Dropout Prediction EDA

In [1]:
```
import pandas as pd
import numpy as np
from hashlib import sha1
import matplotlib.pyplot as plt
import seaborn as sns
import altair as alt
```

General Preprocessing

In [2]:
```
alt.renderers.enable('mimetype')

data_path = '../data/raw/data.csv'
sep = ';'
```

In [3]:
```
df = pd.read_csv(data_path, sep=sep)
df = df.rename(columns={'Nacionality': 'Nationality', 'Daytime/evening attendance\t': 'Daytime_e
df.shape
```

Out[3]: (4424, 37)

In [4]:
```
df.head()
```

Out[4]:

	Marital status	Application mode	Application order	Course	Daytime_evening_attendance	Previous qualification	Previous qualification (grade)	Nationality
0	1	17	5	171	1	1	122.0	1
1	1	15	1	9254	1	1	160.0	1
2	1	1	5	9070	1	1	122.0	1
3	1	17	2	9773	1	1	122.0	1
4	2	39	1	8014	0	1	100.0	1

5 rows × 37 columns

In [5]:
```
df.tail()
```

Out[5]:

	Marital status	Application mode	Application order	Course	Daytime_evening_attendance	Previous qualification	Previous qualification (grade)	Nationa
4419	1	1	6	9773	1	1	125.0	
4420	1	1	2	9773	1	1	120.0	
4421	1	1	1	9500	1	1	154.0	
4422	1	1	1	9147	1	1	180.0	
4423	1	10	1	9773	1	1	152.0	

5 rows × 37 columns

In [6]:
```
df = df.dropna()
df.shape
```

Out[6]:
```
(4424, 37)
```

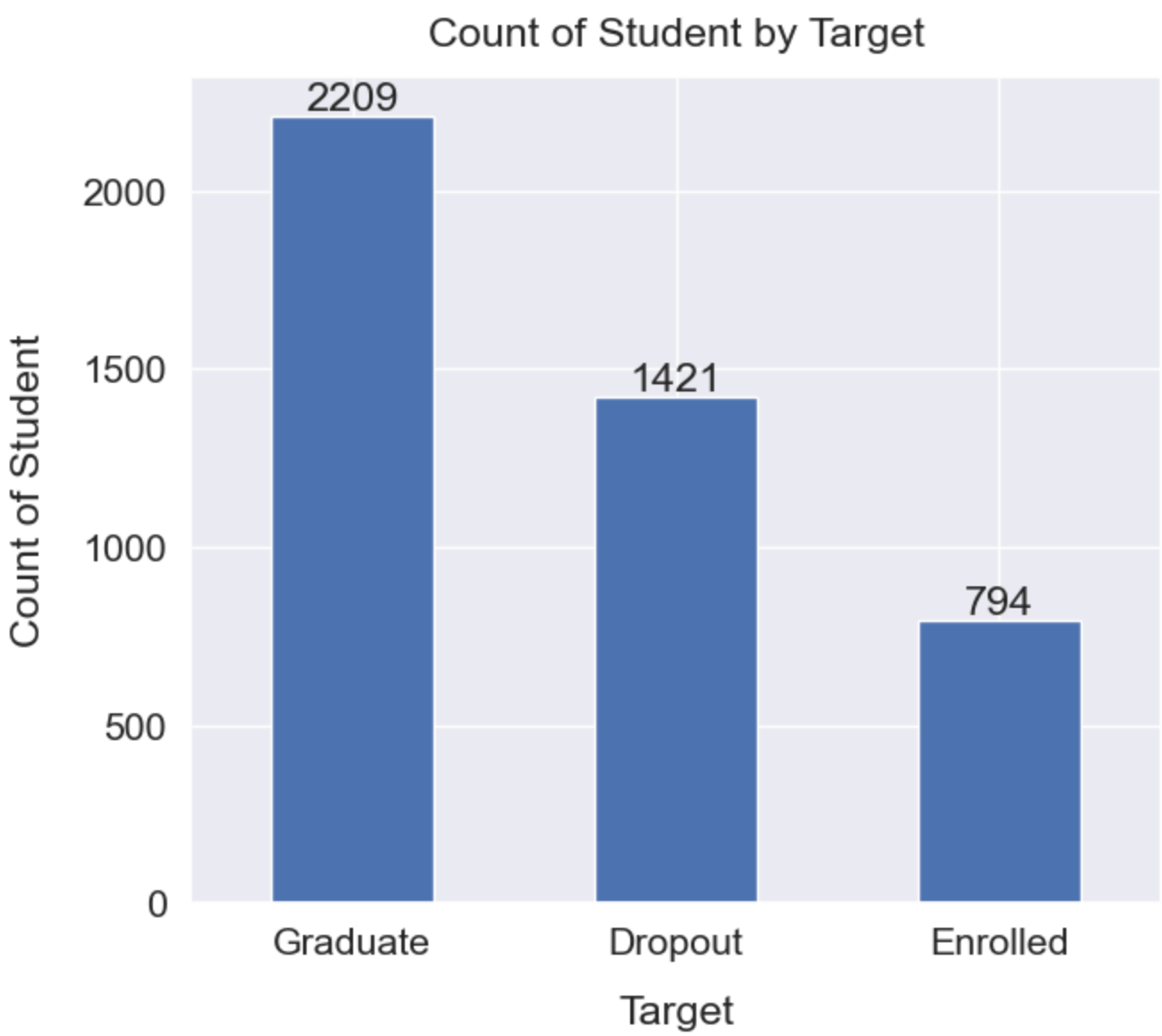
The data-set consists of 4424 records with 35 attributes and contains no missing values. The distribution and statistics are above and below.

