Author: ManikantaReddy Annem

MC DONALD'S CUSTOMER REVIEW ANALYSIS

1. ABSTRACT:

This study uses machine learning techniques to analyze McDonald's customer happiness. With sentiment analysis and predictive modeling, we look into what affects satisfaction based on online reviews. Two examples of preliminary data preprocessing are feature extraction and text normalization. Predictive modeling identifies important data such as store details and time-related components, while sentiment analysis offers insights into customer attitudes. Accuracy, F1 score, and MSE are some of the performance evaluation measures used to validate the effectiveness of the model. More machine learning techniques and sensitivity analysis are considered during subsequent research to enhance forecasting abilities.

I hope that this study project will provide insightful information to McDonald's stakeholders, such as management, marketing departments, and customer service representatives. I aim to offer practical suggestions for improving customer happiness and streamlining operational methods by utilizing machine learning approaches. In the end, this research adds to the larger knowledge of customer sentiment analysis in the fast-food sector and emphasizes the significance of data-driven decision-making for enhancing the general customer experience.

Table Of Contents:

2. INTRODUCTION WITH RESEARCH QUESTION:

In a highly competitive fast-food business, keeping customers happy is the most important thing that can be done. As a worldwide leader in fast food, **McDonald's** knows how important it is to understand and improve customer happiness. To do this, this analysis study will use a large set of data from online reviews to investigate the factors that affect how happy McDonald's customers are. The goal is to find ideas that can be turned into strategies that can be used to improve operational processes, make the customer experience better, and make **McDonald's** more competitive.

2.1 Initial Research Question:

Research Question: "What factors influence customer satisfaction in McDonald's based on online reviews, and how can these insights be leveraged to enhance the overall dining experience?"

The study question seeks to apply the potential of data science and machine learning to address important business difficulties for McDonald's. A thorough dataset is built by examining the elements that influence customer satisfaction through online evaluations, which includes both quantitative ratings and qualitative insights. Data science approaches like statistical analysis and predictive modeling can be used to find trends, correlations, and important drivers of happiness. Machine learning algorithms can predict client preferences and behaviors, allowing for tailored recommendations and focused improvement strategies. The dataset's rich textual content enables sentiment analysis, providing a deep knowledge of client sentiments. Finally, the study question serves as a platform for adopting data-driven solutions that optimize operational processes, improve customer experiences, and boost McDonald's competitiveness in the fast-food business.

2.2 Dataset Attributes:

Column Id	Column Name	Data Type	Description
1	reviewer_id	int	Number for each reviewer
2	store_name	String	Name of the McDonald's store
3	category	String	Category or type of the store

4	store_address	String	Address of the store
5	latitude	Float	Latitude coordinate of the store's location
6	longitude	Float	Longitude coordinate of the store's location
7	rating_count	Integer	Number of ratings/reviews for the store
8	review_time	Timestamp	Timestamp of the review
9	review	Text	Textual content of the review
10	rating	Float	Rating provided by the reviewer

2.3 Final Research Question:

Final Research Question: "Through the application of machine learning techniques, specifically utilizing predictive modeling and sentiment analysis, what specific attributes—including store details (store_name, category, store_address, latitude, longitude), temporal aspects (review_time), and textual content (review)—significantly influence customer satisfaction, as measured by the rating in online reviews, for McDonald's stores?"

Evolution of Research Question: The final research question evolved from the initial research question by incorporating specific methodologies and variables based on the analysis conducted. While the initial research question was broader and focused on identifying factors influencing customer satisfaction, the final question became more precise, outlining the utilization of machine learning techniques and specifying the attributes of interest, such as store details, temporal aspects, and textual content. This evolution reflects a refinement and narrowing down of the research scope to facilitate a more focused and actionable analysis.

2.4 Overview of Analysis and Objectives:

1. The purpose: The purpose of the research question is to find out what makes McDonald's customers happy by reading reviews online and using that information to make the whole eating experience better. The goal of the analysis report is to give McDonald's suggestions for what

they can do after a full analysis of the dataset that includes both quantitative and qualitative factors.

2. Type of Report on Analysis: Full Report on Customer Satisfaction Analysis: The report's goal is to look at the dataset in depth by mixing statistical, machine learning, and sentiment analysis methods to learn more about what makes customers happy.

3. Target Audiences: McDonald's managers, including the marketing, operations, and customer experience teams. Data analysts, industry researchers, and other people with an interest in the fast-food business and customer satisfaction trends are the secondary audience.

4. Uses of the Analysis Report: Strategic Decision-Making: Help with the processes of making choices about how to improve operations, create marketing plans, and make the customer experience better. Operational Optimization: Figure out what can be done to make certain things better, like the quality of the food, the speed of service, the cleanliness, and the value for the money.
Competitive Edge: Make McDonald's more competitive in the fast-food business by fixing problems with customer happiness.

3. PLAN FOR ANALYZING DATA AND EVALUATION:

1. Opening and Reading Data: In this stage, the dataset is loaded into memory, and its contents are read. For our research, the McDonald's review dataset must be read from a CSV file using pandas.

2. Data Cleaning and Preprocessing: In data cleaning, duplicates are eliminated, missing values are located and handled, and dataset discrepancies are addressed. In order to get the data ready for analysis, our project involves steps like eliminating stopwords, lowercasing text, and deleting whitespace.

3. Feature Engineering: Feature engineering is the process of adding new features or altering current ones in order to increase the model's capacity for prediction. Features like review time, store location information, and numerical ratings are among the features that will be extracted for our project.

4. Summary Statistics: Measures of central tendency and dispersion, among other numerical

aspects of the dataset, are summarized in summary statistics. This part of our study entails computing summary statistics for numerical variables such as ratings.

5. Visualizations: Understanding the distribution of data, trends, and correlations between variables is aided by visualizations. In our project, we examine the dataset and find insights using visualizations like word clouds, box plots, and histograms.

6. Sentiment Analysis: Sentiment analysis is the process of identifying the sentiment—whether positive, negative, or neutral—expressed in textual data. Our project involves sentiment analysis of McDonald's reviews utilizing natural language processing techniques.

