Telco 고객 데이터 대시보드 만들기

권효은 | 김예슬

Table of Contents

TOPIC SELECTION

TELCO Customer Data Dashboard

- 고객 데이터
- 유의미한 행동/특성 관측 가능

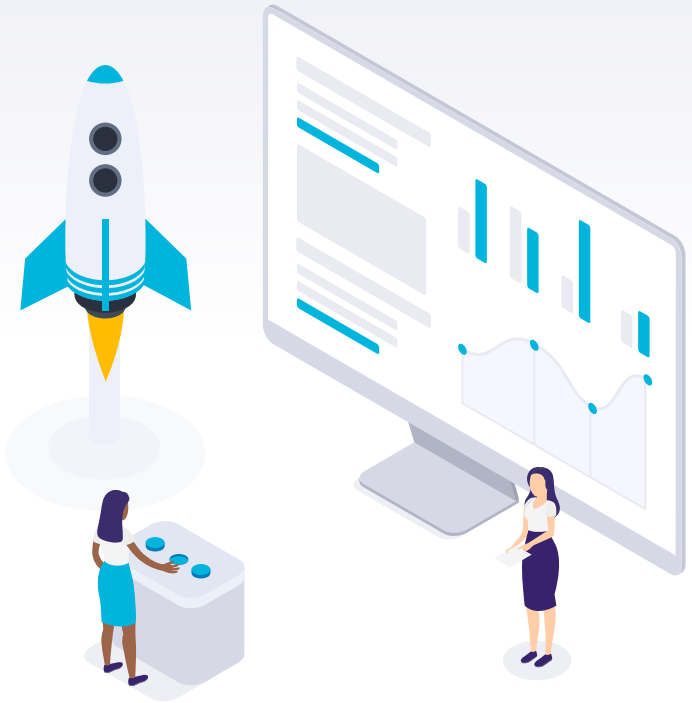
2

Page Design

고객 데이터를 이용해 한 눈에
들어오는 페이지 구성하기

대시보드를 가장 자주, 유용하게
쓸 청자 설정하기:

마케팅, 회계, 경영기획



3

DATA COMPOSITION

1. 데이터 대분류
2. 데이터 소분류
3. 전처리

데이터 대분류 및 소분류

Phone

휴대폰 통신과 관련된 서비스

+

Internet

인터넷과 관련된 서비스

- 온라인 서비스
- 스트리밍 서비스

→

TOTAL

- 비율
- 과금액
- 이용기간
- 납부방식
- 스트리밍 서비스

```
data = pd.read_csv('C:/Users/NTX550/Desktop/ASAC/web/WA_Fn-UseC_-Telco-Customer-Churn.csv')
data_df = pd.DataFrame(data)
data_df['InternetService'] = data_df['InternetService'].apply(lambda x: 'Yes' if x == 'DSL' or x == 'Fiber optic' else 'No')
data_df['OnlineSecurity'] = data_df['OnlineSecurity'].apply(lambda x: 'No' if x == 'No internet service' or x == 'No' else 'Yes')
data_df['OnlineBackup'] = data_df['OnlineBackup'].apply(lambda x: 'No' if x == 'No internet service' or x == 'No' else 'Yes')

data_df['Internet'] = int(0)
for i in range(len(data_df)):
    if data_df['InternetService'][i] == 'Yes' or data_df['OnlineSecurity'][i] == 'Yes' or data_df['OnlineBackup'][i] == 'Yes':
        data_df['Internet'][i] = 'Yes'
    else:
        data_df['Internet'][i] = 'No'


data_df['StreamingTV'] = data_df['StreamingTV'].apply(lambda x: 'No' if x == 'No internet service' or x == 'No' else 'Yes')
data_df['StreamingMovies'] = data_df['StreamingMovies'].apply(lambda x: 'No' if x == 'No internet service' or x == 'No' else 'Yes')

data_df['Streaming'] = 0
for i in range(len(data_df)):
    if data_df['StreamingTV'][i] == 'Yes' or data_df['StreamingMovies'][i] == 'Yes':
        data_df['Streaming'][i] = 'Yes'
    else:
        data_df['Streaming'][i] = 'No'


data_df['TotalCharges'] = data_df['TotalCharges'].apply(lambda x: 0 if x == ' ' else x)
data_df['TotalCharges'] = data_df['TotalCharges'].astype(float)
data_df['MultipleLines'] = data_df['MultipleLines'].apply(lambda x: 'No' if x == 'No phone service' or x == 'No' else 'Yes')
```

