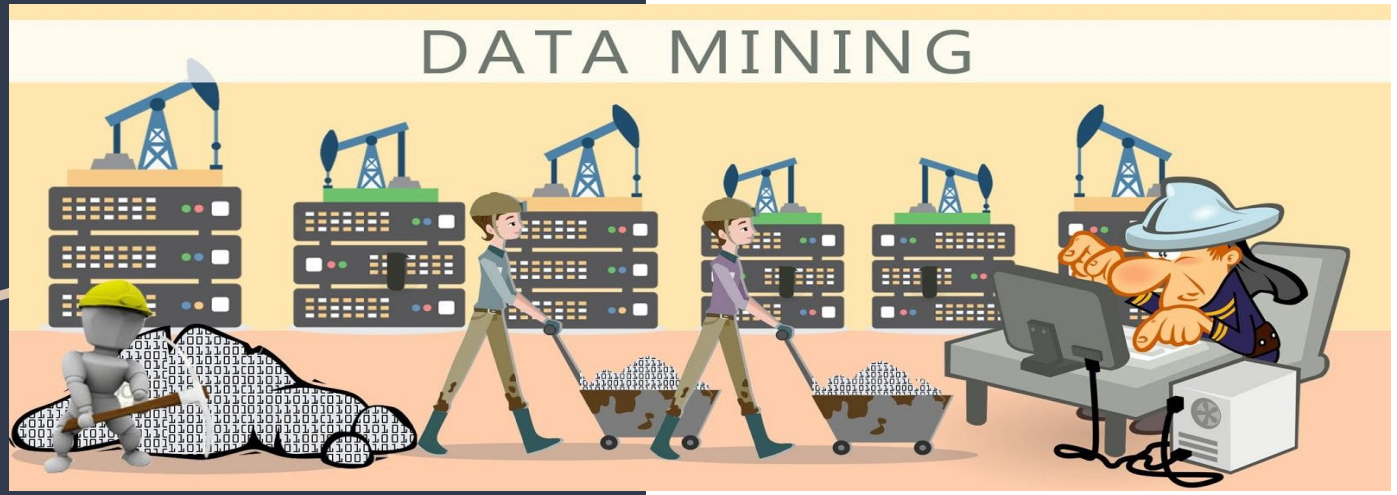
Can We Predict Google Play Store App Ratings

By: Randall Byrd, Riyazh Dholakia, Sujal Patel & Biprojoyti Paul

DATA MINING

- Finding Correlation and anomalies and or patterns.
- Finding meaningfulness
- Trying to make sense of it by finding meaningfulness.



PYTHON

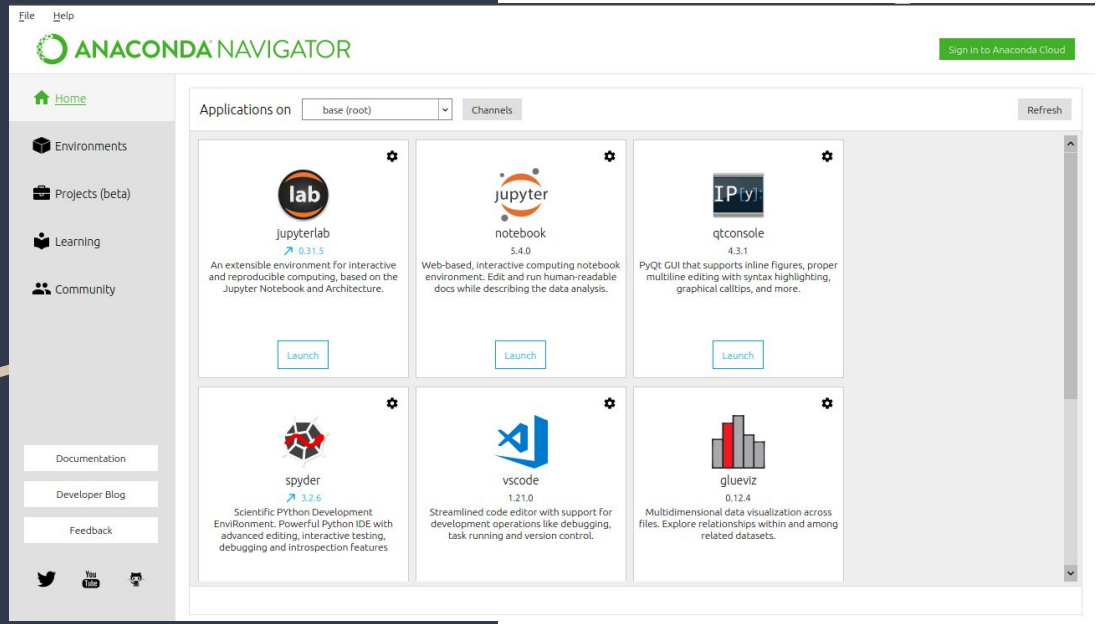

```
import numpy as np
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt
```

