1216 수상작 리뷰

주제

Your goal is to use data from a mental health survey to explore factors that may cause individuals to experience depression.

https://www.kaggle.com/competitions/playground-series-s4e11/overview

데이터

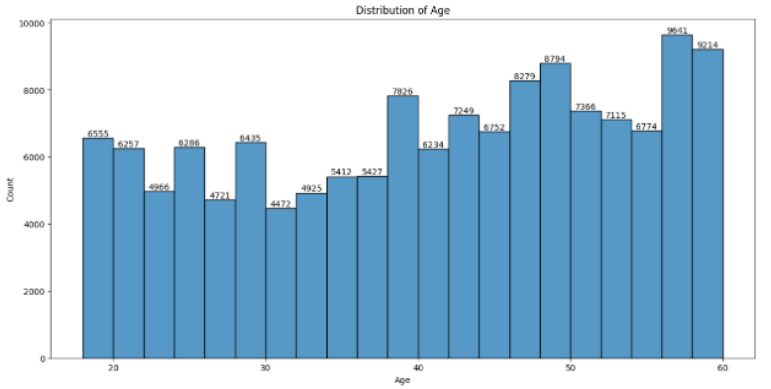
- **train.csv -** the training dataset; `class` is the binary target
- **test.csv -** the test dataset; your objective is to predict target `class` for each row
- **sample_submission.csv -** a sample submission file in the correct format

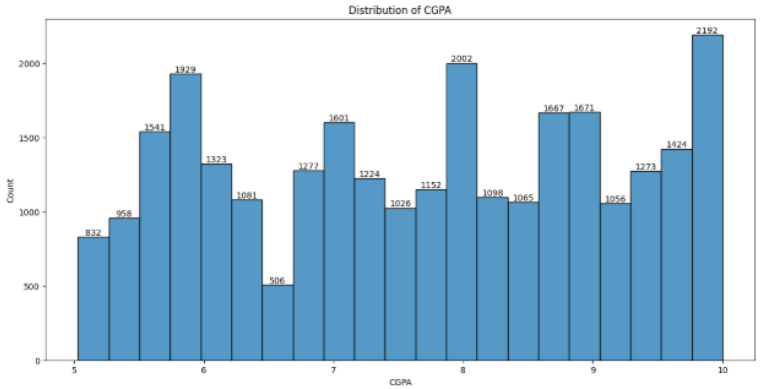
코드 정리

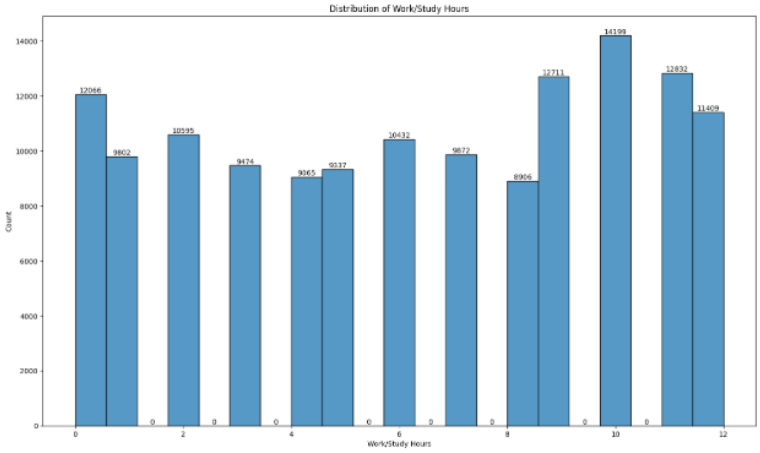
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

df_test = pd.read_csv('/kaggle/input/playground-series-s4e11/tes
df_train = pd.read_csv('/kaggle/input/playground-series-s4e11/tr
```

```
for i, col in enumerate(numerical_columns):
    plt.figure(figsize=(15, 86))
    plt.subplot(10, 1, i + 1)
    plot = sns.histplot(x=df_train[col], bins=21)
```