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4424 entries, 0 to 4423
Data columns (total 37 columns):
 #   Column                                          Non-Null Count  Dtype  
---  ------                                          --------------  -----  
 0   Marital status                                  4424 non-null   int64  
 1   Application mode                                4424 non-null   int64  
 2   Application order                               4424 non-null   int64  
 3   Course                                          4424 non-null   int64  
 4   Daytime_evening_attendance                      4424 non-null   int64  
 5   Previous qualification                          4424 non-null   int64  
 6   Previous qualification (grade)                  4424 non-null   float64
 7   Nationality                                     4424 non-null   int64  
 8   Mother's qualification                          4424 non-null   int64  
 9   Father's qualification                          4424 non-null   int64  
 10  Mother's occupation                             4424 non-null   int64  
 11  Father's occupation                             4424 non-null   int64  
 12  Admission grade                                 4424 non-null   float64
 13  Displaced                                       4424 non-null   int64  
 14  Educational special needs                       4424 non-null   int64  
 15  Debtor                                          4424 non-null   int64  
 16  Tuition fees up to date                         4424 non-null   int64  
 17  Gender                                          4424 non-null   int64  
 18  Scholarship holder                              4424 non-null   int64  
 19  Age at enrollment                               4424 non-null   int64  
 20  International                                   4424 non-null   int64  
 21  Curricular units 1st sem (credited)             4424 non-null   int64  
 22  Curricular units 1st sem (enrolled)             4424 non-null   int64  
 23  Curricular units 1st sem (evaluations)          4424 non-null   int64  
 24  Curricular units 1st sem (approved)             4424 non-null   int64  
 25  Curricular units 1st sem (grade)                4424 non-null   float64
 26  Curricular units 1st sem (without evaluations)  4424 non-null   int64  
 27  Curricular units 2nd sem (credited)             4424 non-null   int64  
 28  Curricular units 2nd sem (enrolled)             4424 non-null   int64  
 29  Curricular units 2nd sem (evaluations)          4424 non-null   int64  
 30  Curricular units 2nd sem (approved)             4424 non-null   int64  
 31  Curricular units 2nd sem (grade)                4424 non-null   float64
 32  Curricular units 2nd sem (without evaluations)  4424 non-null   int64  
 33  Unemployment rate                               4424 non-null   float64
 34  Inflation rate                                  4424 non-null   float64
 35  GDP                                             4424 non-null   float64
 36  Target                                          4424 non-null   object 
dtypes: float64(7), int64(29), object(1)
memory usage: 1.2+ MB
```

In [8]:

```
sns.set(font_scale=1.4)
ax = df['Target'].value_counts().plot(kind='bar', figsize=(7, 6), rot=0)
ax.bar_label(ax.containers[0])
plt.xlabel("Target", labelpad=14)
plt.ylabel("Count of Student", labelpad=14)
plt.title("Count of Student by Target", y=1.02);
```



From above plot, we can see this problem was three-category classification task, and there exists strong imbalance between those three classes. The class, Graduate, has the majority count which is around 50% of the records and Dropout has 32% of total records. The Enrolled only has 18% of total records. Thus, during our training, we need to find a way to fix this imbalance issues.

Dropping Enrolled Student

In [9]:
```
df = df.drop(df[df.Target == 'Enrolled'].index)
df.shape
```

Out[9]:
```
(3630, 37)
```

Variables Correlation

In [11]:
```
numeric_cols = ['Marital status', 'Nationality','Displaced','Gender',
                'Age at enrollment', 'International',
                "Mother's qualification", "Father's qualification",
       "Mother's occupation", "Father's occupation",

       'Educational special needs', 'Debtor',
       'Tuition fees up to date',  'Scholarship holder',
        'Unemployment rate','Inflation rate', 'GDP',
```

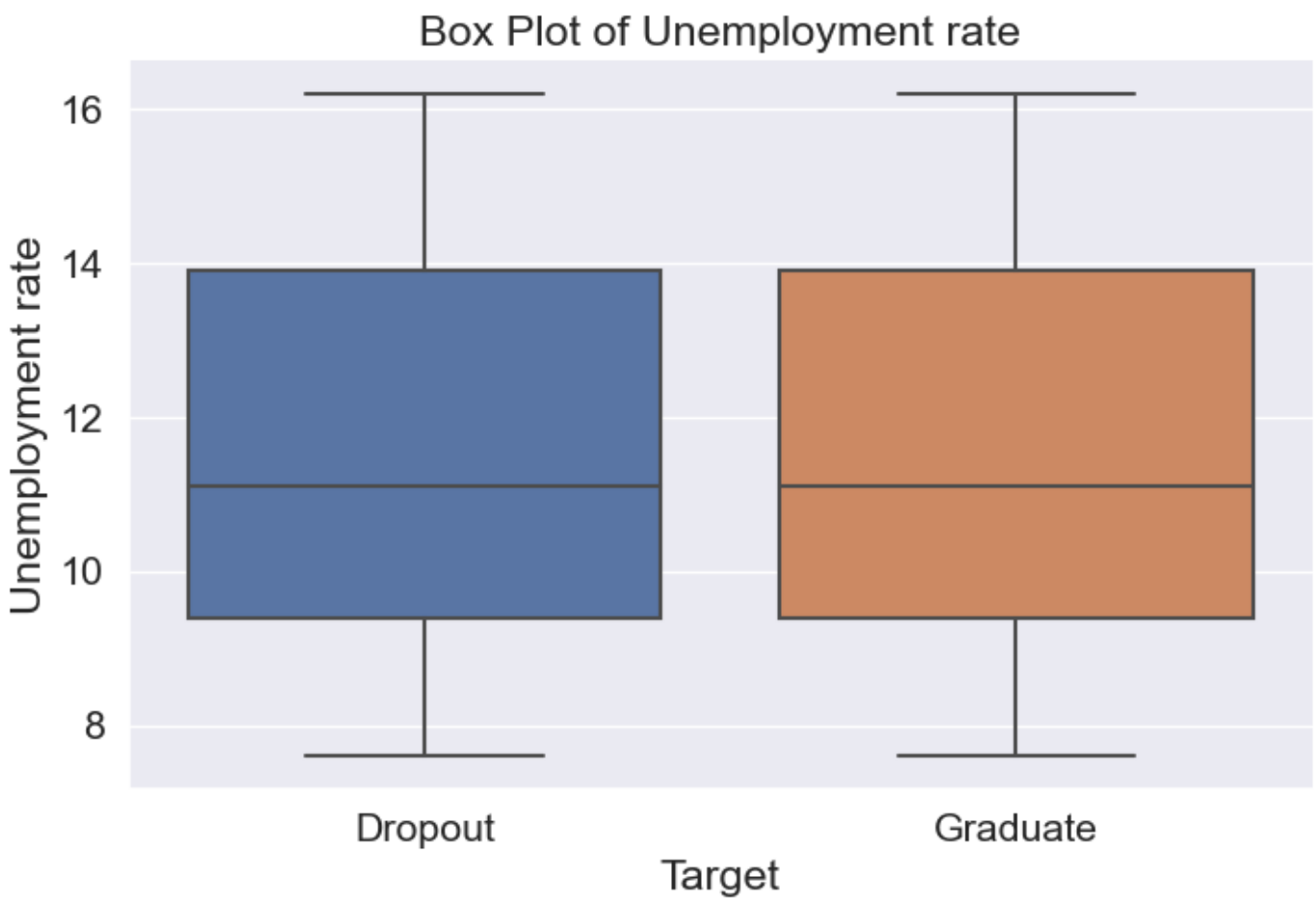
```
        'Application mode', 'Application order', 'Course',