- Python now has 70,000 libraries
- Simplest programming language to pick up compared to other language
- Most popular data built in open source library is Panda

SOFTWARE

- Anaconda Navigator
- Jupyter Notebook
- Python 3

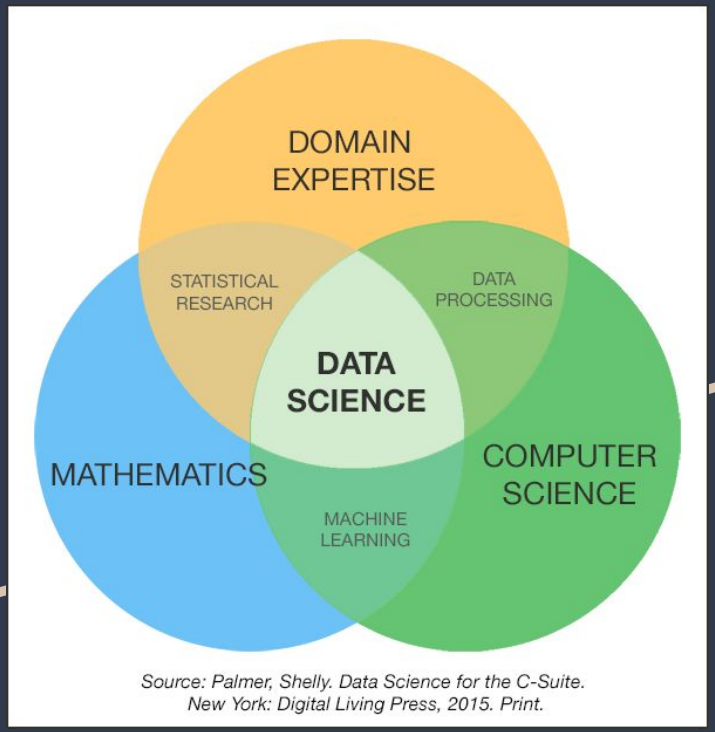


DATA

- Data.Gov
- Federal Reserve
- Kaggle
- Link for our data: https://www.kaggle.com/lava18/google-play-store-apps

	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 & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

DOMAIN KNOWLEDGE



- Deep Learning about that particular dataset
- We have to understand the situation of the data and the importance of it
- We use Domain Knowledge to predict the outcome.

ATTRIBUTES

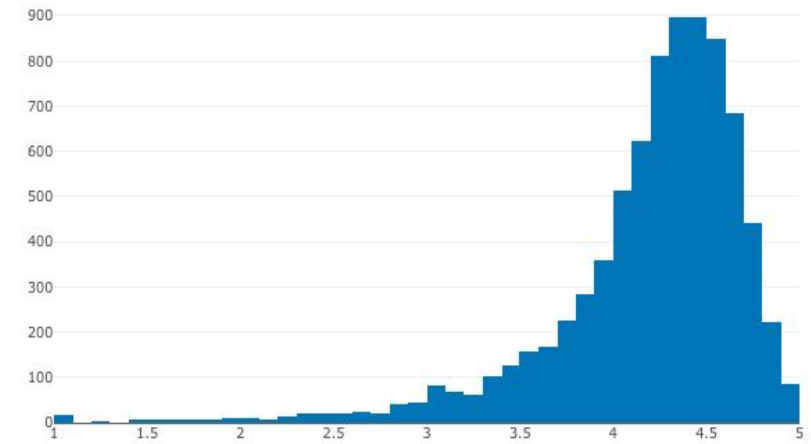
```
In [6]: # Checking the data type of the columns