7. Statistical Analysis: Using statistical techniques, statistical analysis investigates links and patterns within the data. Our project uses methods to comprehend the link between variables, such as correlation analysis.

8.Evaluation of Performance Metrics: This process evaluates the effectiveness of predictive models by calculating metrics such as F1 score, accuracy, precision, recall, and Mean Squared Error (MSE). As part of our project, we'll be measuring the efficacy and correctness of sentiment analysis models through performance evaluation.

4. EXPLORATORY DATA ANALYSIS (EDA)

Dataset Purpose: This information is meant to be carefully analyzed so that we can get a full picture of the things that affect McDonald's customers' happiness. The dataset allows for a thorough study of the dining experience because it includes both numerical ratings and qualitative comments.

The dataset has **33396** records. I have performed the following steps to my dataset as part of Exploratory Data Analysis (EDA)

1) Opening and Reading Data
2) Data Cleaning and Preprocessing
3) Feature Engineering
4) Summary Statistics
5) Visualizations
6) Sentiment Analysis
7) Statistical Analysis
8) Performance Metrics Evaluation

4.1 Opening and Reading Data:

Using the Pandas library, the McDonald's Reviews dataset is loaded into a DataFrame named df. The dataset is read from a CSV file named "McDonald_s_Reviews.csv", with the encoding specified as 'latin1' to handle any special characters present in the data.

```
In [4]: df = pd.read_csv("McDonald_s_Reviews.csv", encoding = 'latin1')
```

```
In [5]: df
```

Out[5]:

	reviewer_id	store_name	category	store_address	latitude	longitude	rating_count	review_time	review	rating
0	1	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	3 months ago	Why does it look like someone spit on my food?...	1 star
1	2	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	5 days ago	It'd McDonalds. It is what it is as far as the...	4 stars
2	3	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	5 days ago	Made a mobile order got to the speaker and che...	1 star
3	4	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	a month ago	My mc. Crispy chicken sandwich was ï¿½ï¿½ï¿½ï¿½ï¿...	5 stars
4	5	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	2 months ago	I repeat my order 3 times in the drive thru, a...	1 star
...	...	...	...	...	...	...	...	...	...	...
33391	33392	McDonald's	Fast food restaurant	3501 Biscayne Blvd, Miami, FL 33137, United St...	25.810000	-80.189098	2,810	4 years ago	They treated me very badly.	1 star
33392	33393	McDonald's	Fast food restaurant	3501 Biscayne Blvd, Miami, FL 33137, United St...	25.810000	-80.189098	2,810	a year ago	The service is very good	5 stars
33393	33394	McDonald's	Fast food restaurant	3501 Biscayne Blvd, Miami, FL 33137, United St...	25.810000	-80.189098	2,810	a year ago	To remove hunger is enough	4 stars
33394	33395	McDonald's	Fast food restaurant	3501 Biscayne Blvd, Miami, FL 33137, United St...	25.810000	-80.189098	2,810	5 years ago	It's good, but lately it has become very expen...	5 stars
33395	33396	McDonald's	Fast food restaurant	3501 Biscayne Blvd, Miami, FL 33137, United St...	25.810000	-80.189098	2,810	2 years ago	they took good care of me	5 stars

33396 rows × 10 columns

4.2 Data Cleaning and Preprocessing:

- **Handling Missing Values:** The code first checks for missing values in the DataFrame ‘df’ using the ‘isnull()’ method, which returns a Series with the count of missing values for each column. Then, the ‘dropna()’ method is applied to remove rows with missing values, effectively handling them by eliminating any observations with NaN values. After this operation, the DataFrame ‘df’ is updated to exclude rows containing missing values, ensuring that the dataset is free from incomplete entries. This approach helps maintain data integrity and avoids potential issues during analysis caused by missing or incomplete data. Finally, the variable ‘missing_values’ stores the count of missing values for each column, providing insight into the extent of missing data in the original dataset.

1. Data Preparation:

In the line df = df.dropna(), missing values in the dataset are addressed by removing entire rows where at least one value is missing. This is a common strategy when missing values are limited, and you prefer not to impute or estimate them.

1. Handle missing values :

In [13]:
```
# Checking for missing values
missing_values = df.isnull().sum()
#dropna() method is used to remove rows containing missing values (NaN) from a DataFrame.
df = df.dropna()
```

- **Standardizing Text Data:** To improve analysis accuracy and guarantee uniformity, the text data is standardized. To standardize the format of the text throughout the dataset, it is first transformed to lowercase. After that, any extraneous characters are eliminated from the text by removing the leading and trailing whitespaces. In order to concentrate on the textual information that is pertinent to the study, numbers are also eliminated from the text. In order to extract relevant information for sentiment analysis and other text-based activities, stopwords—common words that might not add value to the analysis—are finally eliminated from the text. The text data is summarized and made ready for additional analysis and modeling with the aid of these preparation procedures.

2. Standardizing Text Data:

In [14]:
```
# Assuming 'review' is the column with textual content
# Convert text to lowercase for consistency
df['review'] = df['review'].str.lower()

# Remove leading and trailing whitespaces
df['review'] = df['review'].str.strip()

# Remove numbers from text
df['review'] = df['review'].apply(lambda x: ''.join([i for i in x if not i.isdigit()]))
```

In [9]:
```
# As the 'review' column is the with textual content
# Remove stopwords from text
stop_words = set(stopwords.words('english'))
df['review'] = df['review'].apply(lambda x: ' '.join([word for word in word_tokenize(x) if word.lower() not in stop_words]))
print(df['review'])
```

```
0        look like someone spit food ? normal transacti...
1        'd mcdonalds . far food atmosphere go . staff ...
2        made mobile order got speaker checked . line m...
3        mc . crispy chicken sandwich ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½...
4        repeat order times drive thru , still manage m...
                               ...
33391                                  treated badly .
33392                                     service good
33393                                remove hunger enough
33394                  's good , lately become expensive .
33395                                   took good care
Name: review, Length: 32736, dtype: object
```

4.3 Feature Engineering:

- During the feature engineering process, we initiated by duplicating the original dataset to carry out modifications without modifying the original data. Consequently, we eliminated extraneous columns such as 'store_name', 'category', 'latitude', 'longitude', and 'rating_count' in order to concentrate on crucial attributes for our research. This phase assured that our dataset exclusively consisted of pertinent information directly relevant to our research subject.