4

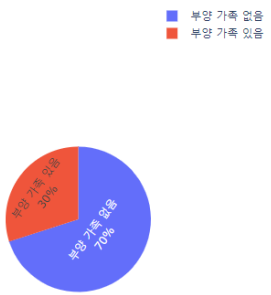
BUILDING PAGE

1. 결과
2. 페이지 개설 과정

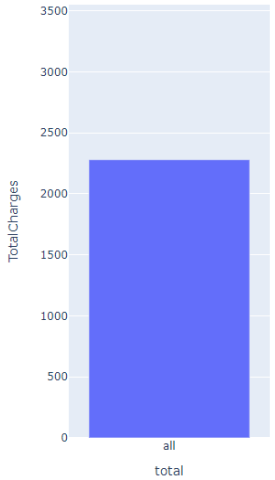
Total Customer	New Customer	Churn Customer	Duration
7043	624	1869	32.4

Total
Phone
Internet

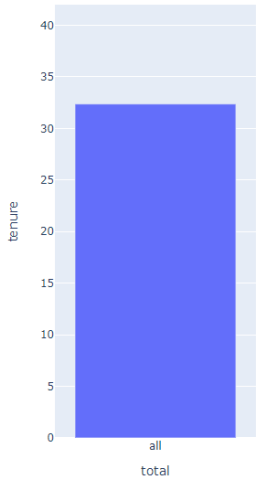
부양 가족 여부



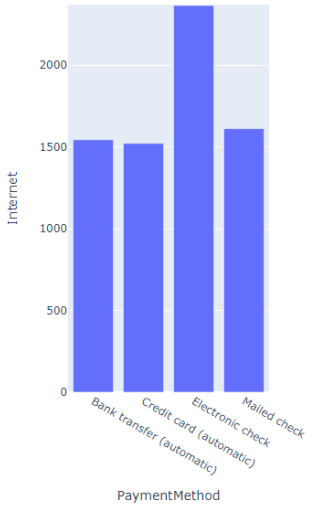
총 고객 평균 과금액



총 고객 평균 이용 기간(개월)



총 고객 납부 방식



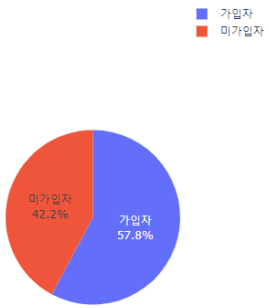
Total Customer	New Customer	Churn Customer	Duration
7043	624	1869	32.4