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
App                10841 non-null object
Category           10841 non-null object
Rating             9367 non-null float64
Reviews            10841 non-null object
Size               10841 non-null object
Installs           10841 non-null object
Type               10840 non-null object
Price              10841 non-null object
Content Rating     10840 non-null object
Genres             10841 non-null object
Last Updated       10841 non-null object
Current Ver        10833 non-null object
Android Ver        10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

- Qualitative (Nominal (N), Ordinal (O), Binary(B))
- Quantitative (Discrete, Continuous)

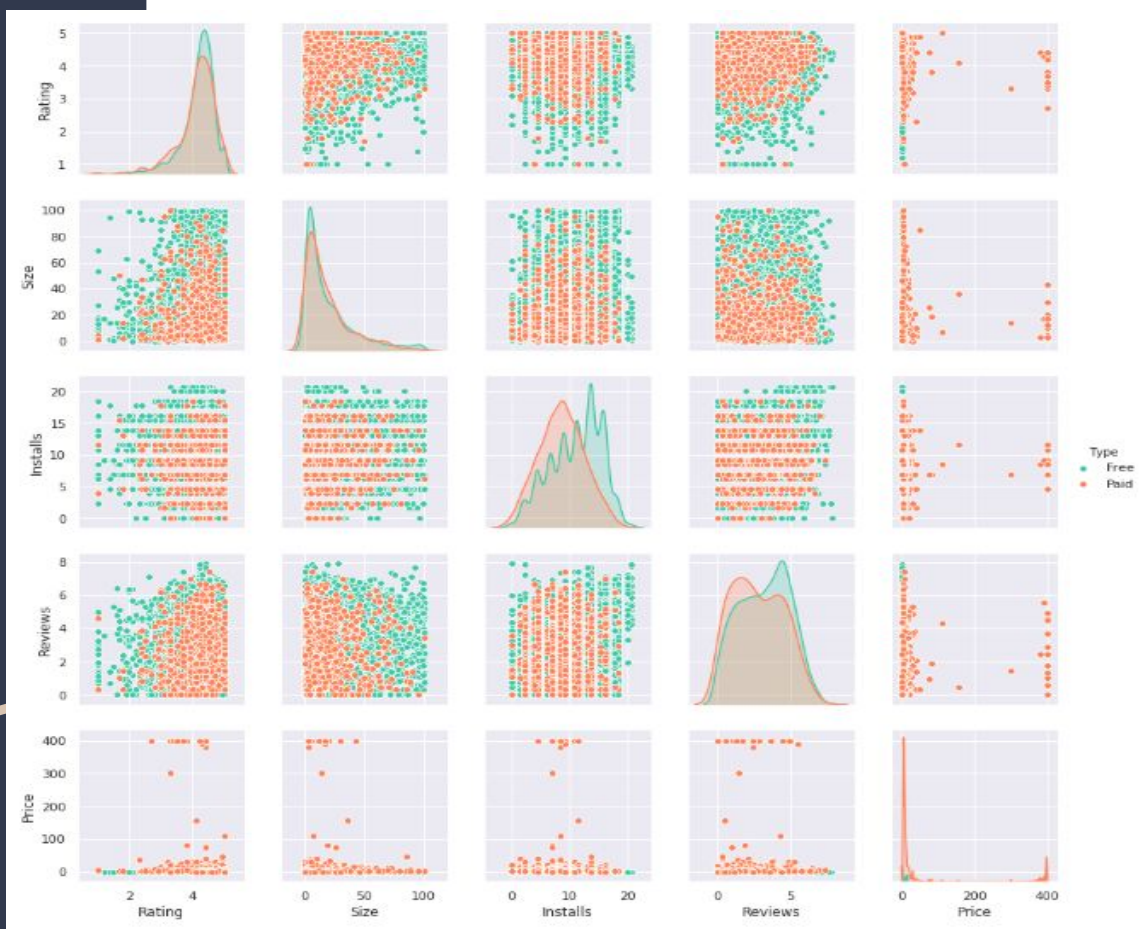
DESCRIPTIVE STATISTICS