       'Daytime_evening_attendance', 'Previous qualification',
       #'Previous qualification (grade)',
       # 'Admission grade',

       'Curricular units 1st sem (credited)',
       'Curricular units 1st sem (enrolled)',
       'Curricular units 1st sem (evaluations)',
       'Curricular units 1st sem (approved)',
       'Curricular units 1st sem (grade)',
       'Curricular units 1st sem (without evaluations)',
       'Curricular units 2nd sem (credited)',
       'Curricular units 2nd sem (enrolled)',
       'Curricular units 2nd sem (evaluations)',
       'Curricular units 2nd sem (approved)',
       'Curricular units 2nd sem (grade)',
       'Curricular units 2nd sem (without evaluations)']
df[numeric_cols].corr('spearman').style.background_gradient()
```

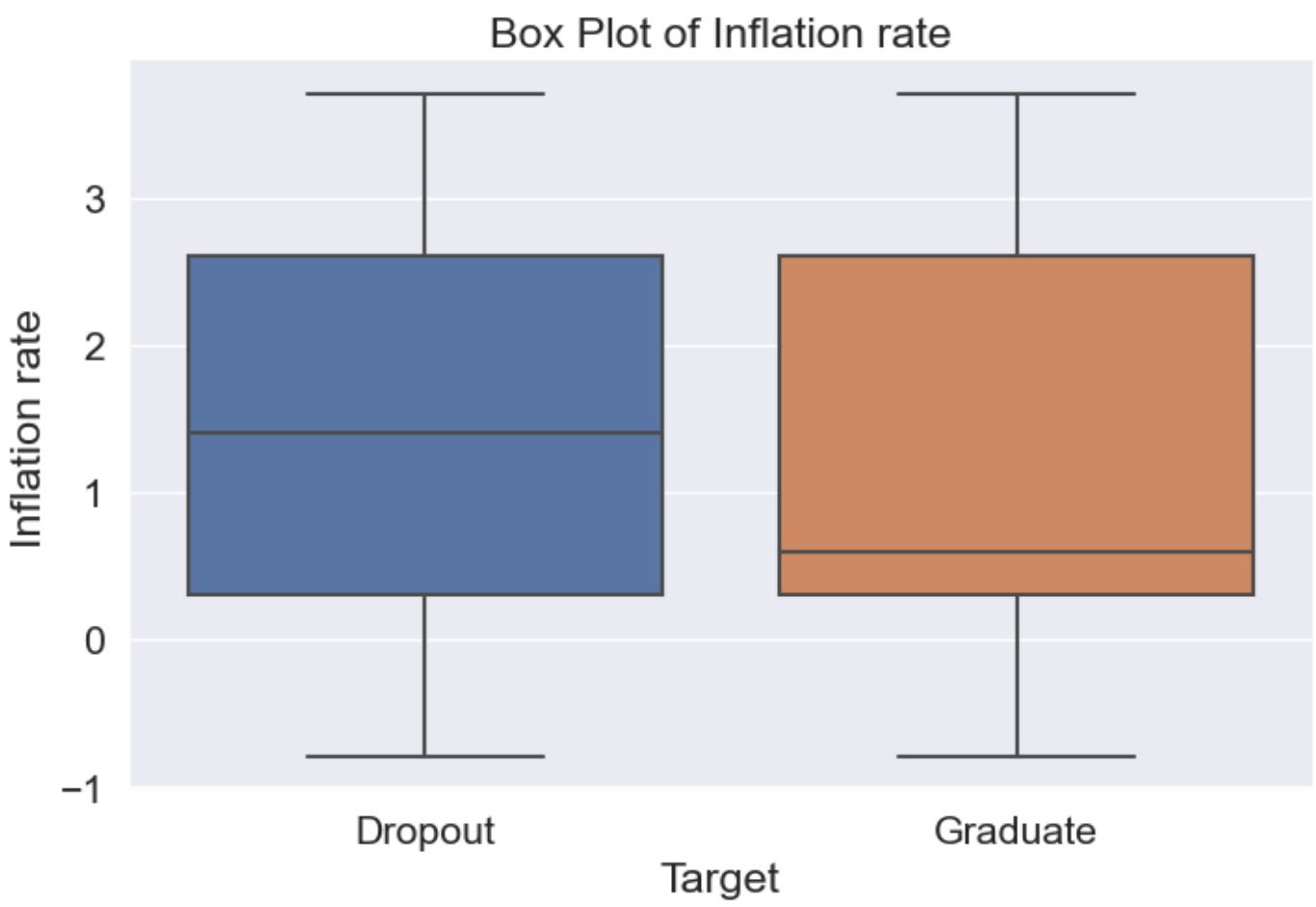
Out[11]:

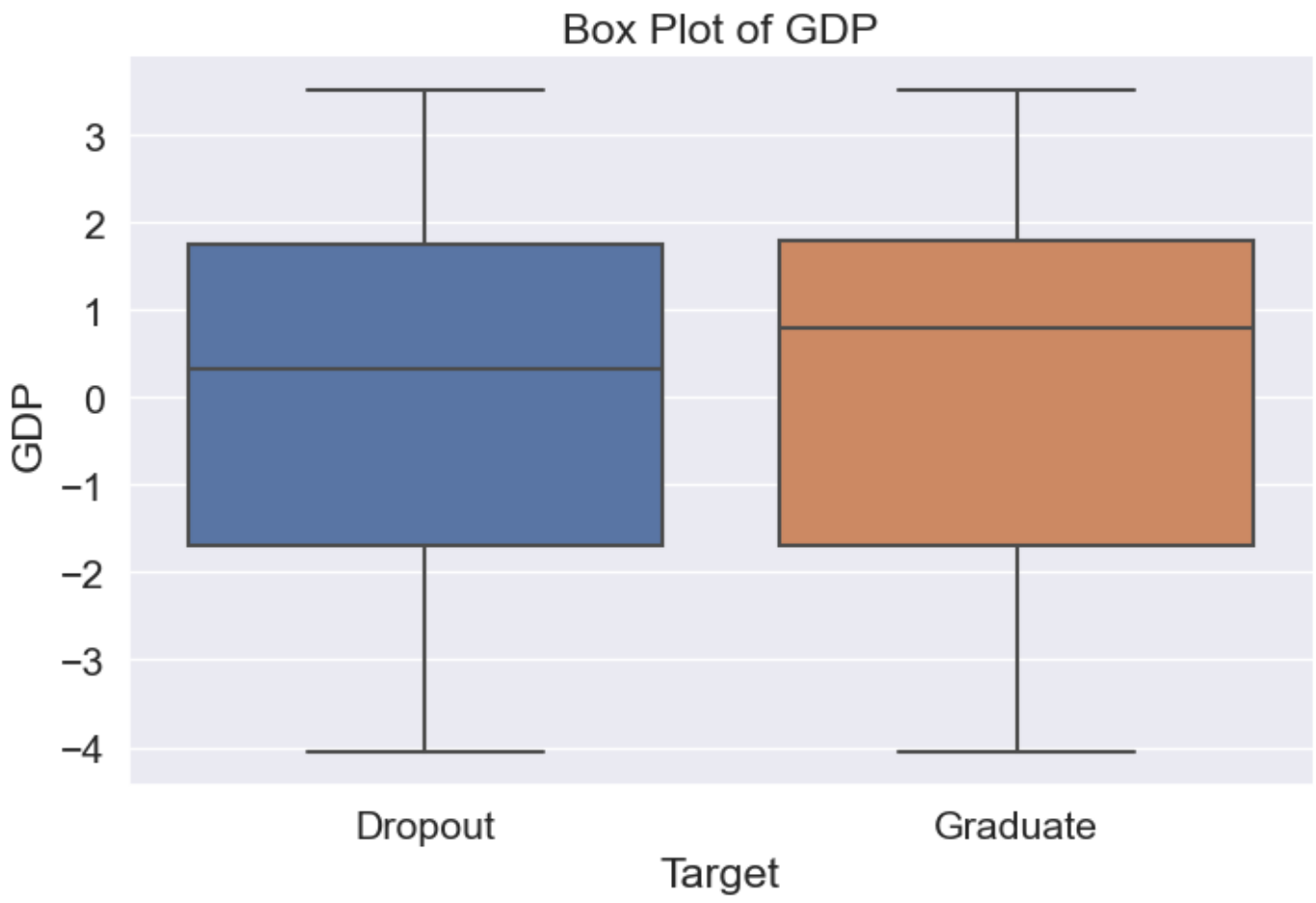
	Marital status	Nationality	Displaced	Gender	Age at enrollment	International	Mother's qualification
Marital status	1.000000	-0.029151	-0.280670	0.034059	0.487771	-0.029312	0.185557
Nationality	-0.029151	1.000000	-0.000933	-0.032763	0.014973	0.999910	-0.016578
Displaced	-0.280670	-0.000933	1.000000	-0.127896	-0.358344	-0.000790	-0.056694
Gender	0.034059	-0.032763	-0.127896	1.000000	0.217921	-0.032755	-0.046191
Age at enrollment	0.487771	0.014973	-0.358344	0.217921	1.000000	0.014839	0.171158
International	-0.029312	0.999910	-0.000790	-0.032755	0.014839	1.000000	-0.016059
