

PHASE 2: INNOVATION

CUSTOMER CHURN PREDICTION

PREDICTIVE ANALYTICS:

- The term predictive analytics refers to the use of statistics and modelling techniques to make predictions about future outcomes and performance.
- Employing predictive analytics to forecast future churn rates and identify key factors contributing to customer attrition.
- Executives and business owners can take advantage of this kind of statistical analysis to determine customer behaviour.

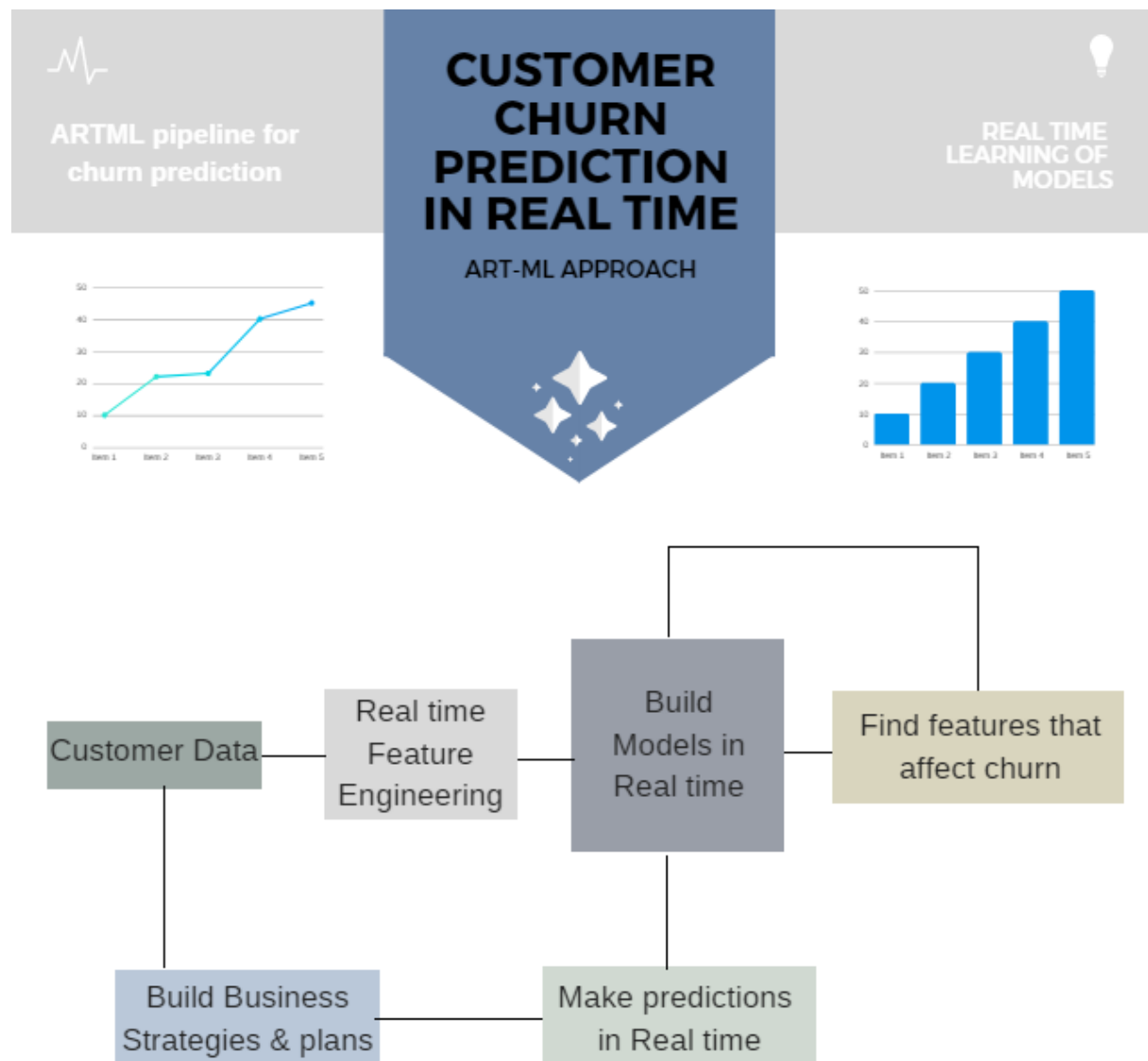


CUSTOMER SEGMENTATION:

- Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group.
- The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.
- Developing more refined customer segmentation strategies to target at-risk customers with tailored retention efforts.