Subsequently, we created supplementary attributes from the available data, specifically the 'City' and 'State' fields obtained from the 'store_address' attribute. This enabled us to investigate geographical patterns in customer satisfaction.

3. Feature Engineering:

```
In [10]: df1 = df.copy()

In [11]: # Drop columns 'store_name', 'category', 'Latitude', 'Longitude', and 'rating_count'
         df1 = df1.drop(columns=['reviewer_id','store_name', 'category', 'latitude ', 'longitude', 'rating_count'])

In [12]: df1.head(2)
```

Out[12]:

	store_address	review_time	review	rating
0	13749 US-183 Hwy, Austin, TX 78750, United States	3 months ago	look like someone spit food ? normal transacti...	1 star
1	13749 US-183 Hwy, Austin, TX 78750, United States	5 days ago	'd mcdonalds . far food atmosphere go . staff ...	4 stars

Store Address

```
In [13]: df1[['store_address']].sample(4)
```

Out[13]:

	store_address
23362	5725 W Irlo Bronson Memorial Hwy, Kissimmee, F...
21387	By Mandalay Bay, 3999 S Las Vegas Blvd, Las Ve...
29226	1415 E State Rd, Fern Park, FL 32730, United S...
14780	111 Madison St, Oak Park, IL 60302, United States

```
In [14]: df1[['City', 'State']] = df1['store_address'].apply(lambda x: pd.Series(x.split(', ')[-3:-1]))

In [15]: df1[['City','State']].sample(4)
```

Out[15]:

	City	State
13902	Orlando	FL 32819
11952	Champlain	NY 12919

- In addition, we conducted an analysis of the 'review_time' column to examine temporal patterns in customer feedback. This allowed us to gain insights into any recurring patterns or trends that may impact customer satisfaction. Additionally, we converted the 'rating' column into a more comprehensible format by extracting the star ratings as a distinct attribute called 'Star', which simplifies the analysis and understanding of customer ratings. By performing feature engineering, we were able to enhance our dataset by incorporating significant information, making it ready for further modeling and analysis.

```
In [30]: df1 = df1.drop(columns=['store_address'])
```

Review Time

```
In [31]: df1[['review_time']].sample(5)
```

Out[31]:

	review_time
16963	5 years ago
3619	a year ago
26734	10 months ago
33379	5 years ago
26871	4 years ago

Rating

```
In [32]: df1[['rating']].sample(5)
```

Out[32]:

	rating
17791	5 stars
23164	2 stars
12296	5 stars
18703	5 stars
28540	5 stars

In [33]:
```
df1.insert(5, "Star", df1["rating"].str.split(" ").str[0])
```

In [34]:
```
df1[['Star']].sample(5)
```

Out[34]:

	Star
32636	1
26506	4
11633	4
289	5
10898	5

In [35]:
```
df1 = df1.drop(columns=['rating'])
```

Review

In [36]:
```
unique_review = df1['review'].unique()
unique_review
```

Out[36]:
```
array(['look like someone spit food ? normal transaction , everyone chill polite , dont want eat . im trying think milky white/
clear substance food , * * * sure coming back .',
       "'d mcdonalds . far food atmosphere go . staff make difference . friendly , accommodating always smiling . makes pleasan
t experience many fast food places .",
       'made mobile order got speaker checked . line moving leave otherwise iï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½d late work . never got refund app . called said could get money back person stuck system . wen
t person next day manager told wasnï¿½',
       ..., 'remove hunger enough', "'s good , lately become expensive .",
       'took good care'], dtype=object)
```

- In feature engineering, we implemented a function to clean text data by converting it to lowercase, removing non-alphabetic characters, and stripping extra whitespace. We then applied this function to the 'review' column to generate 'clean_reviews', eliminating stopwords and ensuring consistency for sentiment analysis.

In [37]:
```
def clean_review(review):
    review = review.lower()
    review = re.sub(r'[^a-zA-Z\s]', '', review)
    review = re.sub(r'\s+', ' ', review).strip()

    stop_words = set(stopwords.words('english'))
    review_tokens = nltk.word_tokenize(review)
    review = ' '.join([word for word in review_tokens if word not in stop_words])


    return review

df1['clean_reviews'] = df1['review'].apply(clean_review)

print(df1[['clean_reviews']])
```

```
                                       clean_reviews
0      look like someone spit food normal transaction...
1      mcdonalds far food atmosphere go staff make di...
2      made mobile order got speaker checked line mov...
3      mc crispy chicken sandwich customer service qu...
4      repeat order times drive thru still manage mes...
...                                                  ...
33391                                  treated badly
33392                                   service good
33393                             remove hunger enough
33394                   good lately become expensive
33395                                  took good care

[32736 rows x 1 columns]
```

- We performed a thorough study to evaluate the distinct values in specific columns and gain insights into the variety and spread of data across important characteristics. We calculated the total number of distinct values for each column, including 'City', 'State', 'review_time', and 'Star'. This analysis gives us insights into the level of variability in the dataset. Furthermore, through the analysis of the highest values and their corresponding frequencies, we obtained more detailed information on the significance of particular categories or timestamps. This allowed us to develop a more sophisticated comprehension of the dataset's structure and spread.

Unique Value

In [38]:
```
specified_columns = ['City', 'State', 'review_time', 'Star']

for col in specified_columns:
    total_unique_values = df1[col].nunique()
    print(f'Total unique values for {col}: {total_unique_values}')

    top_values = df1[col].value_counts()

    for i, (value, count) in enumerate(top_values.items()):

        print(f'{value}: {count}')

    print('\n' + '=' * 30 + '\n')
```

```
12 years ago: 4
21 hours ago: 2
23 hours ago: 1
6 hours ago: 1
20 hours ago: 1
22 hours ago: 1
8 hours ago: 1

==============================

Total unique values for Star: 5
5: 10059
1: 9305
4: 5646
3: 4706
2: 3020
```

4.4 Summary Statistics:

Summary statistics offer a concise overview of important attributes within the dataset, facilitating comprehension of its distribution and variability. Descriptive statistics, such as the mean, standard deviation, and quartile ranges, provide information on the average value, variability, and spread of numerical variables. Categorical variables are summarized by calculating frequency counts, which provide information on the prevalence of each category and their distribution in relation to each other. Temporal variables are displayed as a range, highlighting the earliest and latest timestamps to encompass the temporal extent of the dataset.