Mother's qualification	0.185557	-0.016578	-0.056694	-0.046191	0.171158	-0.016059	1.000000
Father's qualification	0.117648	-0.071854	-0.049759	-0.061476	0.091516	-0.071386	0.445642
Mother's occupation	0.107951	0.016602	-0.030864	-0.023081	0.088996	0.016114	0.348163
Father's occupation	0.046720	0.025482	-0.025477	-0.024909	0.033188	0.025357	0.170351
Educational special needs	-0.030775	0.000875	-0.005099	-0.009794	-0.026482	0.000908	-0.008903
Debtor	0.038310	0.069663	-0.093718	0.052770	0.130889	0.069682	0.005673
Tuition fees up to date	-0.104008	-0.055761	0.105403	-0.122231	-0.218685	-0.055777	-0.014049
Scholarship holder	-0.111171	-0.019974	0.086337	-0.187994	-0.236935	-0.020298	0.025692
Unemployment rate	-0.039405	-0.007001	-0.116721	0.021554	0.011000	-0.007136	-0.093304
Inflation rate	0.003924	-0.001856	-0.005004	-0.007236	0.021894	-0.001865	0.040142
GDP	-0.073733	0.031021	0.064342	-0.024112	-0.062735	0.030689	-0.080006
Application mode	0.283749	-0.001916	-0.275745	0.172056	0.560823	-0.001597	0.069950
Application order	-0.181021	-0.018081	0.391173	-0.125412	-0.374271	-0.018055	-0.047869
Course	0.010963	0.006666	0.012632	-0.091264	-0.094831	0.006865	0.035886
Daytime_evening_attendance	-0.349339	0.032564	0.243653	-0.030507	-0.419255	0.032494	-0.145697
Previous qualification	0.198423	-0.040163	-0.205442	0.099248	0.408268	-0.039960	0.015110
Curricular units 1st sem (credited)	0.101918	0.023905	-0.129085	0.027893	0.303084	0.023964	-0.001530