```
Average app rating =  4.173243045387998
```



- Representation of large dataset
- Mean
- Mode
- Median
- Standard Deviation
- Plots

LEARNING DATA

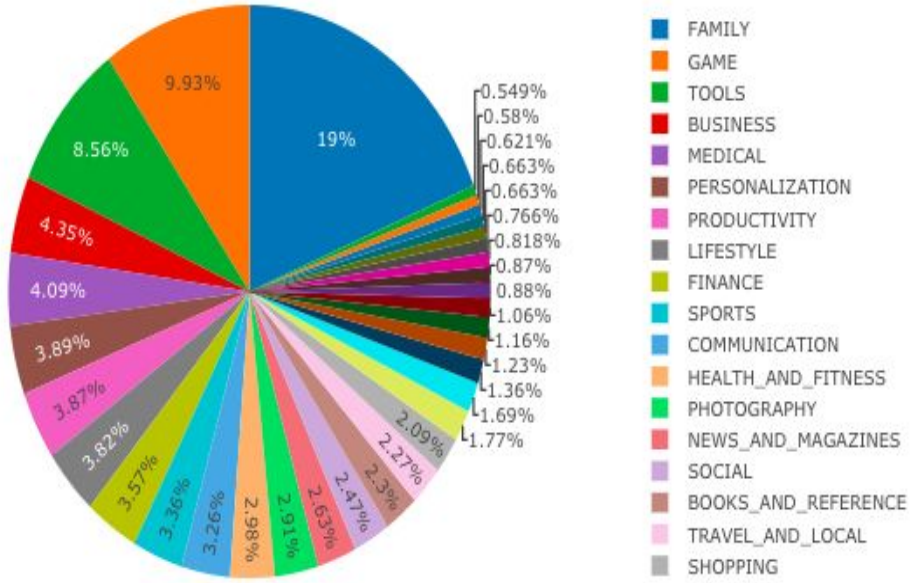


This is the basic exploratory analysis to look for any evident patterns or relationships bet ween the features.

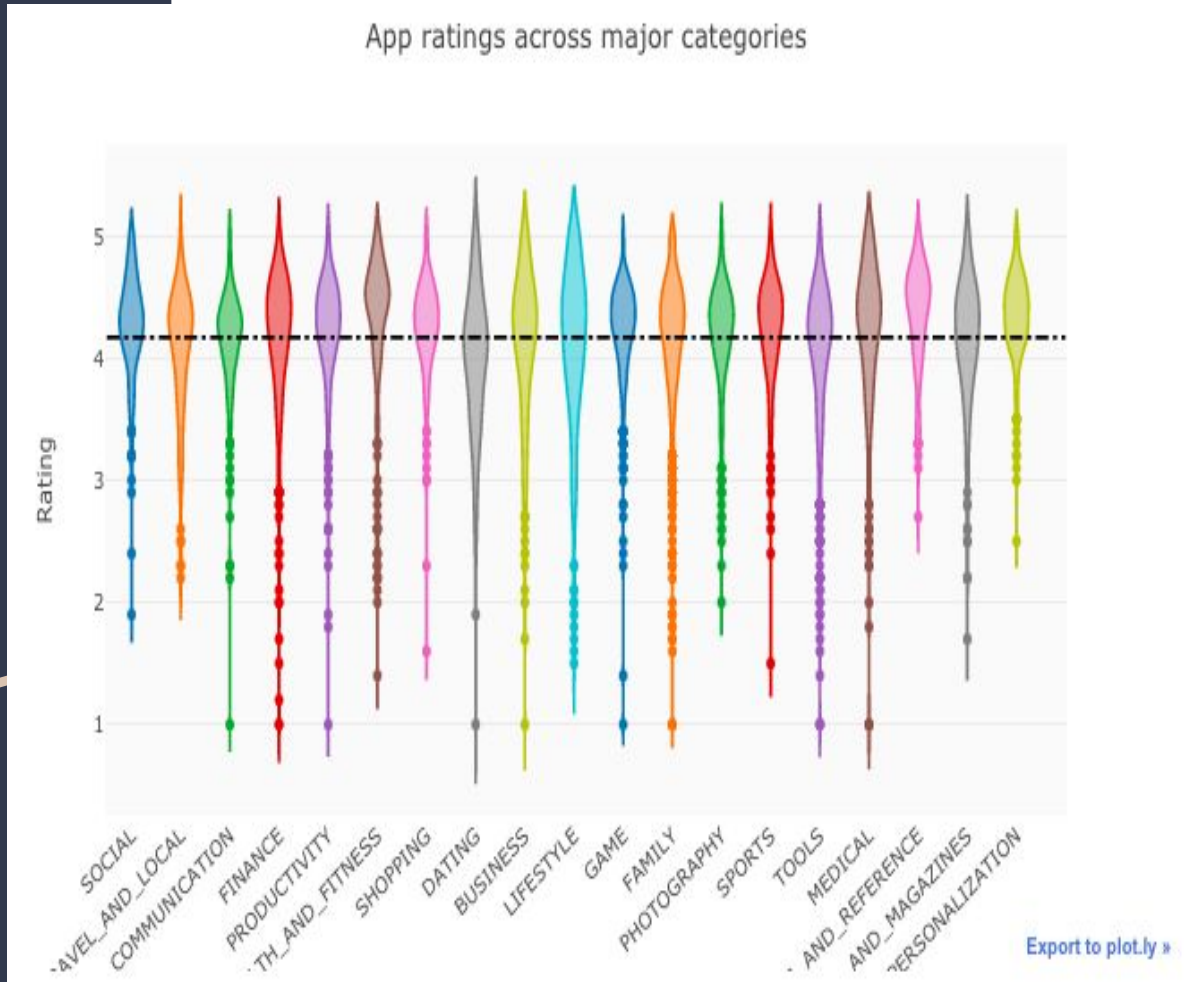
LEARNING DATA

Android market breakdown

Which category has the highest share of (active) apps in the market?



LEARNING DATA



LEARNING DATA

LEARNING DATA

PAID APPS

Negative words: malware, problem

Positive words: great, love, easy

WORDCLOUD - A quick look on reviews

FREE APPS

Negative words: ads, bad, hate

Positive words: good, love, best, great

PRE-PROCESSING

- Missing Values
- Outliers/Anomalies
- Data accuracy, believability and interpretability