Total

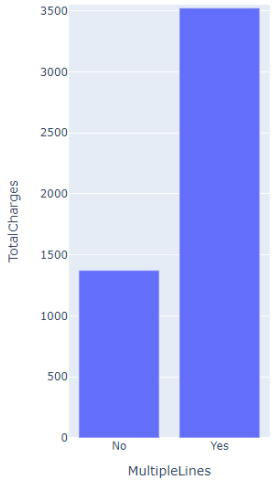
Phone

Internet

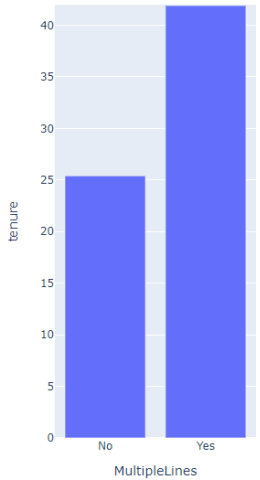
MultiLine 가입 여부



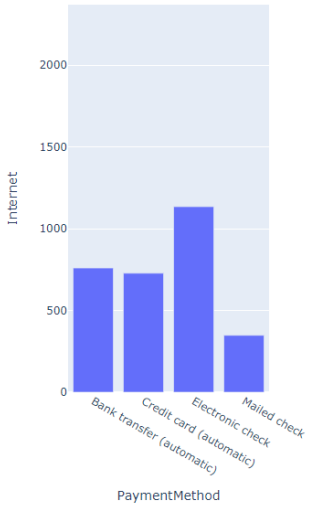
MultiLine 가입 여부별 평균 과금액



MultiLine 가입 여부별 평균 이용 기간(개월)



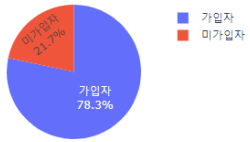
MultiLine 가입자 고객 납부 방식



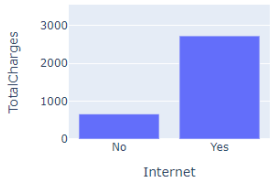
Total Customer	New Customer	Churn Customer	Duration
7043	624	1869	32.4