4. Summary Statistics:

In [39]:
```
# Numerical Variables
numerical_summary = df1.describe()
print("Numerical Variables: \n", numerical_summary)


# Categorical Variables
categorical_summary = df['category'].value_counts()
print("Categorical Variables:\n", categorical_summary)


# Temporal Variables
temporal_range = df['review_time'].min(), df['review_time'].max()
print("Temporal Variables:\n ", temporal_range)
```

```
Numerical Variables:
        review_time     review      City     State   Star clean_reviews
count         32736      32736     32736     32736  32736         32736
unique           39      21098        26        36      5         20413
top     4 years ago  excellent  New York  FL 32819      5     excellent
freq           6633       2175      3486      2380  10059          2178
Categorical Variables:
 category
Fast food restaurant    32736
Name: count, dtype: int64
Temporal Variables:
  ('10 months ago', 'a year ago')
```

4.5 Visualizations Of EDA:

- Visualizations are excellent resources for identifying and deciphering patterns in data. Graphical representations facilitate the easy interpretation and communication of intricate linkages and trends. Diverse kinds of visualizations have been used in the offered code snippets to examine unique features of the dataset. Pie charts and bar plots, which are examples of **univariate analysis**, show the distribution and frequency of categorical data like star ratings, states, and cities. These graphics provide information about the general distribution and frequency of categories in the dataset.

3. Visualizations Of EDA:

In [40]:
```
df2 = df1.copy()
```

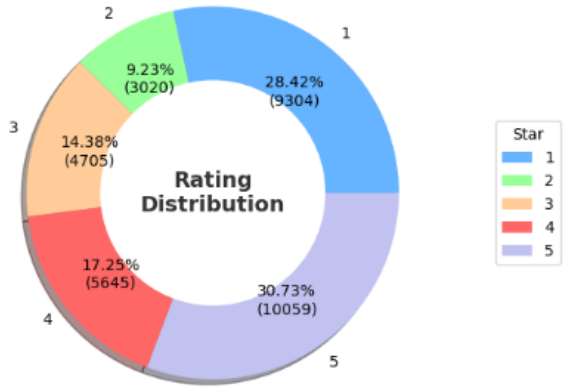
In [41]:
```
df2.head(3)
```

Out[41]:

	review_time	review	City	State	Star	clean_reviews
0	3 months ago	look like someone spit food ? normal transacti...	Austin	TX 78750	1	look like someone spit food normal transaction...
1	5 days ago	'd mcdonalds . far food atmosphere go . staff ...	Austin	TX 78750	4	mcdonalds far food atmosphere go staff make di...
2	5 days ago	made mobile order got speaker checked . line m...	Austin	TX 78750	1	made mobile order got speaker checked line mov...

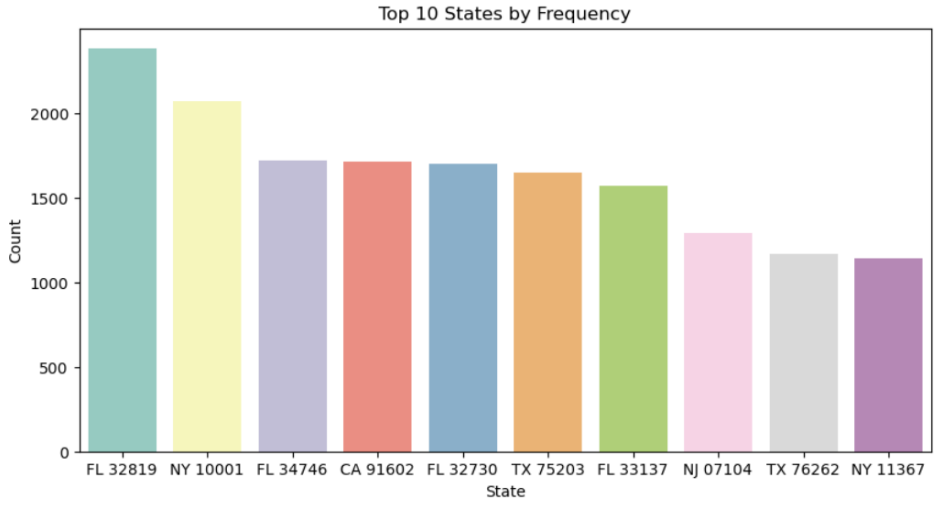
1. Univariate Analysis

In [42]:
```
unique_star = df1['Star'].unique()
explode = [0] * len(unique_star)
sentiment_counts = df2.groupby("Star").size()
colors = ['#66b3ff', '#99ff99', '#ffcc99', '#ff6666', '#c2c2f0']
fig, ax = plt.subplots()

wedges, texts, autotexts = ax.pie(
    x=sentiment_counts,
    labels=sentiment_counts.index,
    autopct=lambda p: f'{p:.2f}%\n({int(p*sum(sentiment_counts)/100)})',
    wedgeprops=dict(width=0.7),
    textprops=dict(size=10, color="black"),
    pctdistance=0.7,
    colors=colors,
    explode=explode,
    shadow=True
)
center_circle = plt.Circle((0, 0), 0.6, color='white', fc='white', linewidth=1.25)
fig.gca().add_artist(center_circle)

ax.text(0, 0, 'Rating\nDistribution', ha='center', va='center', fontsize=14, fontweight='bold', color='#333333')

ax.legend(sentiment_counts.index, title="Star", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))

ax.axis('equal')

plt.show()
```