```
In [8]: #Looks like there are missing values in "Rating", "Type",
        #"Content Rating" and " Android Ver". But most of these missing values in Rating column.

In [9]: df.isnull().sum()

Out[9]: App                0
        Category           0
        Rating          1474
        Reviews            0
        Size               0
        Installs           0
        Type               1
        Price              0
        Content Rating     1
        Genres             0
        Last Updated       0
        Current Ver        8
        Android Ver        3
        dtype: int64
```

PRE-PROCESSING Steps

- Removing records or filled in missing values with a mean, median or mode

In [10]:
```
#There are two strategies to handle missing data, either removing records with these missing values or replacing
#missing values with a specific value like (mean, median or mode) value of the column

# The best way to fill missing values might be using the median instead of mean.
df['Rating'] = df['Rating'].fillna(df['Rating'].median())

# Before filling null values we have to clean all non numerical values & unicode charachters
replaces = [u'\u00AE', u'\u2013', u'\u00C3', u'\u00E3', u'\u00B3', '[', ']', "'"]
for i in replaces:
    df['Current Ver'] = df['Current Ver'].astype(str).apply(lambda x : x.replace(i, ''))

regex = [r'[-+|/:/;(_)@]', r'\s+', r'[A-Za-z]+']
for j in regex:
    df['Current Ver'] = df['Current Ver'].astype(str).apply(lambda x : re.sub(j, '0', x))

df['Current Ver'] = df['Current Ver'].astype(str).apply(lambda x : x.replace('.', ',',1).replace('.', '').replace(',',
df['Current Ver'] = df['Current Ver'].fillna(df['Current Ver'].median())
```

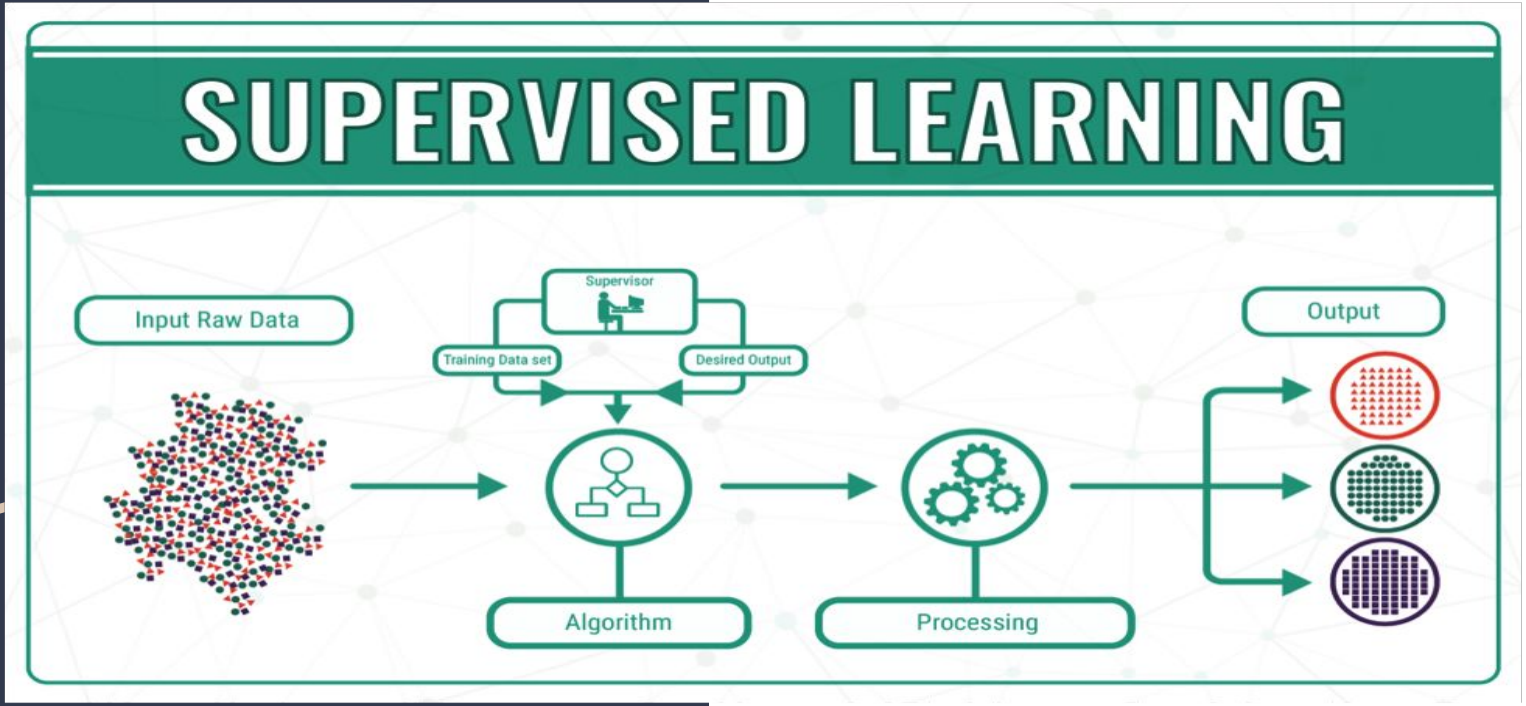
PRE-PROCESSING Steps

- Conversion of value type metrics