- Total
- Phone
- Internet

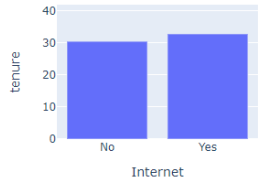
온라인 서비스 가입 여부



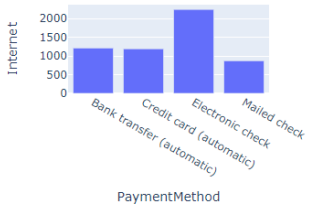
온라인 서비스 가입 여부별 평균 과금액



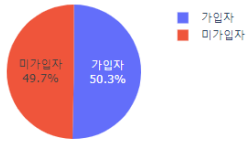
온라인 서비스 가입 여부별 평균 이용 기간(개월



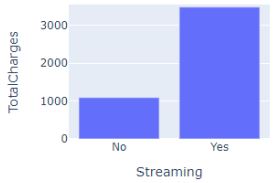
온라인 서비스 가입자 고객 납부 방식



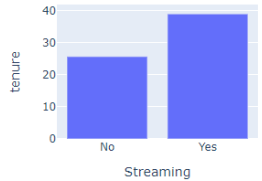
스트리밍 서비스 가입 여부



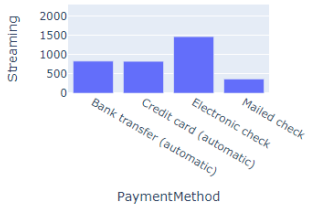
스트리밍 서비스 가입 여부별 평균 과금액



스트리밍 서비스 가입 여부별 평균 이용 기간(기



스트리밍 서비스 가입자 고객 납부 방식



```python
@app.route('/totalgraph')
def totalgraph():
    fig1 = px.pie(data_df, values=data_df['Dependents'].value_counts(), names=['부양 가족 없음', '부양 가족 있음'], title='부양 가족 여부')
    fig1.update_traces(textposition="inside", textinfo="percent+label")
    totalpieJSON = json.dumps(fig1, cls=plotly.utils.PlotlyJSONEncoder)

    charges_df = data_df.groupby(['total']).mean().reset_index()
    fig2 = px.bar(charges_df, x='total', y='TotalCharges', title='총 고객 평균 과금액')
    fig2.update_layout(yaxis_range=[0, 3550])
    totalchargesJSON = json.dumps(fig2, cls=plotly.utils.PlotlyJSONEncoder)

    fig3 = px.bar(charges_df, x='total', y='tenure', title='총 고객 평균 이용 기간(개월)')
    fig3.update_layout(yaxis_range=[0, 42])
    totaltenureJSON = json.dumps(fig3, cls=plotly.utils.PlotlyJSONEncoder)

    df = data_df.groupby(['PaymentMethod']).count().reset_index()
    fig4 = px.bar(df, x='PaymentMethod', y='Internet', title='총 고객 납부 방식')
    fig4.update_layout(yaxis_range=[0, 2370])
    totalmethodJSON = json.dumps(fig4, cls=plotly.utils.PlotlyJSONEncoder)

    result = {"name" : "total",
              "data": {"pie": totalpieJSON, "charges" : totalchargesJSON,
              "tenure": totaltenureJSON, "method" : totalmethodJSON}}
    result = json.dumps(result)
    return result
```

```
function drawCustomerInfo(xhr) {
    var data = xhr.responseText
    var json = JSON.parse(data)
    totalcus.innerText = json['totalcus']
    newcus.innerText = json['newcus']
    churncus.innerText = json['churncus']
    avgdur.innerText = json['avgdur']
}
```

```
<!-- values -->
<div id="top" class="row g-3">
    <div class="col-sm-3">
        <div class="bg-dark text-white rounded p-3">
            <div><p class="fs-5">Total Customer</p></div>
            <div ><p id="totalcus" class="fs-1">Loading</p></div>
        </div>
    </div>
    <div class="col-sm-3">
        <div class="bg-dark text-white rounded p-3">
            <div><p class="fs-5">New Customer</p></div>
            <div><p id="newcus" class="fs-1">Loading</p></div>
        </div>
    </div>
    <div class="col-sm-3">
        <div class="bg-dark text-white rounded p-3">
            <div><p class="fs-5">Churn Customer</p></div>
            <div><p id="churncus" class="fs-1">Loading</p></div>
        </div>
    </div>
    <div class="col-sm-3">
        <div class="bg-dark text-white rounded p-3">
            <div><p class="fs-5">Duration</p></div>
            <div><p id="avgdur" class="fs-1">Loading</p></div>
        </div>
    </div>
</div>
```

```
</script>
<body class="bg-secondary" onload="loadAllInfo()">
    <div class="container-fluid rounded py-5">
```

```
function loadAllInfo(){
    loadCustomerInfo()
    activate('/totalgraph')
}
```

```
<div class="row">
    <input id="section-total" type="radio" name="section" value="total"
    style="appearance:none" checked>
    <label for="section-total" class="pt-3 pl-3 pb-3" onclick="activate
    ('/totalgraph')">
        <span>Total</span>
    </label>
</div>
<div class="row">
    <input id="section-phone" type="radio" name="section" value="phone"
    style="appearance:none" >
    <label for="section-phone"  class="pt-3 pl-3 pb-3" onclick="activate
    ('/phonegraph')">
        <span>Phone</span>
    </label>
</div>
<div class="row">
    <input id="section-internet" type="radio" name="section"
    value="internet" style="appearance:none">
    <label for="section-internet" class="pt-3 pl-3 pb-3"
    onclick="activate('/internetgraph')">
        <span>Internet</span>
    </label>
</div>
```

```
<style>
    input[type="radio"] + label{
        width:100%;
    }

    input[type="radio"]:checked + label{
        background-color: white;
        color: black;
    }
</style>
```

```
function drawInternet(data){
    var first_row = document.getElementById('first-graph-row')
    var second_row = document.getElementById('second-graph-row')
    first_row.className = first_row.className.replace('h-100', 'h-50')
    second_row.style.display = ''
```

```
function drawPhone(data){
    var first_row = document.getElementById('first-graph-row')
    var second_row = document.getElementById('second-graph-row')
    first_row.className = first_row.className.replace('h-50', 'h-100')
    second_row.style.display = 'none'
```

THANKS!

Any questions?