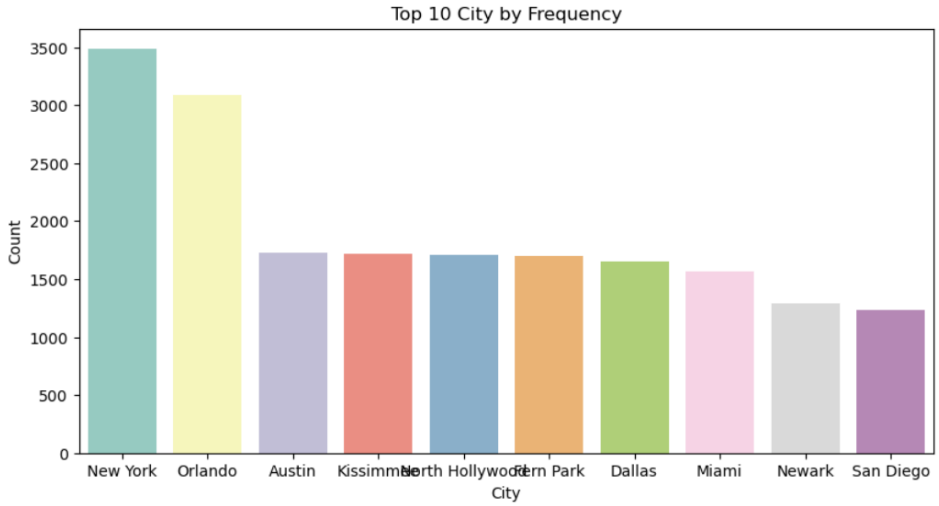
Top 10 States by Frequency:

In [43]:
```
top_10_states = df2['State'].value_counts().nlargest(10)

plt.figure(figsize=(10, 5))
sns.set_palette("Set3")
sns.countplot(x='State', data=df2, order=top_10_states.index)
plt.title('Top 10 States by Frequency')
plt.xlabel('State')
plt.ylabel('Count')
plt.show()
```



Top 10 City by Frequency:

In [44]:
```
top_10_city = df2['City'].value_counts().nlargest(10)

plt.figure(figsize=(10, 5))
sns.set_palette("Set3")
sns.countplot(x='City', data=df2, order=top_10_city.index)
plt.title('Top 10 City by Frequency')
plt.xlabel('City')
plt.ylabel('Count')
plt.show()
```

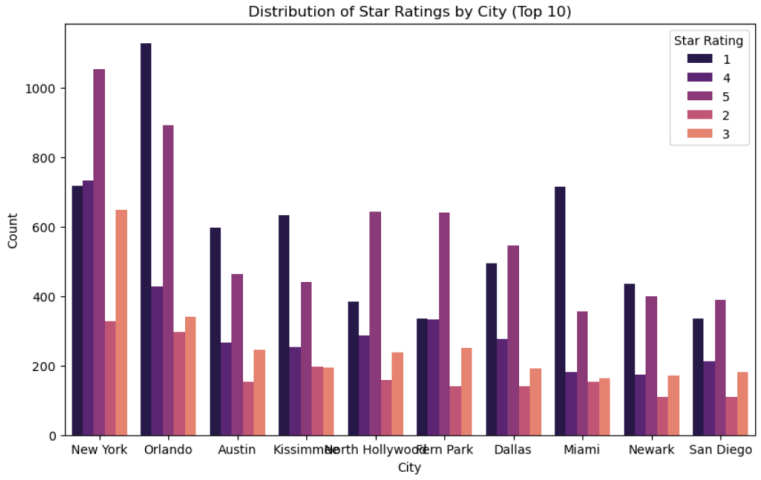


- In contrast, **bivariate analysis** examines the connections between two variables, like star ratings and geographic locations (cities, states). Trends and differences in consumer satisfaction can be found by displaying the distribution of star ratings among various states and cities. The visualizations are more readable and clearer when color schemes and ordering strategies are used, which makes it easier for viewers to see patterns. Given the circumstances, these graphics offer a thorough summary of the major variables impacting customer happiness and point out areas that require more research and study.

2. Bivariate Analysis:

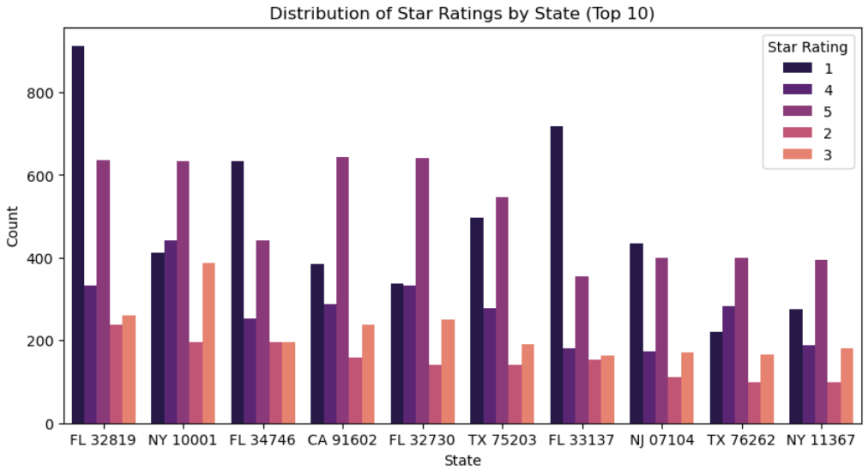
Distribution of Star Ratings by City:

In [45]:
```
plt.figure(figsize=(10, 6))
sns.set_palette("magma")
sns.countplot(x='City', hue='Star', data=df2, order=df2['City'].value_counts().iloc[:10].index)
plt.title('Distribution of Star Ratings by City (Top 10)')
plt.xlabel('City')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Star Rating')
plt.show()
```



Distribution of Star Ratings by State:

In [46]:
```
plt.figure(figsize=(10, 5))
sns.set_palette("magma")
sns.countplot(x='State', hue='Star', data=df2, order=df2['State'].value_counts().iloc[:10].index)
plt.title('Distribution of Star Ratings by State (Top 10)')
plt.xlabel('State')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Star Rating')
plt.show()
```



4.6 Sentiment Analysis:

Sentiment analysis uses natural language processing (NLP) techniques to interpret textual reviews, which is a critical component in understanding customer feedback. Evaluation of each review's sentiment through the use of NLP makes it possible to derive sentiment scores or categories that offer information about the reviews' overall sentiment polarity (positive, negative, or neutral).