```
In [26]: # Convert kbytes to Mbytes
         k_indices = df['Size'].loc[df['Size'].str.contains('k')].index.tolist()
         converter = pd.DataFrame(df.loc[k_indices, 'Size'].apply(lambda x: x.strip('k')).astype(float).apply(lambda x: x / 102
         df.loc[k_indices,'Size'] = converter
```

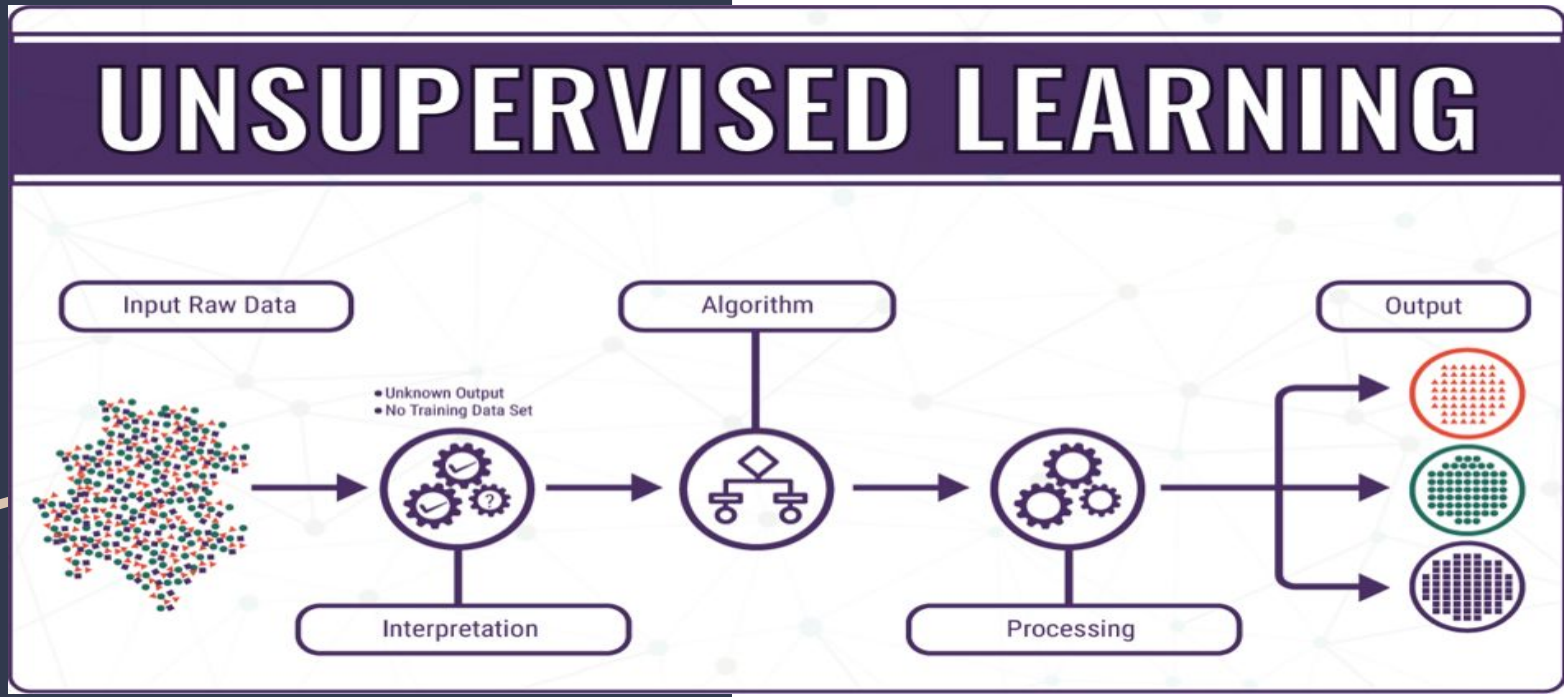
SUPERVISED LEARNING

- Known input set to do regression or classification model
- Example: KNN



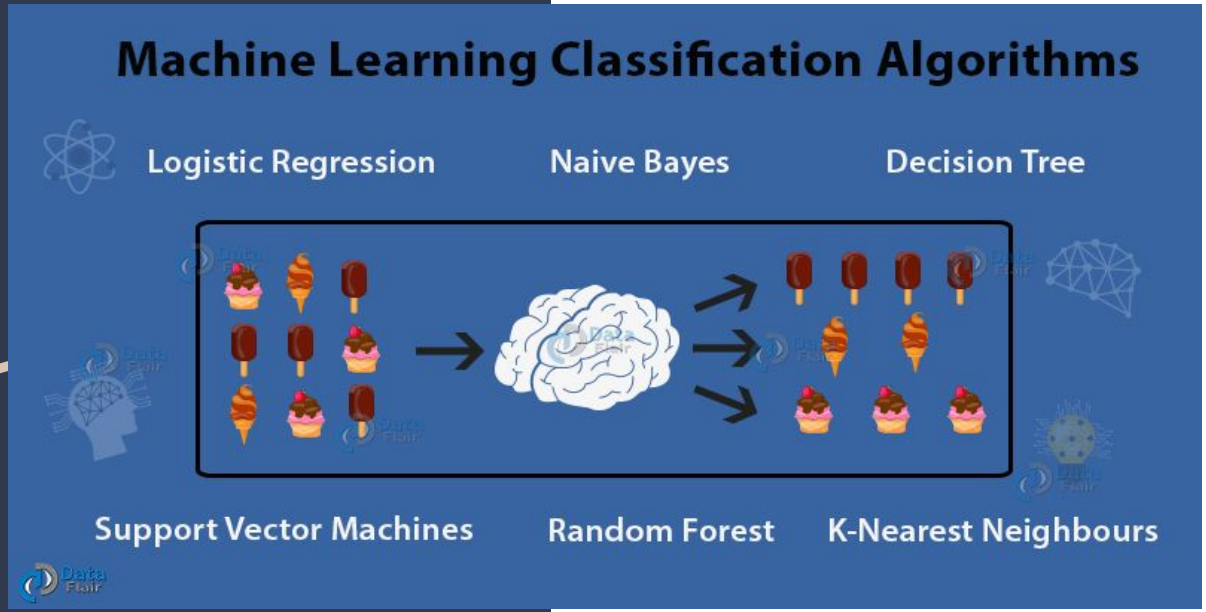
UNSUPERVISED LEARNING

- Does not require labelled data to train a model.
- Example: K-Means Clustering



ALGORITHMS

- KNN
- Random Foresting
- K-Means



TRAINING & TEST DATA SPLIT

