

import os

from collections import Counter

import logging

import time

import pickle

import itertools

*# Set log*

logging.basicConfig(format='**%(asctime)s** : **%(levelname)s** : **%(message)s**', level=logging.INFO)

Using TensorFlow backend.

*# DATASET*

DATASET\_COLUMNS = ["target", "ids", "date", "flag", "user", "text"]

DATASET\_ENCODING = "ISO-8859-1"

TRAIN\_SIZE = 0.8

*# TEXT CLENAING*

TEXT\_CLEANING\_RE = "@\S+|https?:\S+|http?:\S|[^A-Za-z0-9]+"

*# WORD2VEC*

W2V\_SIZE = 300

W2V\_WINDOW = 7

W2V\_EPOCH = 32

W2V\_MIN\_COUNT = 10

*# KERAS*

SEQUENCE\_LENGTH = 300

EPOCHS = 8

BATCH\_SIZE = 1024

*# SENTIMENT*

POSITIVE = "POSITIVE"

NEGATIVE = "NEGATIVE"

NEUTRAL = "NEUTRAL"

SENTIMENT\_THRESHOLDS = (0.4, 0.7)

*# EXPORT*

KERAS\_MODEL = "model.h5"

WORD2VEC\_MODEL = "model.w2v"

TOKENIZER\_MODEL = "tokenizer.pkl"

ENCODER\_MODEL = "encoder.pkl"

df.head(5)

output:

Feature Engineering:

Feature engineering is a crucial aspect of building a house price prediction model using machine learning. It involves creating new features, transforming existing ones, and selecting the most relevant variables to improve the model's predictive power. Here are some feature engineering ideas for house price prediction:

1.Total Area Features:

Combine individual room areas to create features like "Total Living Area," "Total Bedroom Area," or "Total Bathroom Area." These can be significant predictors of house price.

2.Ratio Features:

Create features that represent ratios, such as the "Bedroom to Bathroom Ratio" or "Living Area to Lot Area Ratio." These ratios may capture the property's layout and functionality.

3.Age of the Property:

Calculate the age of the property by subtracting the construction year from the current year. Newer properties might have higher values.

4.Neighborhood Statistics:

Aggregate neighborhood-level statistics, such as the average income, crime rate, school ratings, or proximity to amenities, and use these as features.

5.Distance to Key Locations:

Calculate distances from the property to essential places like schools, parks, shopping centers, or public transportation hubs. Closer proximity to such amenities can affect the price.

6.Categorical Encodings:

Use techniques like one-hot encoding, label encoding, or target encoding for categorical variables, such as property type, heating system, or garage type.

7.Seasonal Features:

Create features indicating the season during which the house was sold. Seasonality can influence property demand and prices.

8.Historical Data:

Incorporate historical data on house prices and local real estate market trends. This can help the model account for cyclical patterns.

9.Exterior Features:

Develop features related to the property's exterior, such as the presence of a swimming pool, patio, or garden. These features can be valuable for determining a property's appeal.

10.Quality Scores:

Create a combined quality score by aggregating the quality ratings of various components of the property, such as kitchen quality, bathroom quality, and overall house quality.

11.Logarithmic Transformations:

Apply logarithmic transformations to features like "Lot Area" or "Number of Bedrooms" to make their distributions more normal.

12.Interaction Features:

Create interaction terms by multiplying or dividing relevant features. For example, "Number of Bathrooms" multiplied by "Total Living Area" can represent the total bathroom area.

13.Missing Value Indicators:

Create binary indicators for missing values in the dataset. The presence of missing data can be an informative feature.

14.Density Features:

Compute population density in the neighborhood or the density of certain property types. High density might impact property prices.

15.Sentiment Analysis:

Analyze online reviews or social media sentiment related to the property or neighborhood to capture public perception.

16.Time-Related Features:

Incorporate time-related features like day of the week, month, or year when the property was listed or sold.

17.Zoning Information:

Include zoning information that can affect property use, such as residential, commercial, or mixed-use zoning.

18.Accessibility Features:

Create features to represent accessibility, like the number of nearby public transport stations or major highways.