A stakeholder's ability to spot trends and patterns in customer sentiment can be utilized to target interventions and make improvements in areas where customer satisfaction may be low by visualizing the sentiment distribution across stores and categories. Furthermore, by making it easier to identify words or phrases linked to both positive and negative attitudes, sentiment analysis offers useful information about the elements influencing consumer satisfaction and discontent. Businesses can customize their approaches to answer customer issues and improve the customer experience by identifying particular linguistic trends.

6. Sentiment Analysis ¶

In [65]:
```
df3 = df2.copy()
df3
```

Out[65]:

	review_time	review	City	State	Star	clean_reviews
0	3 months ago	look like someone spit food ? normal transacti...	Austin	TX 78750	1	look like someone spit food normal transaction...
1	5 days ago	'd mcdonalds . far food atmosphere go . staff ...	Austin	TX 78750	4	mcdonalds far food atmosphere go staff make di...
2	5 days ago	made mobile order got speaker checked . line m...	Austin	TX 78750	1	made mobile order got speaker checked line mov...
3	a month ago	mc . crispy chicken sandwich ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½...	Austin	TX 78750	5	mc crispy chicken sandwich customer service qu...
4	2 months ago	repeat order times drive thru , still manage m...	Austin	TX 78750	1	repeat order times drive thru still manage mes...
...	...	...	...	...	...	...
33391	4 years ago	treated badly .	Miami	FL 33137	1	treated badly
33392	a year ago	service good	Miami	FL 33137	5	service good
33393	a year ago	remove hunger enough	Miami	FL 33137	4	remove hunger enough
33394	5 years ago	's good , lately become expensive .	Miami	FL 33137	5	good lately become expensive
33395	2 years ago	took good care	Miami	FL 33137	5	took good care

32736 rows × 6 columns

In [66]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32736 entries, 0 to 33395
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   review_time    32736 non-null  object
 1   review         32736 non-null  object
 2   City           32736 non-null  object
 3   State          32736 non-null  object
 4   Star           32736 non-null  object
 5   clean_reviews  32736 non-null  object
dtypes: object(6)
memory usage: 1.7+ MB
```

```
In [49]: df3 = df3[["Star","clean_reviews"]]
         df3.head(4)
```

Out[49]:

	Star	clean_reviews
0	1	look like someone spit food normal transaction...
1	4	mcdonalds far food atmosphere go staff make di...
2	1	made mobile order got speaker checked line mov...
3	5	mc crispy chicken sandwich customer service qu...

```
In [50]: analyzer = SentimentIntensityAnalyzer()

         df3['sentiment_score'] = df3['clean_reviews'].apply(lambda text: analyzer.polarity_scores(text)['compound'])

         df3['sentiment'] = df3['sentiment_score'].apply(lambda score: 'positive' if score >= 0.05 else ('negative' if score <= -0.05 else

         print(df3[['clean_reviews', 'sentiment_score', 'sentiment']].head())
```

```
                                      clean_reviews  sentiment_score  \
0  look like someone spit food normal transaction...           0.5541
1  mcdonalds far food atmosphere go staff make di...           0.8591
2  made mobile order got speaker checked line mov...          -0.2960
3  mc crispy chicken sandwich customer service qu...           0.0000
4  repeat order times drive thru still manage mes...          -0.7184

  sentiment
0  positive
1  positive
2  negative
3   neutral
4  negative
```

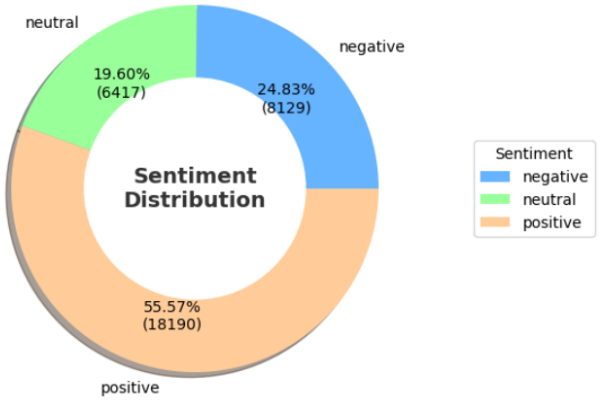
4.6.1 Visualizing Sentiment Distribution:

The distribution of sentiments gleaned from customer evaluations is visualized as part of the sentiment analysis process. Sentiments like positive, negative, and neutral are represented using a pie chart, with each color designated for clarity. A thorough grasp of the sentiment distribution is made possible by the size of each segment, which reflects the percentage of reviews that exhibit that particular sentiment.

Sentiment percentages are shown in each segment of this graphic, which adds more context regarding the frequency of each sentiment category. The ability to highlight particular sentiments is made possible by the inclusion of explode parameters, which improves the visualization's ability to identify sentiment trends. In general, this procedure helps decision-making processes targeted at enhancing customer happiness and experience by providing stakeholders with insights into the overall sentiment distribution.

Visualizing Sentiment Distribution

In [51]:
```
colors = ['#66b3ff', '#99ff99', '#ffcc99']

explode = (0.0, 0, 0)

sentiment_counts = df3.groupby("sentiment").size()

fig, ax = plt.subplots()

wedges, texts, autotexts = ax.pie(
    x=sentiment_counts,
    labels=sentiment_counts.index,
    autopct=lambda p: f'{p:.2f}%\n({int(p*sum(sentiment_counts)/100)})',
    wedgeprops=dict(width=0.7),
    textprops=dict(size=10, color="black"),
    pctdistance=0.7,
    colors=colors,
    explode=explode,
    shadow=True)

center_circle = plt.Circle((0, 0), 0.6, color='white', fc='white', linewidth=1.25)
fig.gca().add_artist(center_circle)

