Lesson 2\_2

**Dataset**

import warnings

import numpy as np

import pandas as pd

warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt

import seaborn as sns

sns.set()

%config InlineBackend.figure\_format = 'retina'

**Read dataset**

import pandas as pd

df = pd.read\_csv('telecom\_churn.csv')

df.head()

**Histogram**

features = ["Total day minutes", "Total intl calls"]

df[features].hist(figsize=(10, 4))

**Another Histogram**

df[features].plot(

kind='density', subplots=True, layout=(1, 2), sharex=False, figsize=(10, 4)

)

**Distplot chart bar**

import seaborn as sns

sns.distplot(df['Total intl calls'])

**box plot**

import seaborn as sns

sns.boxplot(x='Total intl calls', data=df)

**violin plot**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("telecom\_churn.csv")

\_, axes = plt.subplots(1,2,sharey=True, figsize=(6,4))

sns.boxplot(data=df["Total intl calls"], ax=axes[0])

sns.violinplot(data=df["Total intl calls"], ax=axes[1])

**describe table**

features = ["Total intl calls", "Total day minutes"]

df[features].describe()

**frequency table**

df["Churn"].value\_counts()

**bar plot**

import seaborn as sns

import matplotlib.pyplot as plt

\_, axes = plt.subplots(nrows=1, ncols=2, figsize=(12,4))

sns.countplot(x="Churn", data=df, ax=axes[0])

sns.countplot(x="State", data=df, ax=axes[1])

**correlation matrix**

numerical = list(

set(df.columns)

- {

"State",

"International plan",

"Voice mail plan",

"Area code",

"Churn",

"Customer service calls",

}

)

corr\_matrix = df[numerical].corr()

sns.heatmap(corr\_matrix)

**another correlation matrix**

numerical = list(

set(numerical)

- {

"Total day charge",

"Total eve charge",

"Total night charge",

"Total intl charge"

}

)

corr\_matrix = df[numerical].corr()

sns.heatmap(corr\_matrix)

**scatter plot**

import matplotlib.pyplot as plt

plt.scatter(df["Total day minutes"], df["Total night minutes"])

**jointplot**

sns.jointplot(x="Total day minutes", y="Total night minutes", data=df, kind="scatter")

**pairplot**

%config InlineBackend.figure\_format = 'png'

sns.pairplot(df[numerical])

**tight layout**

numerical.append("Customer service calls")

fig, axes = plt.subplots(nrows=3, ncols=4, figsize=(10, 7))

for idx, feat in enumerate(numerical):

ax = axes[int(idx/4), idx % 4]

sns.boxplot(x="Churn", y=feat, data=df, ax=ax)

ax.set\_xlabel("")

ax.set\_ylabel(feat)

fig.tight\_layout()

**another box and violin plot**

\_, axes = plt.subplots(1,2,sharey=True, figsize=(10,4))

sns.boxplot(x="Churn", y="Total day minutes", data=df, ax=axes[0])

sns.violinplot(x="Churn", y="Total day minutes", data=df, ax=axes[1])

**cat plot**

sns.catplot(

x="Churn",

y="Total day minutes",

col="Customer service calls",

data=df[df["Customer service calls"] < 8],

kind="box",

col\_wrap=4,

height=3,

aspect=0.8

)

**Count plot**

sns.countplot(x="Customer service calls", hue="Churn", data=df)

**another count plot**

\_, axes = plt.subplots(1,2,sharey=True, figsize=(10,4))

sns.countplot(x="International plan", hue="Churn", data=df, ax=axes[0])

sns.countplot(x="Voice mail plan", hue="Churn", data=df, ax=axes[1])

**crosstab**

pd.crosstab(df["State"], df["Churn"]).T

**group by**

import numpy as np

df.groupby(["State"])["Churn"].agg([np.mean]).sort\_values(by="mean", ascending=False).T