Curricular units 1st sem (enrolled)	-0.003613	-0.003167	0.016764	-0.192456	-0.022160	-0.002866	0.021331
Curricular units 1st sem (evaluations)	0.060678	-0.008466	-0.051751	-0.033322	0.160976	-0.008153	0.043886
Curricular units 1st sem (approved)	-0.074359	-0.002099	0.096292	-0.249881	-0.204002	-0.001856	-0.023753
Curricular units 1st sem (grade)	-0.098640	0.000291	0.096843	-0.186811	-0.243264	0.000368	-0.053202
Curricular units 1st sem (without evaluations)	0.049069	-0.010229	-0.030897	0.004499	0.079966	-0.010031	0.000099
Curricular units 2nd sem (credited)	0.098985	0.022434	-0.132305	0.029235	0.302135	0.022549	-0.000300

	Marital status	Nationality	Displaced	Gender	Age at enrollment	International	Mother's qualification
Curricular units 2nd sem (enrolled)	-0.025889	-0.011215	0.033914	-0.198612	-0.057878	-0.011010	0.005997
Curricular units 2nd sem (evaluations)	0.012913	-0.027296	-0.019175	-0.058061	0.061485	-0.026901	0.021302
Curricular units 2nd sem (approved)	-0.080764	-0.016196	0.094870	-0.262117	-0.222433	-0.015868	-0.017841
Curricular units 2nd sem (grade)	-0.096833	-0.002299	0.087677	-0.205081	-0.246500	-0.002015	-0.045127
Curricular units 2nd sem (without evaluations)	0.059087	-0.023510	-0.046051	0.070138	0.116112	-0.023427	0.030562

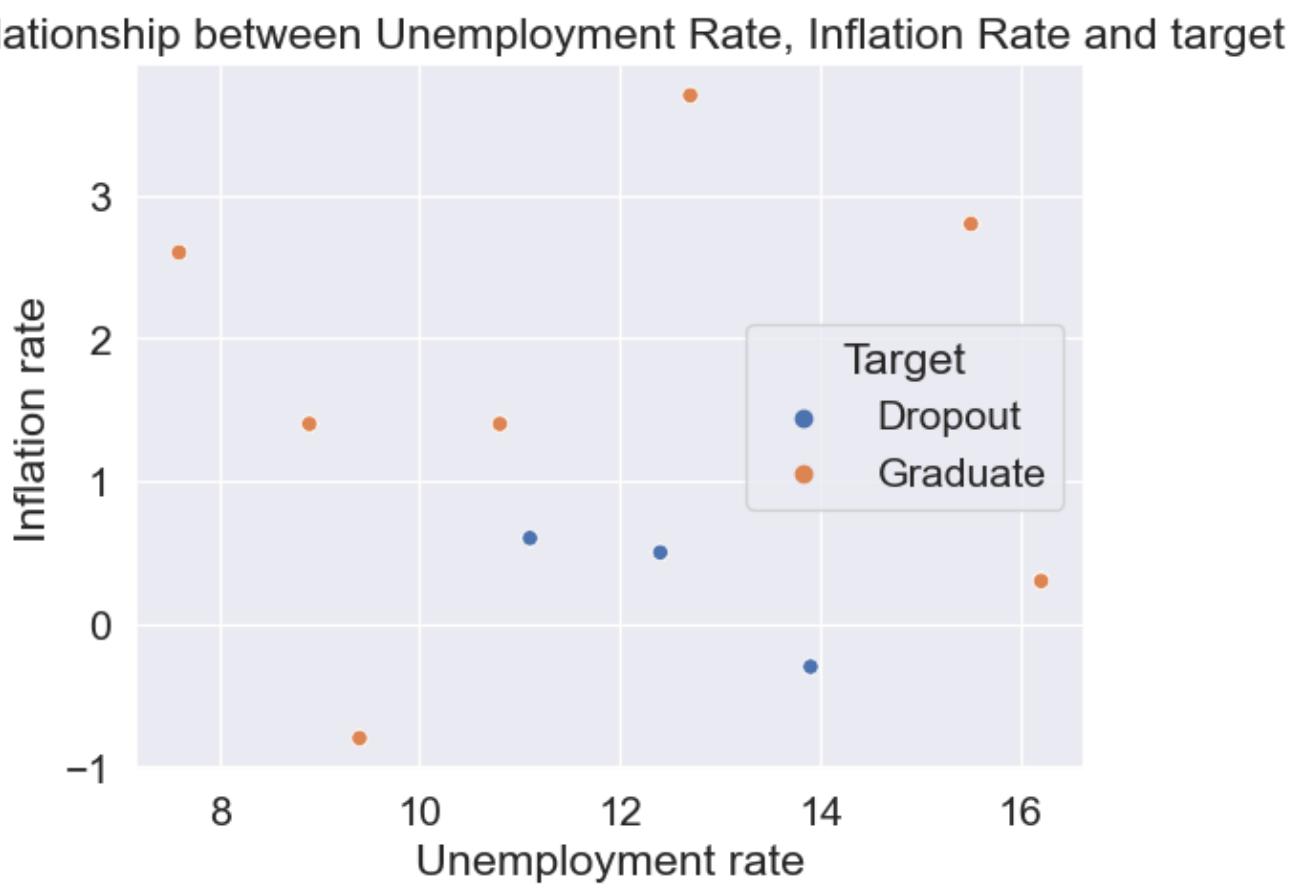
In [12]:
```
macroeconomic_col = ['Unemployment rate','Inflation rate', 'GDP']