```
    plt.title(f'Distribution of {col}')

    for i in plot.containers:
        plot.bar_label(i)

plt.tight_layout()
plt.show()
```

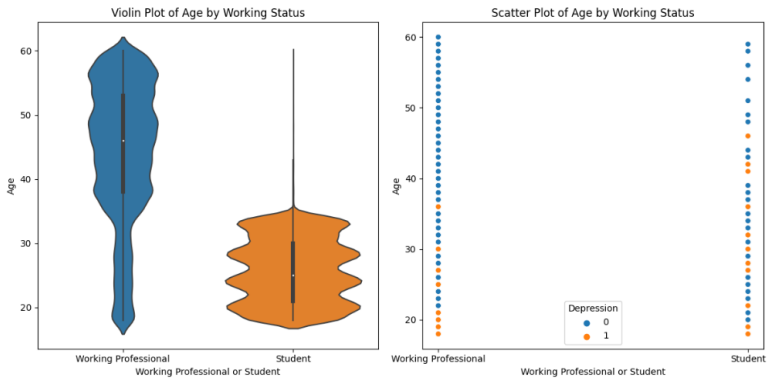






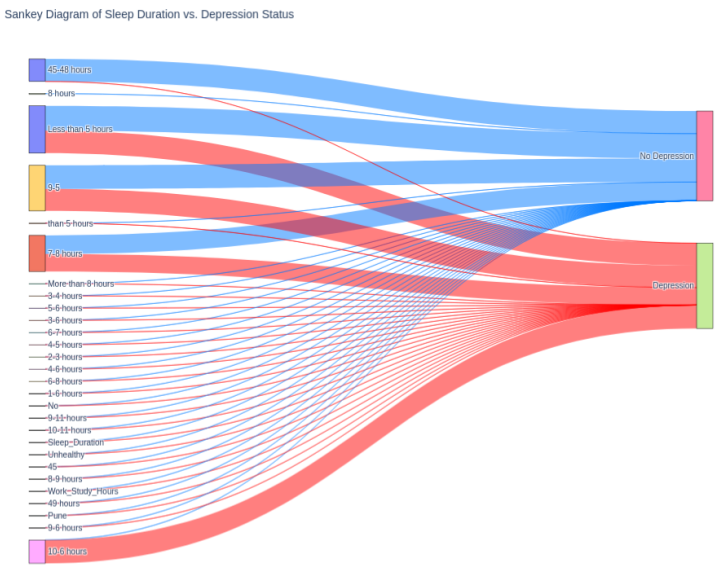

```
# Set up the figure with specified size
plt.figure(figsize=(12, 6))

# First subplot: Violin plot
plt.subplot(1, 2, 1)
```

```
sns.violinplot(data=df_train, x="Working Professional or Student
plt.title("Violin Plot of Age by Working Status")

# Second subplot: Scatter plot
plt.subplot(1, 2, 2)
sns.scatterplot(data=df_train, x="Working Professional or Stude
plt.title("Scatter Plot of Age by Working Status")

# Show the plots
plt.tight_layout()
plt.show()
```




```
# Set up source and target for the Sankey diagram
source = [label_to_index[sleep] for sleep in sleep_labels for _
target = [label_to_index["No Depression"] if i % 2 == 0 else lab
value = values_no_depression + values_depression

# Define colors for the links based on depression status
link_colors = ['rgba(0, 123, 255, 0.5)' if target[i] == label_to
               for i in range(len(target))]

# Create the Sankey diagram
fig = go.Figure(data=[go.Sankey(
    node=dict(
        pad=15,
```

```
        thickness=20,
        line=dict(color="black", width=0.5),
        label=labels,
    ),
    link=dict(
        source=source,
        target=target,
        value=value,
        color=link_colors  # Apply colors to the links
    )
)])

# Set the figure size and title
fig.update_layout(
    title_text="Sankey Diagram of Sleep Duration vs. Depression
    font_size=10,
    width=1000,
    height=800
)