```
# Split data into training and testing sets
features = ['App', 'Reviews', 'Size', 'Installs', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver']
features.extend(category_list)
X = df[features]
y = df['Rating']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 10)
```

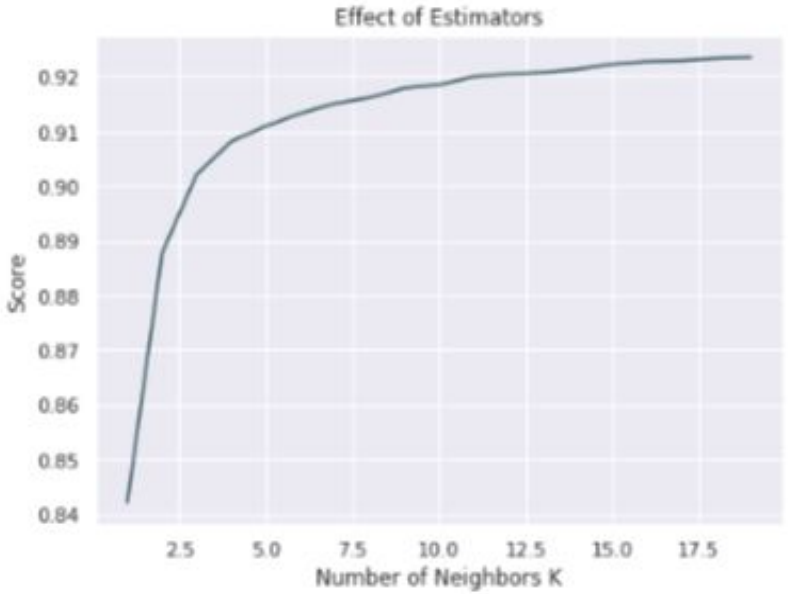
KNN

```
# Look at the 15 closest neighbors
model = KNeighborsRegressor(n_neighbors=15)
```

```
# Find the mean accuracy of knn regression using X_test and y_test
model.fit(X_train, y_train)
```

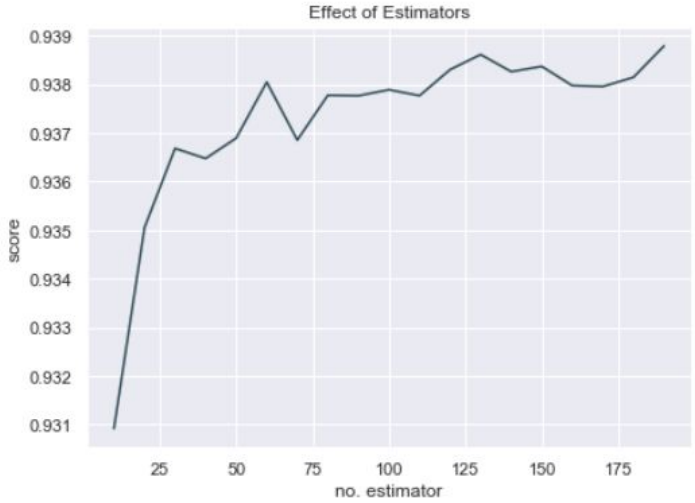
In [52]:
```
# Calculate the mean accuracy of the KNN model
accuracy = model.score(X_test,y_test)
'Accuracy: ' + str(np.round(accuracy*100, 2)) +
```

Out[52]: 'Accuracy: 92.22%'



Random Foresting

```
model = RandomForestRegressor(n_jobs=-1)
# Try different numbers of n_estimators - this will take a minute or so
estimators = np.arange(10, 200, 10)
scores = []
for n in estimators:
    model.set_params(n_estimators=n)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
plt.figure(figsize=(7, 5))
plt.title("Effect of Estimators")
plt.xlabel("no. estimator")
plt.ylabel("score")
plt.plot(estimators, scores)
results = list(zip(estimators,scores))
results
```




```
predictions = model.predict(X_test)
'Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions)
```