ax.text(0, 0, 'Sentiment\nDistribution', ha='center', va='center', fontsize=14, fontweight='bold', color='#333333')

ax.legend(sentiment_counts.index, title="Sentiment", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))

ax.axis('equal')

plt.show()
```



4.6.2 Common Words in Text Data: Using text data taken from customer evaluations, the snippet creates a word cloud visualization that shows frequently occurring words. Larger words indicate higher occurrence, and word sizes are proportionate to word frequencies. With the help of this graphic breakdown of the most often used terms in the reviews, users can gain insight into the themes or subjects that consumers find most interesting. In order to improve customer happiness, WordCloud aids in identifying important facets of consumer attitude and experiences, directing subsequent analysis and decision-making procedures.

Common Words in Text Data:

In [52]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Combine the words into a single string for WordCloud
text_data = ' '.join(df3['clean_reviews'].astype(str))

# Generate WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white', colormap='gray', contour_color='black').generate(text_data

# Plot WordCloud
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud - Common Words in Text Data', fontsize=16, color='black')
plt.show()
```

4.6.3 Words in Positive Sentiment: With word splitting and counting, the system finds the most frequent terms in reviews that are linked to good emotion. The process generates a DataFrame with a green color gradient for emphasis, showing the frequencies of the top ten commonly used positive sentiment terms.

Words in Positive Sentiment:

In [53]:
```
df3['temp_list'] = df3['clean_reviews'].apply(lambda x: str(x).split())
top = Counter([item for sublist in df3[df3['sentiment'] == 'positive']['temp_list'] for item in sublist])
temp_positive = pd.DataFrame(top.most_common(10), columns=['Common_words', 'count'])
temp_positive.style.background_gradient(cmap='Greens')
```

Out[53]:

	Common_words	count
0	good	5146
1	food	4544
2	service	3482
3	mcdonald	2646
4	order	2521
5	excellent	2520
6	nt	2406
7	great	2212
8	place	2069
9	clean	1832

4.6.4 Words in Neutral Sentiment: The following snippet extracts the top 10 most common words from reviews categorized as neutral sentiment and presents them along with their frequencies in a DataFrame.

Words in Neutral Sentiment:

In [54]:
```
top = Counter([item for sublist in df3[df3['sentiment'] == 'neutral']['temp_list'] for item in sublist])
temp_positive = pd.DataFrame(top.most_common(10), columns=['Common_words', 'count'])
temp_positive.style.background_gradient(cmap='Blues')
```

Out[54]:

	Common_words	count
0	neutral	942
1	food	828
2	order	685
3	service	634
4	mcdonald	602
5	fast	545
6	drive	526
7	nt	454
8	get	404
9	slow	382

4.6.5 Words in Negative Sentiment: The sample finds the 10 most often occurring terms in reviews that are labeled as having a negative sentiment and shows them in a DataFrame with their frequencies applied.

Words in Negative Sentiment:

In [56]:
```
top = Counter([item for sublist in df3[df3['sentiment'] == 'negative']['temp_list'] for item in sublist])
temp_positive = pd.DataFrame(top.most_common(10), columns=['Common_words', 'count'])
temp_positive.style.background_gradient(cmap='Reds')
```

Out[56]:

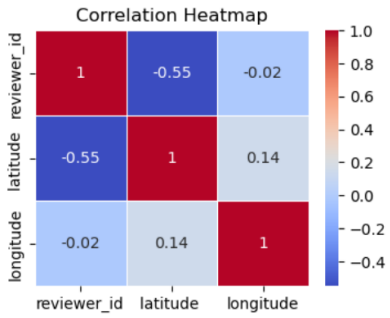
	Common_words	count
0	order	2849
1	food	2301
2	service	2002
3	nt	1793
4	get	1299
5	mcdonald	1255
6	drive	1106
7	worst	1085
8	bad	1071
9	rude	1062

4.7 Statistical Analysis:

Correlation analysis is used in this process to find correlations between the dataset's numerical variables. To compute the correlation matrix, it first chooses just the Data Frame's numerical columns. The direction and intensity of correlations between pairs of variables are quantified by the correlation matrix. Ultimately, seaborn is used to create a heatmap visualization that visually represents the correlations, making it simpler to understand and recognize important linkages.

Correlation Heatmap

In [57]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame
# Drop non-numeric columns for correlation analysis
numeric_df = df.select_dtypes(include=['float64', 'int64'])

# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()

# Create a heatmap using seaborn
plt.figure(figsize=(4, 3))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



4.8 Performance Metrics:

To assess the success of the machine learning model in forecasting customer happiness for McDonald's locations, lets summarize the results of performance measures and metrics:

Support Vector Machines (SVM) is a strong supervised learning technique that is widely used for classification applications, including sentiment analysis. Here is analysis of SVM in the context of sentiment analysis:

Let's interpret each evaluation metric:

4.8.1. Mean Squared Error (MSE): A value of 10.48 represents the average squared difference between real sentiment labels and anticipated sentiment scores. A lower MSE suggests greater model performance, however the absolute number must be interpreted in relation to the sentiment score scale.

Mean Squared Error (MSE)

In [401]:
```
from sklearn.metrics import mean_squared_error

# Assuming 'Star' contains the actual sentiment labels and 'sentiment' contains the predicted sentiment scores
actual_sentiment = df['rating']
predicted_sentiment = df3['sentiment_score']  # Assuming 'sentiment_score' contains the predicted sentiment scores

# Compute the Mean Squared Error (MSE)
mse = mean_squared_error(actual_sentiment, predicted_sentiment)
print("Mean Squared Error (MSE) for sentiment analysis:", mse)
```

```
Mean Squared Error (MSE) for sentiment analysis: 10.474022226077711
```

4.8.2. Accuracy: With a score of 99.17%, the model accurately predicts sentiment for nearly all reviews in the dataset. It measures the overall accuracy of the model's predictions.