for i in range(0, len(macroeconomic_col)):

    sns.catplot(x='Target', y=macroeconomic_col[i], data=df, kind="box", aspect=1.5)
    plt.title("Box Plot of " + macroeconomic_col[i])
```







In [13]:
```
sns.scatterplot(x='Unemployment rate', y='Inflation rate', hue="Target", data=df)
plt.title("Relationship between Unemployment Rate, Inflation Rate and target")
plt.show()
```



In [14]:
```
numeric_cols = ['Age at enrollment',
                'Unemployment rate','Inflation rate', 'GDP',
        'Application mode', 'Application order'
        ]

fig, axs = plt.subplots(2,3, figsize=(15, 10), facecolor='w', edgecolor='k')
fig.subplots_adjust(hspace = .5, wspace=.5)

axs = axs.ravel()

for i in range(0, len(numeric_cols)):

    axs[i].hist(df[numeric_cols[i]], bins=20)
    axs[i].set_title("Histogram of " + numeric_cols[i])
```

In [15]:
```
sns.pairplot(df[['Age at enrollment',
                 'Unemployment rate','Inflation rate', 'GDP',
        'Application mode', 'Application order' , 'Target']],
             hue='Target')
```

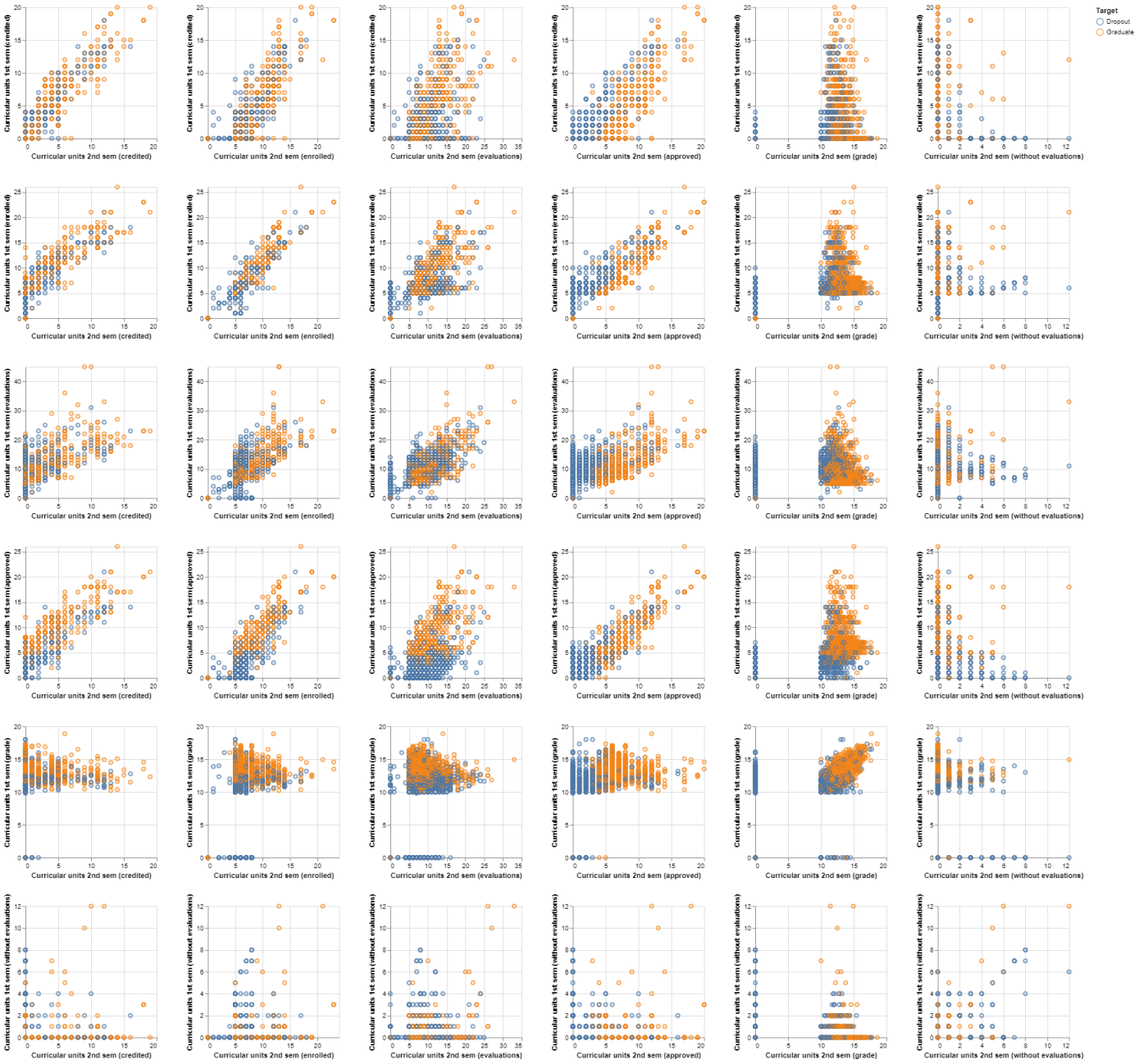
Out[15]:
```
<seaborn.axisgrid.PairGrid at 0x183ce406500>
```

In [16]:

```
numeric_cols = df.select_dtypes('number').columns.tolist()
numeric_cols

# performance of two semester vs Target


df_new = df[['Curricular units 1st sem (credited)', 'Curricular units 1st sem (enrolled)', 'Curr

first_sem = ['Curricular units 1st sem (credited)', 'Curricular units 1st sem (enrolled)', 'Curr

second_sem = ['Curricular units 2nd sem (credited)', 'Curricular units 2nd sem (enrolled)', 'Cur

plot = alt.Chart(df_new).mark_point().encode(
     alt.X(alt.repeat('column'), type='quantitative'),
     alt.Y(alt.repeat('row'), type='quantitative'),
     color = 'Target'
).properties(
    width=200,
    height=200
).repeat(
    row = first_sem,
    column = second_sem
)
```