# Show the plot
fig.show()
```




```
#Random forest
random_forest = RandomForestClassifier(n_estimators = 100, rando
random_forest.fit(X_train, y_train)
y_pred = random_forest.predict(X_test)
acc_random_forest = round(random_forest.score(X_test, y_test) *
mse_random_forest = round(mean_squared_error(y_test, Y_pred),2)
model_accuracies["acc_random_forest"]=[acc_random_forest,mse_ran
acc_random_forest
```

91.32

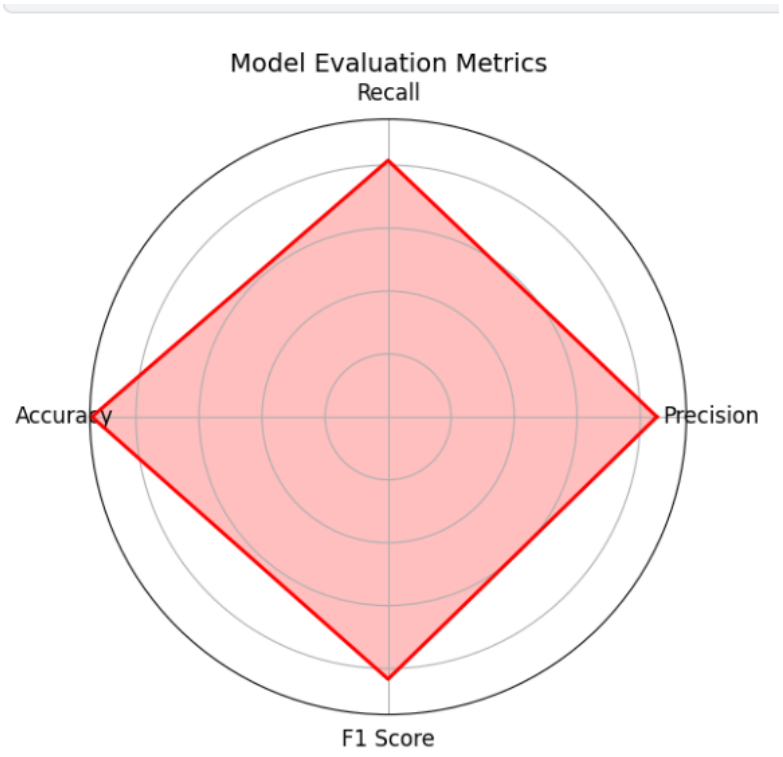
```
#Decision Tree Classifier
tree_clf = DecisionTreeClassifier(random_state=42)
tree_clf.fit(X_train, y_train)
y_pred_tree = tree_clf.predict(X_test)
decisiontree_score = round(accuracy_score(y_test, y_pred_tree)*1
decisiontree_mse = round(mean_squared_error(y_test, Y_pred),2)
model_accuracies["DecisionTree"]=[decisiontree_score,decisiontre
decisiontree_score
```

86.12

```
#AdaBoost with Random Forest
weak_classifier = RandomForestClassifier(n_estimators = 100)
ada_boost = AdaBoostClassifier(estimator=weak_classifier, n_esti
ada_boost.fit(X_train, y_train)
y_pred = ada_boost.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
mse = round(mean_squared_error(y_test, y_pred),2)
print(f"Accuracy of AdaBoost: {accuracy * 100:.2f}")
model_accuracies["AdaBoostRandomForest"]=[round(accuracy*100,2),
```

```
import matplotlib.pyplot as plt
import numpy as np

# Metrics and values
metrics = ['Precision', 'Recall', 'Accuracy', 'F1 Score']
values = [precision*100, recall*100, accuracy * 100, f1*100]

num_vars = len(metrics)

# Compute angle for each axis
angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tol

# Make the chart circular
values += values[:1]  # To close the circle
angles += angles[:1]

# Create the radar chart
fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=Tru
ax.fill(angles, values, color='red', alpha=0.25)
ax.plot(angles, values, color='red', linewidth=2)

# Set the labels
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
```

```
ax.set_xticklabels(metrics, fontsize=12)

# Title
plt.title('Model Evaluation Metrics', size=14)

# Show the plot
plt.tight_layout()
plt.show()
```



차별점 및 배울점

- 다양한 시각화를 통해 전처리를 진행해 다방면에서 코드에 대해서 이해할 수 있었으며, 흔하게 볼 수 없는 시각화를 이용한 것이 인상적이었다.
- 라인 다이어그램은 데이터의 흐름과 양을 한눈에 파악할 수 있도록 하였으며, 복잡한 관계를 명확하게 전달하였다는 점에서 새로운 시각화를 알아갈 수 있었다.