Data Preperation:

1. Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error, mean_squared_log_error, explained_variance_score
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor, GradientBoostingClassifier, GradientBoostingRegressor
from sklearn.svm import SVC, SVR
from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor
from sklearn.metrics import classification_report
```

2. Load Dataset

```
# Load the dataset
data = pd.read_csv('/googleplaystore.csv')
data.head()
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw &	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8,	Varies with	4.2 and

Next steps: Generate code with `data` View recommended plots

```
data.dtypes
```

```
App               object
Category          object
Rating           float64
Reviews           object
Size              object
Installs          object
Type              object
Price             object
Content Rating    object
Genres            object
Last Updated      object
Current Ver       object
Android Ver       object
dtype: object
```

Data PreProcessing

```
# Drop missing values
data.dropna(inplace=True)

# Encode categorical variables (e.g., 'Category')
label_encoder = LabelEncoder()
data['Category'] = label_encoder.fit_transform(data['Category'])

#label_encoder = LabelEncoder()
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = label_encoder.fit_transform(data[column])
```

Feature Selection:

```
# Select features and target
X = data.drop('Rating', axis=1)  # Example: Predicting 'Rating'
y = data['Rating']
```

```
y
```

```
0        4.1
1        3.9
2        4.7
3        4.5
4        4.3
        ...
10834    4.0
10836    4.5
10837    5.0
10839    4.5
10840    4.5
Name: Rating, Length: 9360, dtype: float64
```

Train-Test Split:

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train
```

	App	Category	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
4438	5566	23	983	332	2	0	72	1	78	175	1935	12
10780	5185	11	1346	207	1	0	72	1	101	812	891	16
5401	254	20	2421	112	8	0	72	1	69	907	428	12
7536	6476	29	2328	412	18	0	72	1	105	187	1948	30
7738	8	6	1	112	7	1	4	1	34	916	1041	5
...	...	...	...	...	...	...	...	...	...	...	...	...
6144	869	25	1196	54	1	0	72	1	80	350	170	7
5516	5219	19	4742	412	6	0	72	1	68	192	2582	30
5728	2322	14	5881	345	6	0	72	4	0	736	2050	9
913	1572	9	5790	19	6	0	72	4	50	589	1981	16
8189	3934	11	5832	175	6	0	72	1	25	972	307	16

7488 rows × 12 columns

Next steps: Generate code with X_train View recommended plots

Data Scaling

```
# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Model Selection For Classification:

```
# Linear Regression Choosed
model=LinearRegression()
```

Model Evaluation for Classification:

```
model.fit(X_train, y_train)
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
mae = mean_absolute_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
msle = mean_squared_log_error(y_test, predictions)
explained_variance = explained_variance_score(y_test, predictions)

print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("R-squared:", r2)
print("Mean Squared Log Error (MSLE):", msle)
print("Explained Variance:", explained_variance)
```

```
Mean Squared Error (MSE): 0.2529108279389309
Mean Absolute Error (MAE): 0.348937073227601
R-squared: 0.02113193507423805
Mean Squared Log Error (MSLE): 0.01248827617790973
Explained Variance: 0.02125913569128901
```