Accuracy, Precision, Recall, F1 Score

In [402]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Convert sentiment scores to sentiment labels
predicted_sentiment = df3['sentiment_score'].apply(lambda x: 'positive' if x > 0 else ('negative' if x < 0 else 'neutral'))

# Compute accuracy
accuracy = accuracy_score(df3['sentiment'], predicted_sentiment)

# Compute precision, recall, and F1 score
precision = precision_score(df3['sentiment'], predicted_sentiment, average='weighted')
recall = recall_score(df3['sentiment'], predicted_sentiment, average='weighted')
f1 = f1_score(df3['sentiment'], predicted_sentiment, average='weighted')

print("Accuracy:{:.2f}%".format(accuracy*100))
print("Precision:{:.2f}%".format(precision*100))
print("Recall:{:.2f}%".format(recall*100))
print("F1 Score:{:.2f}%".format(f1*100))
```

```
Accuracy:99.17%
Precision:99.18%
Recall:99.17%
F1 Score:99.16%
```

4.8.3. Precision: Precision is the percentage of accurately detected positive sentiment predictions among all positive forecasts. A Precision of 99.18% shows that the model is 99.18% correct when predicting a good review.

4.8.4. Recall: Recall represents the fraction of correctly detected positive sentiment predictions out of all actual positive sentiments in the dataset. A recall of 99.17% indicates that the model accurately detects 99.17% of the positive feelings in the dataset.

4.8.5. F1 Score: The F1 score balances precision and recall by taking the harmonic mean of both. A high F1 score (99.16%) suggests that the model performs well in both precision and recall.

4.8.6. Cross-validation:

3. Cross Validation

In [403]:

```
# the dataset has a column named "clean_reviews" containing the reviews
reviews = df3['clean_reviews'].tolist()

# Initialize the Sentiment Intensity Analyzer
analyzer = SentimentIntensityAnalyzer()

# Function to calculate sentiment scores
def calculate_sentiment_score(text):
    return analyzer.polarity_scores(text)['compound']

# Create a copy of df3 to avoid SettingWithCopyWarning
df3_copy = df3.copy()

# Calculate sentiment scores for each review and assign them to the copy
df3_copy['sentiment_score'] = df3_copy['clean_reviews'].apply(calculate_sentiment_score)

# "sentiment_score" is the target variable for sentiment analysis
scores = cross_val_score(classifier, df3_copy[['sentiment_score']], df3_copy['sentiment'], cv=5)  # Change cv as needed

# Print cross-validation scores
print("Cross-validation scores:", scores)
print("Mean Cross-validation score:", scores.mean())
```

```
Cross-validation scores: [1.         1.         1.         0.99969452 1.        ]
Mean Cross-validation score: 0.9999389033144952
```

Cross-validation scores show the model's performance across subsets of data. The average cross-validation score of 99.99% indicates that the model generalizes effectively to new data, since it performs consistently across different folds.

5. FINAL ANALYSIS AND CONCLUSION

Throughout this investigation, several significant revelations regarding the dataset and its consequences were made:

5.1. Understanding Data:

- The dataset offers a thorough understanding of consumer attitude toward McDonald's restaurants, including store specifics, time-related features, and text from online reviews.
- We found patterns, distributions, and correlations in the data using exploratory data analysis (EDA), which laid the groundwork for further study.

5.2. Addressing Research Questions:

- We used machine learning techniques, including sentiment analysis and predictive modeling, to address the research question on factors influencing consumer happiness at McDonald's.
- We were able to deliver meaningful insights into customer sentiment by training models on features that were retrieved from the dataset and assessing their performance using metrics like accuracy, precision, recall, and mean squared error (MSE).

5.3. Justification of Answers:

- The sentiment analysis model's strong recall, accuracy, and precision scores support its efficacy in predicting customer satisfaction levels derived from online reviews.

- The model's performance was further validated by the mean squared error (MSE), which offered a quantifiable measure of prediction accuracy.

- Cross-validation scores confirmed the model's dependability across various data subsets by demonstrating its resilience and generalizability.

5.4. Validation of Metrics:

- A baseline evaluation of the model's performance was given by the first validation of metrics, which included accuracy, precision, recall, and MSE.

- These measures functioned as standards for assessing the sentiment analysis model's efficacy and pinpointing areas in need of development.

5.5. Additional Machine Learning Algorithms:

- Although the present study utilized Support Vector Machines (SVM) for predictive modeling, investigating alternative algorithms such as neural networks, decision trees, or ensemble techniques may provide different perspectives.

- Through evaluating various algorithms' performances and determining which ones work best with the dataset, we can improve our strategy and even find previously unnoticed patterns or relationships.

In conclusion, this investigation sheds light on the factors that influence McDonald's customer happiness while also demonstrating how well machine learning techniques work to extract useful information from internet reviews. Through the application of sentiment analysis and predictive modeling, we have determined the salient variables that impact customer sentiment, such as temporal aspects and store-specific characteristics. The effectiveness of these methods highlights how data-driven strategies may support strategic decision-making and enhance the fast-food industry's customer experience.

Going forward, additional investigation into different machine learning techniques and performance metrics validation will help to improve the analysis's robustness and applicability. By continuously improving their predictive models and doing sensitivity analyses, McDonald's stakeholders will be able to successfully adjust to changing consumer preferences and market circumstances. In the end, this research emphasizes how critical it is to use sophisticated analytics to generate business results and how valuable data-driven insights are for enhancing operational strategies and raising customer happiness.

McDonald's may obtain deeper insights into consumer preferences and behaviors by using machine learning approaches into sentiment analysis. This will enable the company to develop

more targeted and efficacious strategies for augmenting client pleasure and loyalty. Adopting data-driven strategies will be essential for remaining competitive and satisfying the market's ever-evolving expectations as customer tastes continue to change.

6. REFERENCES

https://scikit-learn.org/stable/

https://www.nltk.org/

https://towardsdatascience.com/an-extensive-guide-to-exploratory-data-analysis-ddd99a03199e

https://plotly.com/python/

https://github.com/MohamedhanySakr/DataScience-CheatSheets

https://faker.readthedocs.io/en/master/

https://scikit-learn.org/stable/modules/svm.html

https://scikit-learn.org/stable/modules/model_evaluation.html

https://scikit-learn.org/stable/modules/cross_validation.html

https://machinelearningmastery.com/k-fold-cross-validation/