```
6]: ('Mean Absolute Error:', 0.24074655272868536)
```

```
'Mean Squared Error:', metrics.mean_squared_error(y_test, predictions)
```

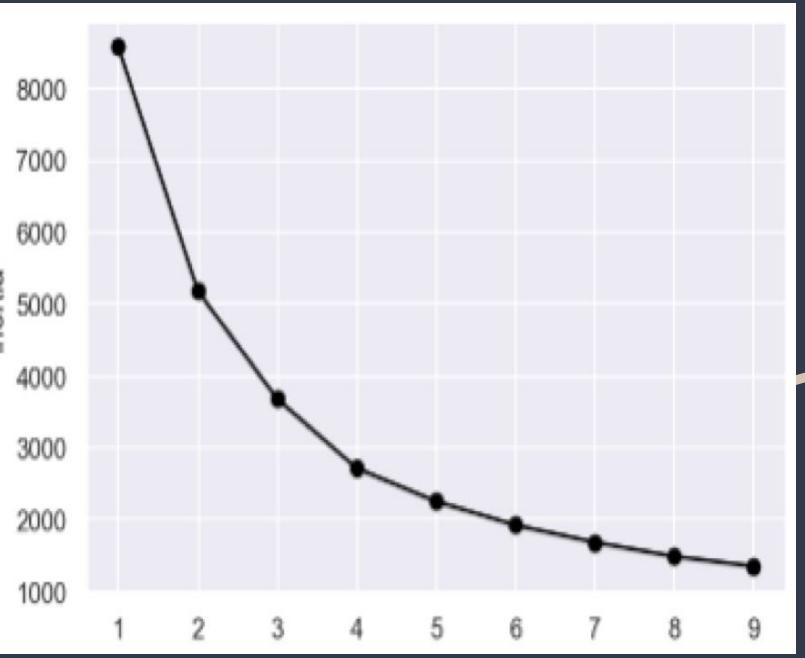
```
7]: ('Mean Squared Error:', 0.15970111477956886)
```

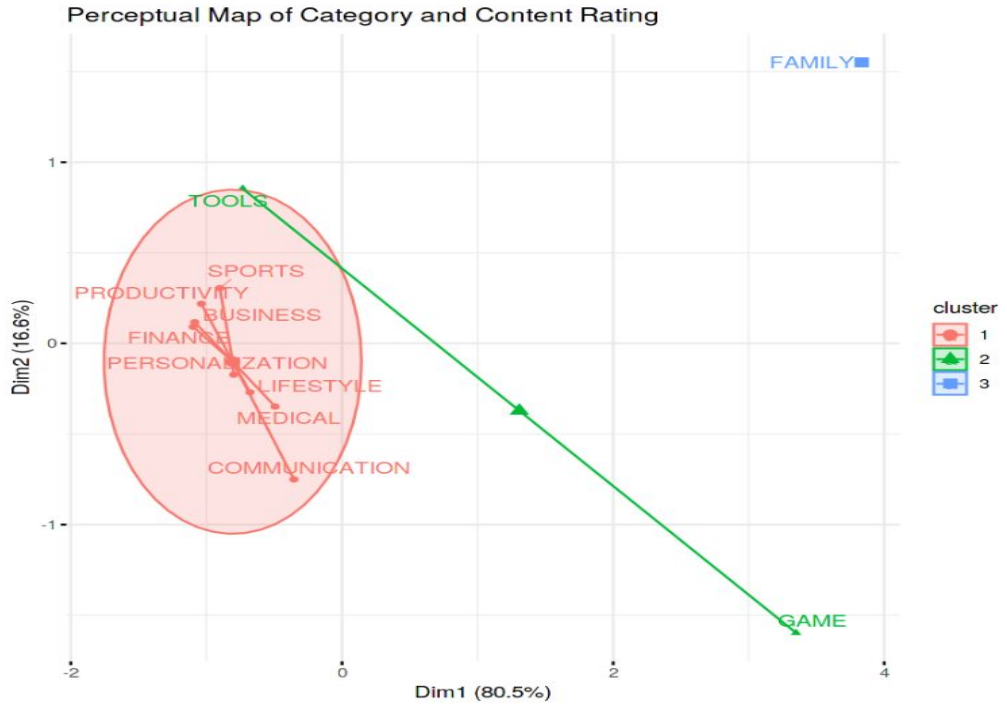
```
'Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions))
```

```
8]: ('Root Mean Squared Error:', 0.399626218834011)
```

K-Means

```
# Cluster Analysis
set.seed(123)
km.res <- kmeans(contingency_table, 3, 25)
fviz_cluster(km.res, data = contingency_table, palete=c("#2E9FDF","#00AFBB"),
             ellipse.type="euclid", star.plot=TRUE, repel = TRUE, ggtheme = theme_minimal(),
             main="Perceptual Map of Category and Content Rating")
```





CONCLUSION

RESOURCES

- Medium
- Kaggle
- YouTube
- GitHub
- https://github.com/riyazhdholakia/PredictGooglePlayStoreAppRatings
- https://medium.com/quickknowledge