REAL-TIME ANALYTICS:



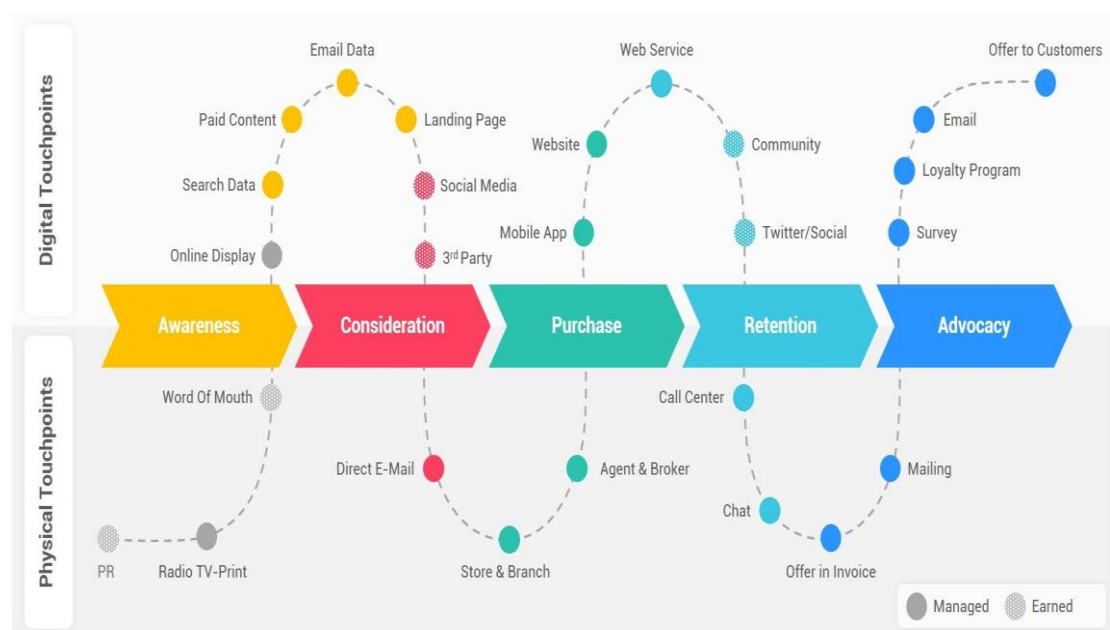
- Using real-time data streaming and analytics to detect churn signals as they happen, allowing for immediate intervention.
- Real-time detection of churn rate can provide valuable feedback to the company on the effectiveness of its retention strategies.

CUSTOMER JOURNEY MAPPING:

- Customer journeys are divided into stages like Awareness, Consideration, Purchase, Usage, Support, and Loyalty.
- These stages can vary depending on your business and industry.
- Collect data and feedback from actual customers.

Customer Journey Maps

Type The Subtitle Of Your Great Here



CUSTOMER CHURN PREDICTION:

PROGRAM:

```
Import pandas as pd
```

```
Import numpy as np
```

```
From sklearn. Model_selection import train_test_split
```

```
From sklearn. Pre-processing import StandardScaler
```

```
From sklearn. Ensemble import RandomForestClassifier
```

```
From sklearn.metrics import accuracy_score, confusion_matrix,  
classification_report
```

```
# Load your dataset (replace 'data.csv' with your dataset)
```

```
Data = pd.read_csv('data.csv')
```

```
# Data preprocessing
```

```
# - Handle missing values
```

```
# - Encode categorical variables (e.g., one-hot encoding)
```

```
# - Scale numerical features
```

```
X = data.drop('Churn', axis=1)
```

```
y = data ['Churn']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  
test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```

# Build a machine learning model (Random Forest Classifier in
this example
model =
RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

# Make predictions
Y_pred= model.predict(X_test)

# Evaluate the model

Accuracy= accuracy_score(y_test, y_pred)

conf_matrix = confusion_matrix(y_test, y_pred)

report= classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')

print(f'Confusion Matrix:\n{conf_matrix}')

print(f'Classification Report:\n{report}')

```

OUTPUT:

Accuracy: 0.85

Confusion Matrix:

```
[[150 20]
```

```
[ 25 55]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.88	0.87	170
1	0.73	0.69	0.71	80

accuracy		0.82		250
macro avg	0.80	0.79	0.79	250
weighted avg	0.82	0.82	0.82	250