```
plot
```

Out[16]:

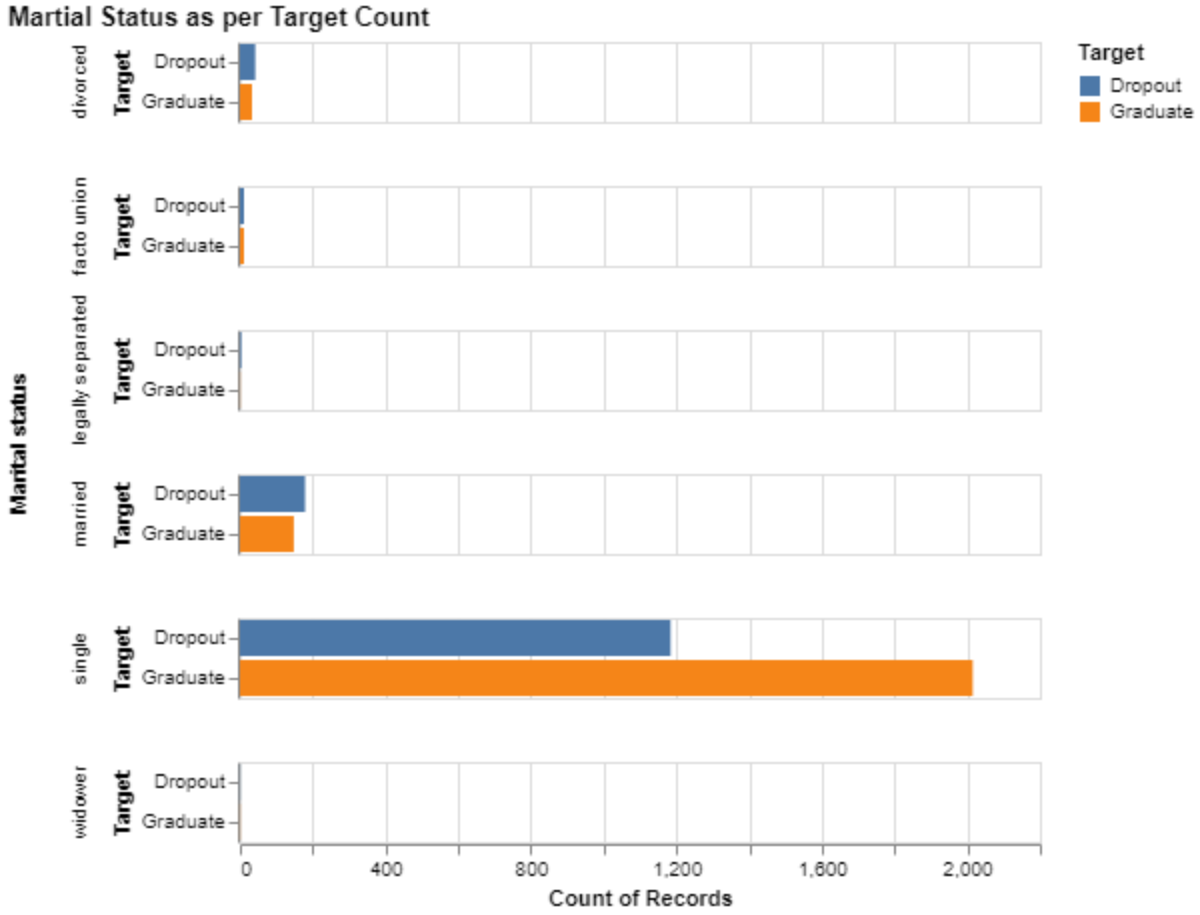


Marital Status

In [17]:

```
marital_dict = {1: 'single', 2: 'married', 3 : 'widower', 4 : 'divorced', 5 : 'facto union' , 6
df2=df.replace({"Marital status": marital_dict})

marital_bar = alt.Chart(df2).mark_bar().encode(
    x='count()',
    y='Target',
    color='Target',
    tooltip='count()'
).facet(
    row='Marital status',
    spacing=30,
    title='Martial Status as per Target Count'
)

marital_bar
```

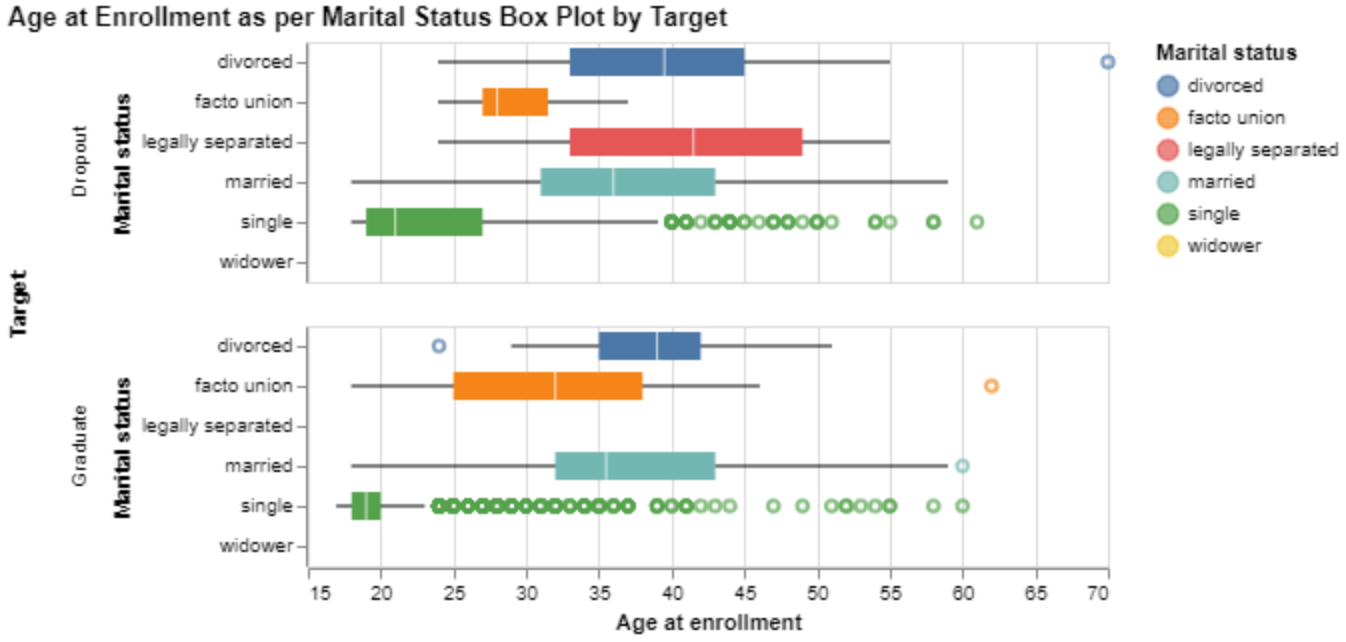
Out[17]:



In [18]:

```
marital_boxplot = alt.Chart(df2).mark_boxplot().encode(
    x=alt.X('Age at enrollment', scale=alt.Scale(domain=(15, 70))),
    y='Marital status:N',
    color='Marital status:N'
).facet(
    row='Target',
    title='Age at Enrollment as per Marital Status Box Plot by Target'
)