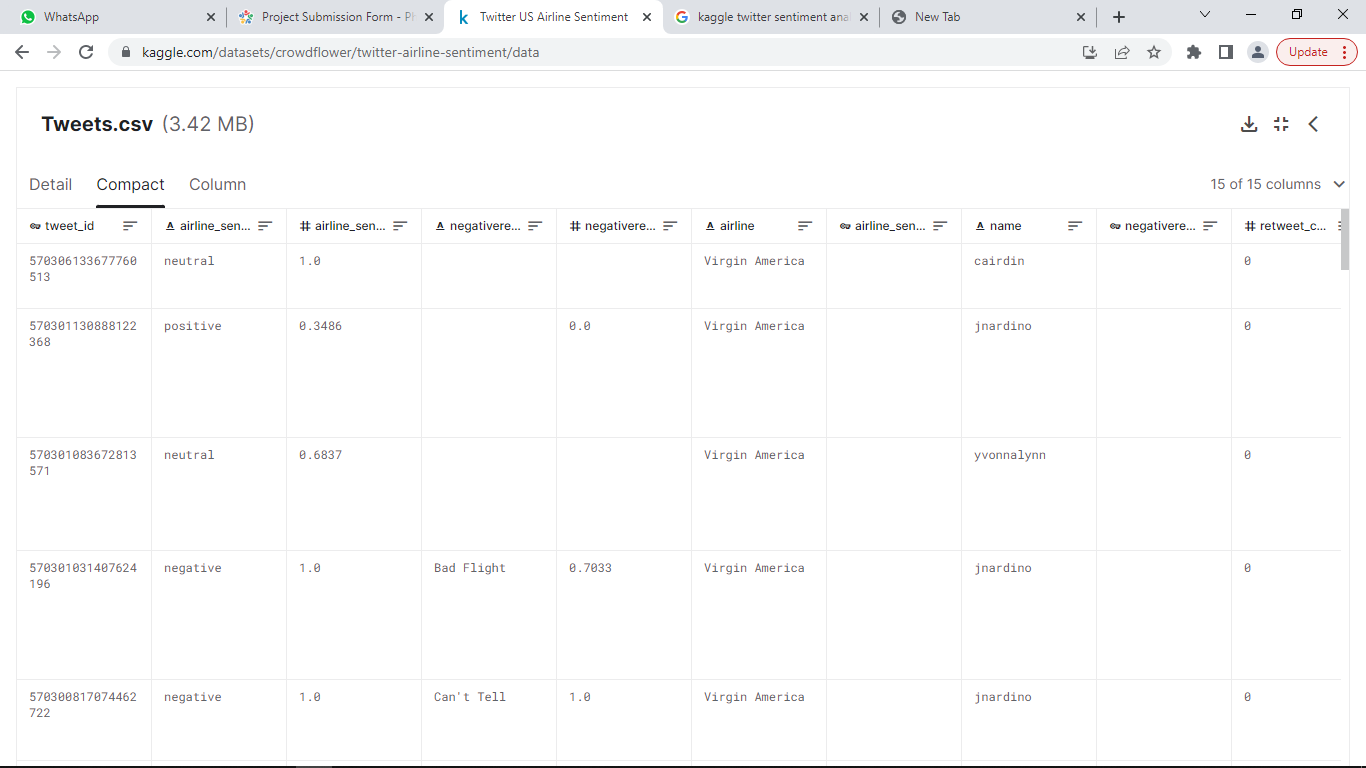
19.Energy Efficiency:

Include features related to energy-efficient components, such as insulation, energy-efficient appliances, or solar panels.

20.Demographic Data

Use demographic data for the area to understand the potential buyer's income levels, family sizes, and preferences.

Various feature to perform model training:



1. Use a variety of feature engineering techniques.  
     
   Feature engineering is the process of transforming raw data into features that are more informative and predictive for machine learning models. By using a variety of feature engineering techniques, you can create a set of features that will help your model to predict house prices more accurately.
2. Use cross-validation.  
     
   Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use crossvalidation to evaluate the performance of your model during the training process. This will help you to avoid overfitting and to ensure that your model will generalize well to new data.
3. Use ensemble methods.  
     
   Ensemble methods are machine learning methods that combine the predictions of multiple models to produce a more accurate prediction. Ensemble methods can often achieve better performance than individual machine learning models.
4. Use cross-validation.  
     
   Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use crossvalidation to evaluate the performance of your model during the evaluation process. This will help you to avoid overfitting and to ensure that the model will generalize well to new data.
5. Use a holdout test set.
   1. holdout test set is a set of data that is not used to train orevaluate the model during the training process. This data is used to evaluate the performance of the model on unseen data after the training process is complete.
6. Compare the model to a baseline.
   1. baseline is a simple model that is used to compare theperformance of your model to. For example, you could use the mean house price as a baseline.
7. Analyze the model's predictions.  
     
   Once you have evaluated the performance of the model, you can analyze the model's predictions to identify any patterns or biases. This will help you to understand the strengths and weaknesses of the model and to improve it.  
     
   **Conclusion:**

Product ratings and chatter are the gold standards that drive online sales and higher conversion rates. Finding a quantifiable, measurable way to analyze and impact them is imperative.

Sentiment analysis is an incredibly useful tool for extracting information, but when you pair it with other forms of software, the true strengths start to shine through.

With AI-powered engines capable of using machine learning to grow and expand when new factors are introduced, sentiment analysis software will continue to grow and adapt to the language, slang, and syntax changes.

This constant evolution will help sentiment analysis keep up with the growth of ecommerce ratings and reviews, offering a way to align with the top of mind of customers in your industry and what they like and dislike. This is done by leveraging sentiment analysis across retailers, brands, and products. With this, you can drive conclusions as to what drives product rating success (or failure):