marital_boxplot
```

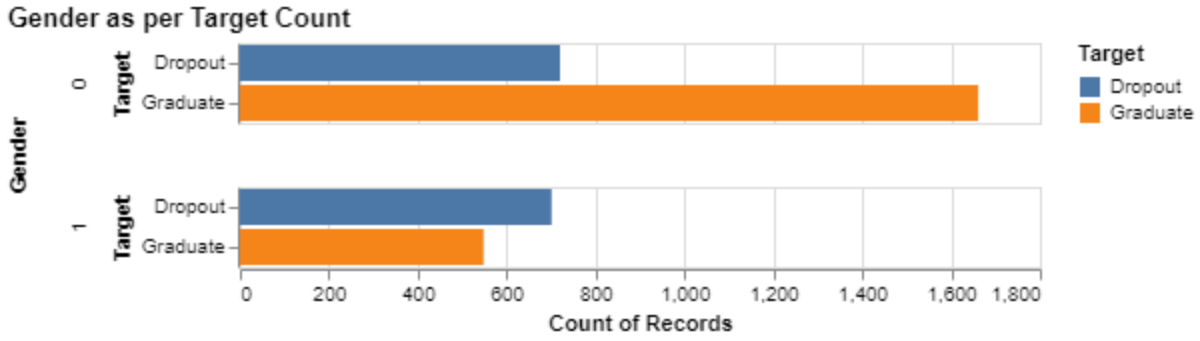
Out[18]:



Gender

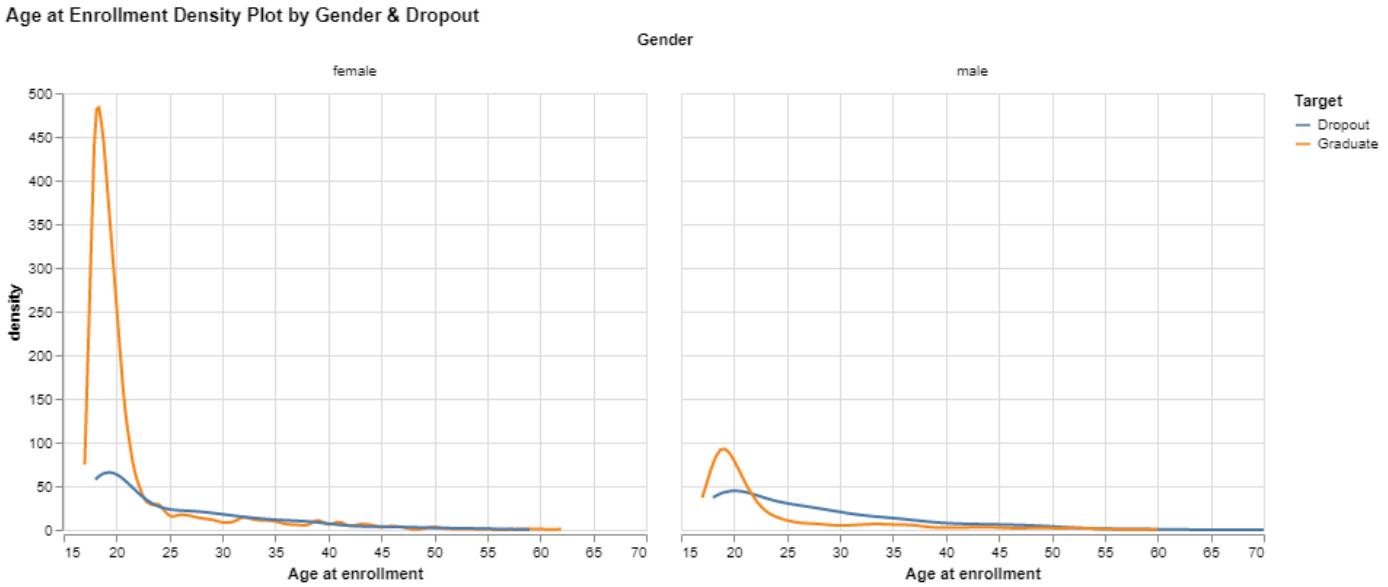
In [19]:
```
gender_bar = alt.Chart(df2).mark_bar().encode(
    x='count()',
    y='Target',
    color='Target',
    tooltip='count()'
).facet(
    row='Gender',
    spacing=30,
    title='Gender as per Target Count'
)

gender_bar
```

Out[19]:



In [20]:
```
gender_dict = {1: 'male', 0: 'female'}
df2=df2.replace({"Gender": gender_dict})

gender_density = (alt.Chart(df2)
 .transform_density(
     'Age at enrollment',
     groupby=['Target','Gender'],
     as_=['Age at enrollment', 'density'],
     counts=True,
 )
 .mark_line().encode(
     x='Age at enrollment',
     y='density:Q',
     color='Target',
 tooltip='Age at enrollment')
 .facet('Gender',
       title="Age at Enrollment Density Plot by Gender & Dropout"
       )
)

gender_density
```

Out[20]:

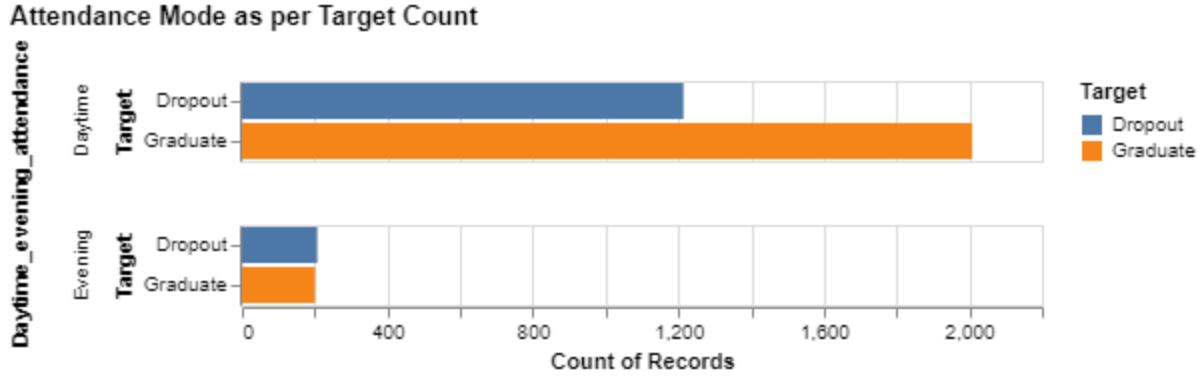


Attendance Mode

In [21]:

```
day_evening_dict = {1: 'Daytime', 0: 'Evening'}
df2 = df2.replace({'Daytime_evening_attendance': day_evening_dict})
gender_bar = alt.Chart(df2).mark_bar().encode(
    x='count()',
    y='Target',
    color='Target',
    tooltip='count()'
).facet(
    row="Daytime_evening_attendance",
    spacing=30,
    title='Attendance Mode as per Target Count'
)

gender_bar
```

Out[21]:



In [ ]:

In [ ]:

In [ ]: