Untitled

June 1, 2024

[6]:

```python
import pandas as pd

# Load the datasets
train_df = pd.read_csv('/Users/tvvr/Downloads/churn-bigml-80.csv')
test_df = pd.read_csv('/Users/tvvr/Downloads/churn-bigml-20.csv')

# Display the first few rows of the training dataset
print("Training Dataset:")
print(train_df.head())

# Display the first few rows of the testing dataset
print("Testing Dataset:")
print(test_df.head())

# Summary statistics for training dataset
print("Training Dataset Summary:")
print(train_df.describe())

# Summary statistics for testing dataset
print("Testing Dataset Summary:")
print(test_df.describe())

# Check for missing values in the training dataset
print("Missing Values in Training Dataset:")
print(train_df.isnull().sum())

# Check for missing values in the testing dataset
print("Missing Values in Testing Dataset:")
print(test_df.isnull().sum())

# Get a summary of the columns in the training dataset
print("Training Dataset Columns:")
print(train_df.columns)

# Get a summary of the columns in the testing dataset
print("Testing Dataset Columns:")
print(test_df.columns)
```

```
Training Dataset:
  State  Account length  Area code International plan Voice mail plan  \
0    KS             128        415                 No             Yes
1    OH             107        415                 No             Yes
2    NJ             137        415                 No              No
3    OH              84        408                Yes              No
4    OK              75        415                Yes              No

   Number vmail messages  Total day minutes  Total day calls  \
0                     25              265.1              110
1                     26              161.6              123
2                      0              243.4              114
3                      0              299.4               71
4                      0              166.7              113

   Total day charge  Total eve minutes  Total eve calls  Total eve charge  \
0             45.07              197.4               99             16.78
1             27.47              195.5              103             16.62
2             41.38              121.2              110             10.30
3             50.90               61.9               88              5.26
4             28.34              148.3              122             12.61

   Total night minutes  Total night calls  Total night charge  \
0                244.7                 91               11.01
1                254.4                103               11.45
2                162.6                104                7.32
3                196.9                 89                8.86
4                186.9                121                8.41

   Total intl minutes  Total intl calls  Total intl charge  \
0                10.0                 3               2.70
1                13.7                 3               3.70
2                12.2                 5               3.29
3                 6.6                 7               1.78
4                10.1                 3               2.73

   Customer service calls  Churn
0                       1  False
1                       1  False
2                       0  False
3                       2  False
4                       3  False
Testing Dataset:
  State  Account length  Area code International plan Voice mail plan  \
0    LA             117        408                 No              No
1    IN              65        415                 No              No
2    NY             161        415                 No              No
3    SC             111        415                 No              No
```

```
4    HI              49       510                No              No

   Number vmail messages  Total day minutes  Total day calls  \
0                      0              184.5               97
1                      0              129.1              137
2                      0              332.9               67
3                      0              110.4              103
4                      0              119.3              117

   Total day charge  Total eve minutes  Total eve calls  Total eve charge  \
0             31.37              351.6               80             29.89
1             21.95              228.5               83             19.42
2             56.59              317.8               97             27.01
3             18.77              137.3              102             11.67
4             20.28              215.1              109             18.28

   Total night minutes  Total night calls  Total night charge  \
0                215.8                 90                9.71
1                208.8                111                9.40
2                160.6                128                7.23
3                189.6                105                8.53
4                178.7                 90                8.04

   Total intl minutes  Total intl calls  Total intl charge  \
0                 8.7                 4               2.35
1                12.7                 6               3.43
2                 5.4                 9               1.46
3                 7.7                 6               2.08
4                11.1                 1               3.00

   Customer service calls  Churn
0                       1  False
1                       4   True
2                       4   True
3                       2  False
4                       1  False
Training Dataset Summary:
       Account length    Area code  Number vmail messages  Total day minutes  \
count     2666.000000  2666.000000            2666.000000         2666.00000
mean       100.620405   437.438860               8.021755          179.48162
std         39.563974    42.521018              13.612277           54.21035
min          1.000000   408.000000               0.000000            0.00000
25%         73.000000   408.000000               0.000000          143.40000
50%        100.000000   415.000000               0.000000          179.95000
75%        127.000000   510.000000              19.000000          215.90000
max        243.000000   510.000000              50.000000          350.80000

       Total day calls  Total day charge  Total eve minutes  Total eve calls  \
```

```
count      2666.000000       2666.000000        2666.000000      2666.000000
mean        100.310203         30.512404         200.386159       100.023631
std          19.988162          9.215733          50.951515        20.161445
min           0.000000          0.000000           0.000000         0.000000
25%          87.000000         24.380000         165.300000        87.000000
50%         101.000000         30.590000         200.900000       100.000000
75%         114.000000         36.700000         235.100000       114.000000
max         160.000000         59.640000         363.700000       170.000000

       Total eve charge  Total night minutes  Total night calls  \
count       2666.000000          2666.000000        2666.000000
mean          17.033072           201.168942         100.106152
std            4.330864            50.780323          19.418459
min            0.000000            43.700000          33.000000
25%           14.050000           166.925000          87.000000
50%           17.080000           201.150000         100.000000
75%           19.980000           236.475000         113.000000
max           30.910000           395.000000         166.000000

       Total night charge  Total intl minutes  Total intl calls  \
count         2666.000000         2666.000000       2666.000000
mean             9.052689           10.237022          4.467367
std              2.285120            2.788349          2.456195
min              1.970000            0.000000          0.000000
25%              7.512500            8.500000          3.000000
50%              9.050000           10.200000          4.000000
75%             10.640000           12.100000          6.000000
max             17.770000           20.000000         20.000000

       Total intl charge  Customer service calls
count        2666.000000             2666.000000
mean            2.764490                1.562641
std             0.752812                1.311236
min             0.000000                0.000000
25%             2.300000                1.000000
50%             2.750000                1.000000
75%             3.270000                2.000000
max             5.400000                9.000000
Testing Dataset Summary:
       Account length   Area code  Number vmail messages  Total day minutes  \
count      667.000000  667.000000             667.000000         667.000000
mean       102.841079  436.157421               8.407796         180.948126
std         40.819480   41.783305              13.994480          55.508628
min          1.000000  408.000000               0.000000          25.900000
25%         76.000000  408.000000               0.000000         146.250000
50%        102.000000  415.000000               0.000000         178.300000
75%        128.000000  415.000000              20.000000         220.700000
max        232.000000  510.000000              51.000000         334.300000
```

```
       Total day calls  Total day charge  Total eve minutes  Total eve calls  \
count       667.000000        667.000000         667.000000       667.000000
mean        100.937031         30.761769         203.355322       100.476762
std          20.396790          9.436463          49.719268        18.948262
min          30.000000          4.400000          48.100000        37.000000
25%          87.500000         24.860000         171.050000        88.000000
50%         101.000000         30.310000         203.700000       101.000000
75%         115.000000         37.520000         236.450000       113.000000
max         165.000000         56.830000         361.800000       168.000000

       Total eve charge  Total night minutes  Total night calls  \
count        667.000000           667.000000         667.000000
mean          17.285262           199.685307         100.113943
std            4.226160            49.759931          20.172505
min            4.090000            23.200000          42.000000
25%           14.540000           167.950000          86.000000
50%           17.310000           201.600000         100.000000
75%           20.095000           231.500000         113.500000
max           30.750000           367.700000         175.000000

       Total night charge  Total intl minutes  Total intl calls  \
count          667.000000          667.000000        667.000000
mean             8.985907           10.238381          4.527736
std              2.239429            2.807850          2.482442
min              1.040000            0.000000          0.000000
25%              7.560000            8.600000          3.000000
50%              9.070000           10.500000          4.000000
75%             10.420000           12.050000          6.000000
max             16.550000           18.300000         18.000000

       Total intl charge  Customer service calls
count         667.000000              667.000000
mean            2.764948                1.563718
std             0.758167                1.333357
min             0.000000                0.000000
25%             2.320000                1.000000
50%             2.840000                1.000000
75%             3.255000                2.000000
max             4.940000                8.000000
Missing Values in Training Dataset:
State                     0
Account length            0
Area code                 0
International plan        0
Voice mail plan           0
Number vmail messages     0
Total day minutes         0
```

```
Total day calls           0
Total day charge          0
Total eve minutes         0
Total eve calls           0
Total eve charge          0
Total night minutes       0
Total night calls         0
Total night charge        0
Total intl minutes        0
Total intl calls          0
Total intl charge         0
Customer service calls    0
Churn                     0
dtype: int64
Missing Values in Testing Dataset:
State                     0
Account length            0
Area code                 0
International plan        0
Voice mail plan           0
Number vmail messages     0
Total day minutes         0
Total day calls           0
Total day charge          0
Total eve minutes         0
Total eve calls           0
Total eve charge          0
Total night minutes       0
Total night calls         0
Total night charge        0
Total intl minutes        0
Total intl calls          0
Total intl charge         0
Customer service calls    0
Churn                     0
dtype: int64
Training Dataset Columns:
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
Testing Dataset Columns:
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
```

```
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
```

[7]:
```
# Calculate total charges by summing all charge columns
train_df['total_charges'] = (train_df['Total day charge'] +
                             train_df['Total eve charge'] +
                             train_df['Total night charge'] +
                             train_df['Total intl charge'])

test_df['total_charges'] = (test_df['Total day charge'] +
                            test_df['Total eve charge'] +
                            test_df['Total night charge'] +
                            test_df['Total intl charge'])

# Assuming monthly_charges can be derived directly from total_charges by␣
↪dividing by the account length
train_df['monthly_charges'] = train_df['total_charges'] / train_df['Account␣
↪length']
test_df['monthly_charges'] = test_df['total_charges'] / test_df['Account␣
↪length']

# Handle any potential missing or infinite values resulting from the␣
↪calculations
train_df['total_charges'].replace([float('inf'), -float('inf')], pd.NA,␣
↪inplace=True)
test_df['total_charges'].replace([float('inf'), -float('inf')], pd.NA,␣
↪inplace=True)

train_df['total_charges'].fillna(0, inplace=True)
test_df['total_charges'].fillna(0, inplace=True)

# Check the first few rows to confirm the calculations
print("Training Dataset with Calculated Charges:")
print(train_df[['Account length', 'Total day charge', 'Total eve charge',␣
↪'Total night charge', 'Total intl charge', 'total_charges',␣
↪'monthly_charges']].head())

print("Testing Dataset with Calculated Charges:")
print(test_df[['Account length', 'Total day charge', 'Total eve charge', 'Total␣
↪night charge', 'Total intl charge', 'total_charges', 'monthly_charges']].
↪head())
```

```
Training Dataset with Calculated Charges:
```

```
   Account length  Total day charge  Total eve charge  Total night charge  \
0             128             45.07             16.78               11.01
1             107             27.47             16.62               11.45
2             137             41.38             10.30                7.32
3              84             50.90              5.26                8.86
4              75             28.34             12.61                8.41

   Total intl charge  total_charges  monthly_charges
0               2.70          75.56         0.590313
1               3.70          59.24         0.553645
2               3.29          62.29         0.454672
3               1.78          66.80         0.795238
4               2.73          52.09         0.694533
Testing Dataset with Calculated Charges:
   Account length  Total day charge  Total eve charge  Total night charge  \
0             117             31.37             29.89                9.71
1              65             21.95             19.42                9.40
2             161             56.59             27.01                7.23
3             111             18.77             11.67                8.53
4              49             20.28             18.28                8.04

   Total intl charge  total_charges  monthly_charges
0               2.35          73.32         0.626667
1               3.43          54.20         0.833846
2               1.46          92.29         0.573230
3               2.08          41.05         0.369820
4               3.00          49.60         1.012245
```

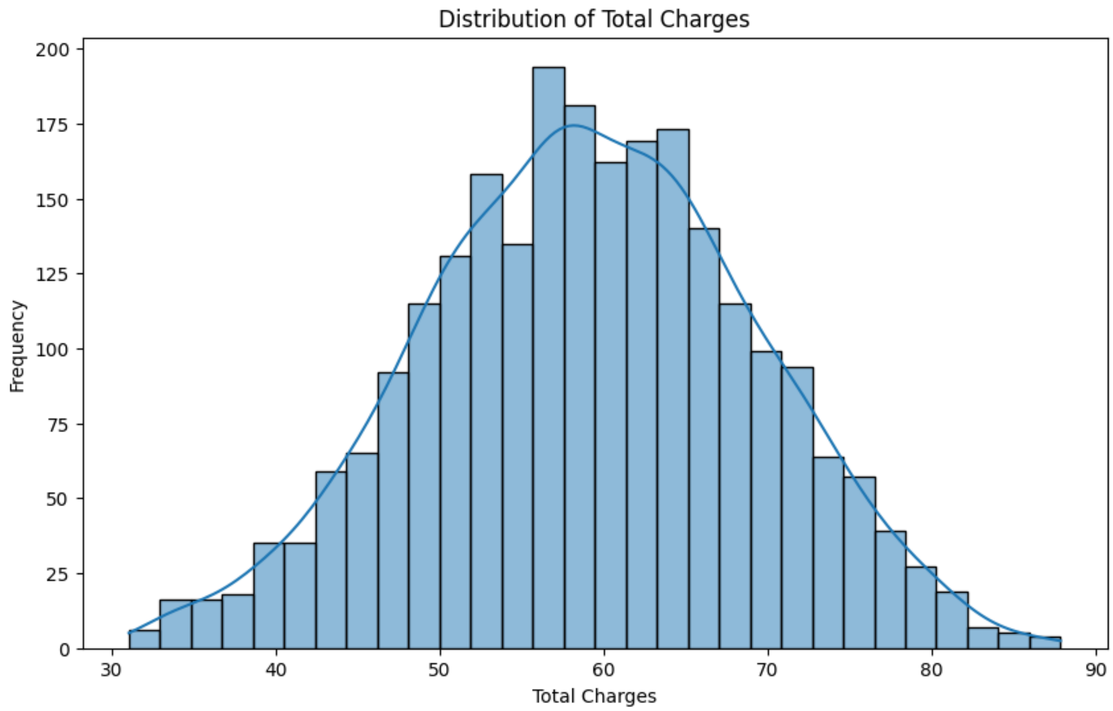
[8]:
```
# Handle missing values
train_df = train_df.dropna()
test_df = test_df.dropna()

# Handle outliers using IQR method
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    return df

# Apply to relevant columns
train_df = remove_outliers(train_df, 'total_charges')
train_df = remove_outliers(train_df, 'monthly_charges')

# Repeat for test dataset if necessary
```

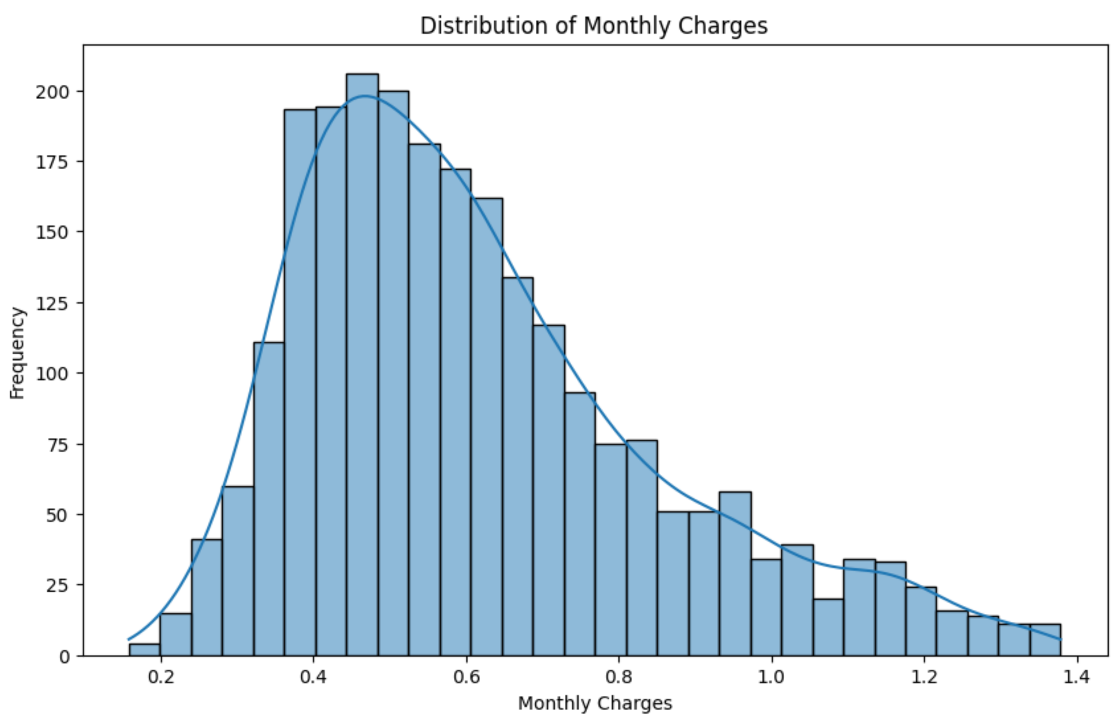
```
test_df = remove_outliers(test_df, 'total_charges')
test_df = remove_outliers(test_df, 'monthly_charges')

# Check for inconsistencies and correct them
train_df = train_df[train_df['total_charges'] >= 0]
test_df = test_df[test_df['total_charges'] >= 0]
```

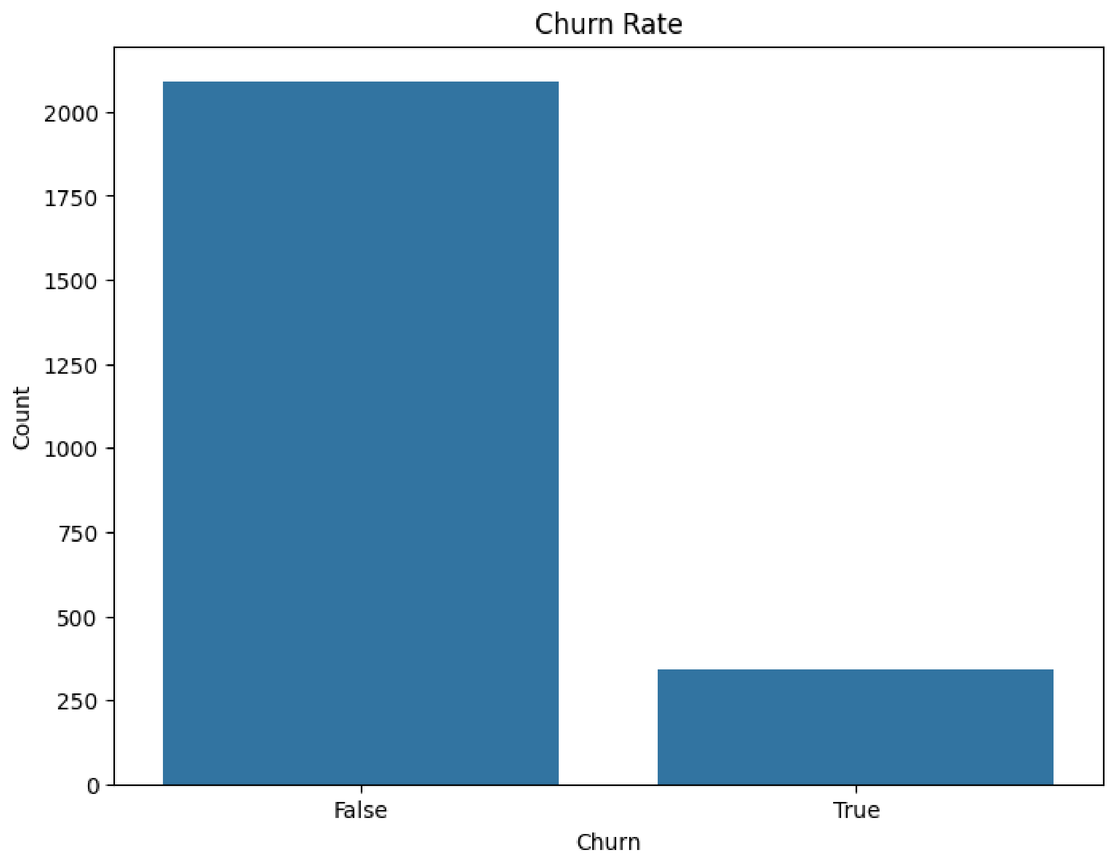
[9]:
```
# Feature Engineering
# Create new features based on domain knowledge

# Example: Creating a feature for 'average_call_duration'
train_df['average_call_duration'] = train_df['Total day minutes'] /␣
 ↪train_df['Total day calls']
test_df['average_call_duration'] = test_df['Total day minutes'] /␣
 ↪test_df['Total day calls']

# Example: Creating a feature for 'total_interaction_time' if not present
train_df['total_interaction_time'] = (train_df['Total day calls'] +
                                       train_df['Total eve calls'] +
                                       train_df['Total night calls'] +
                                       train_df['Total intl calls']) *␣
 ↪train_df['average_call_duration']

test_df['total_interaction_time'] = (test_df['Total day calls'] +
                                      test_df['Total eve calls'] +
                                      test_df['Total night calls'] +
                                      test_df['Total intl calls']) *␣
 ↪test_df['average_call_duration']

# Drop columns that are redundant or not useful for prediction
columns_to_drop = ['State']  # Example of a column to drop
train_df = train_df.drop(columns=columns_to_drop)
test_df = test_df.drop(columns=columns_to_drop)

# Convert categorical variables to numerical using one-hot encoding
train_df = pd.get_dummies(train_df, drop_first=True)
test_df = pd.get_dummies(test_df, drop_first=True)

# Ensure that both datasets have the same columns after encoding
train_df, test_df = train_df.align(test_df, join='inner', axis=1)
```

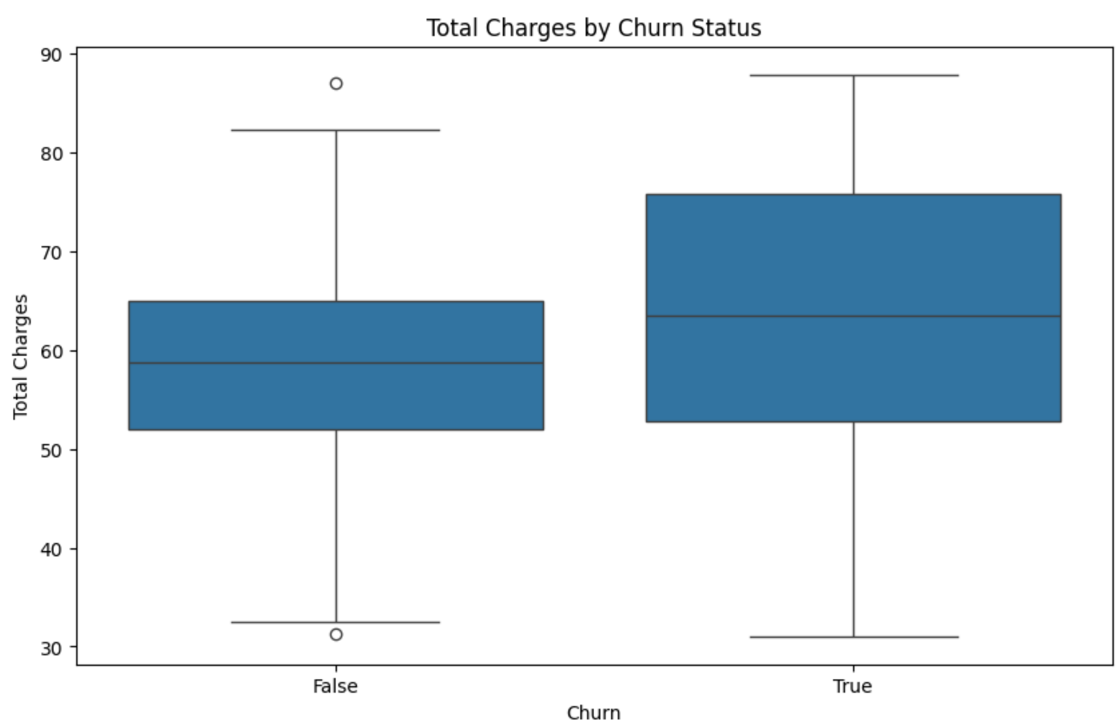
[11]:
```
# EDA visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Distribution of Total Charges
plt.figure(figsize=(10, 6))
```

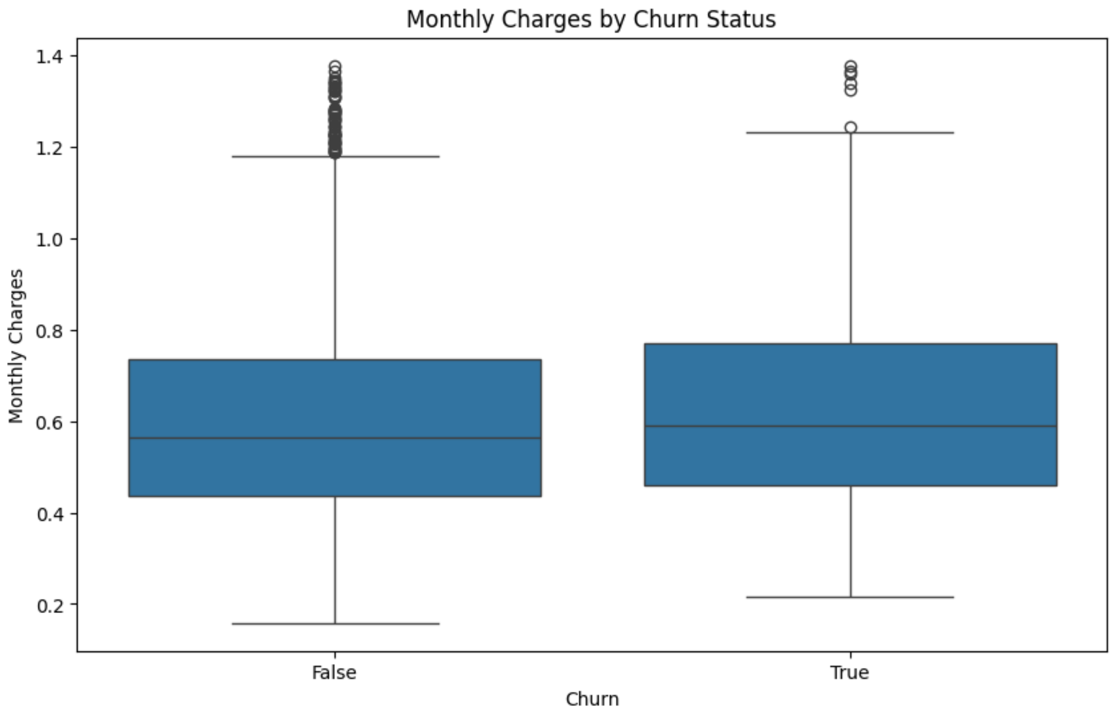
```python
sns.histplot(train_df['total_charges'], kde=True, bins=30)
plt.title('Distribution of Total Charges')
plt.xlabel('Total Charges')
plt.ylabel('Frequency')
plt.show()

# 2. Distribution of Monthly Charges
plt.figure(figsize=(10, 6))
sns.histplot(train_df['monthly_charges'], kde=True, bins=30)
plt.title('Distribution of Monthly Charges')
plt.xlabel('Monthly Charges')
plt.ylabel('Frequency')
plt.show()

# 3. Churn rate
plt.figure(figsize=(8, 6))
sns.countplot(x='Churn', data=train_df)
plt.title('Churn Rate')
plt.xlabel('Churn')
plt.ylabel('Count')
plt.show()

# 4. Total charges by churn status
plt.figure(figsize=(10, 6))
sns.boxplot(x='Churn', y='total_charges', data=train_df)
plt.title('Total Charges by Churn Status')
plt.xlabel('Churn')
plt.ylabel('Total Charges')
plt.show()

# 5. Monthly charges by churn status
plt.figure(figsize=(10, 6))
sns.boxplot(x='Churn', y='monthly_charges', data=train_df)
plt.title('Monthly Charges by Churn Status')
plt.xlabel('Churn')
plt.ylabel('Monthly Charges')
plt.show()

# 6. Correlation heatmap
plt.figure(figsize=(12, 10))
corr_matrix = train_df.corr()
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

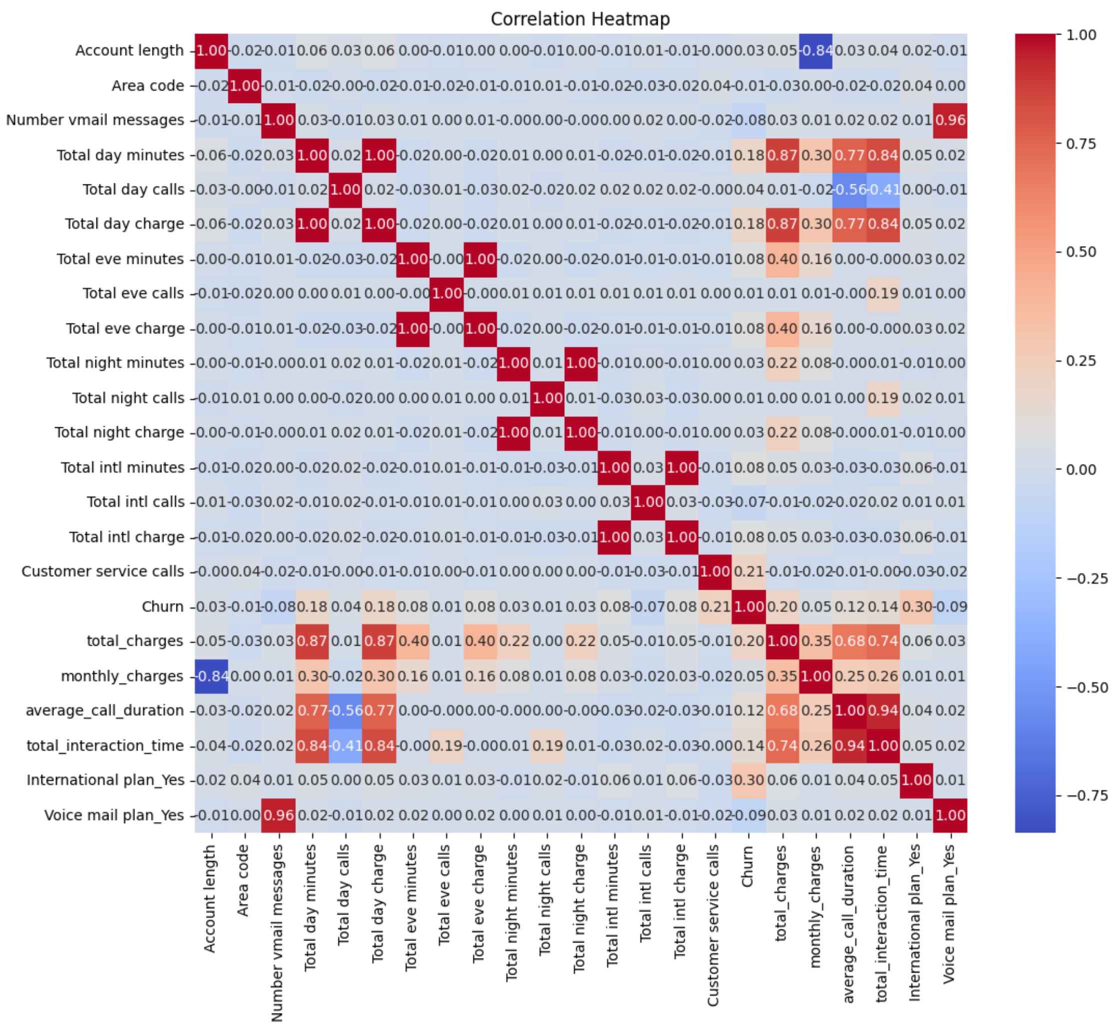

Distribution of Total Charges
Frequency
Total Charges
0
25
50
75
100
125
150
175
200
30
40
50
60
70
80
90


Distribution of Monthly Charges
Frequency
Monthly Charges
0
25
50
75
100
125
150
175
200
0.2
0.4
0.6
0.8
1.0
1.2
1.4


Churn Rate
2000
1750
1500
1250
1000
750
500
250
0
Count
False
True
Churn


Total Charges by Churn Status
90
80
70
60
50
40
30
Total Charges
False
True
Churn


Monthly Charges by Churn Status
Monthly Charges
1.4
1.2
1.0
0.8
0.6
0.4
0.2
False
True
Churn




```
[ ]: """
Key Observations from correlation heatmap:
1. Total Charges and Monthly Charges have a strong positive correlation with␣
 ↪each other.
2. Total Day Charge, Total Eve Charge, Total Night Charge, and Total Intl␣
 ↪Charge are strongly correlated with Total Charges, which is expected as they␣
 ↪contribute to the total charges.
3. Average Call Duration has a moderate correlation with Total Day Minutes and␣
 ↪Total Day Calls.
4. Churn does not show strong correlation with most features, which indicates␣
 ↪the necessity of more complex models for prediction.
"""
```

```
[12]: from sklearn.model_selection import train_test_split
```

```
# Define features and target
X = train_df.drop('Churn', axis=1)
y = train_df['Churn'].astype('int')  # Ensure the target is of integer type

# Split the dataset into train and validation sets
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

print(f"Training set size: {X_train.shape}")
print(f"Validation set size: {X_val.shape}")
```

```
Training set size: (1944, 22)
Validation set size: (486, 22)
```

[14]:

```
from sklearn.preprocessing import StandardScaler

# Initialize the scaler
scaler = StandardScaler()

# Scale the training and validation sets
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)

# Reinitialize the models
log_reg = LogisticRegression(max_iter=2000)
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
xgb_clf = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')

# Train Logistic Regression
log_reg.fit(X_train_scaled, y_train)
log_reg_pred = log_reg.predict(X_val_scaled)

# Train Random Forest
rf_clf.fit(X_train, y_train)
rf_clf_pred = rf_clf.predict(X_val)

# Train XGBoost
xgb_clf.fit(X_train, y_train)
xgb_clf_pred = xgb_clf.predict(X_val)
```

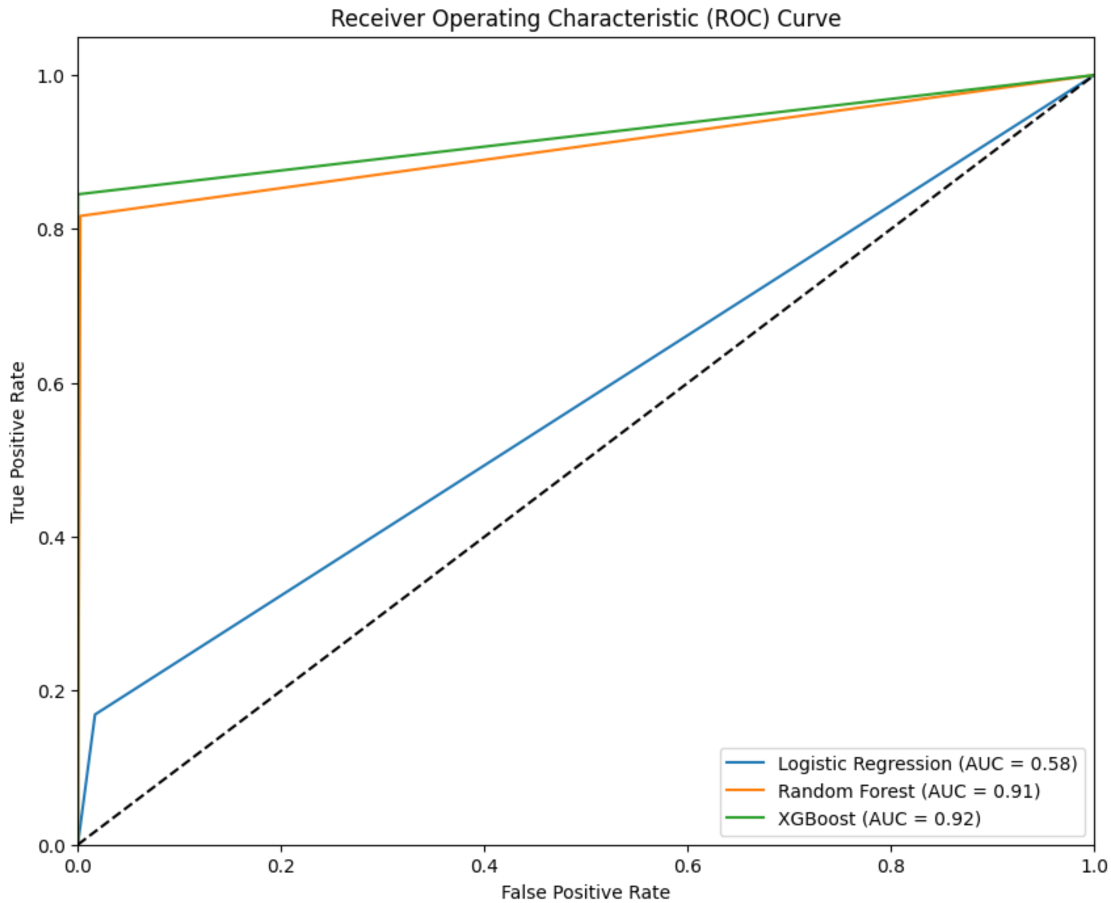
[16]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score, roc_auc_score, roc_curve, auc
import matplotlib.pyplot as plt

# Function to evaluate model performance
def evaluate_model(y_true, y_pred, model_name):
    accuracy = accuracy_score(y_true, y_pred)
```

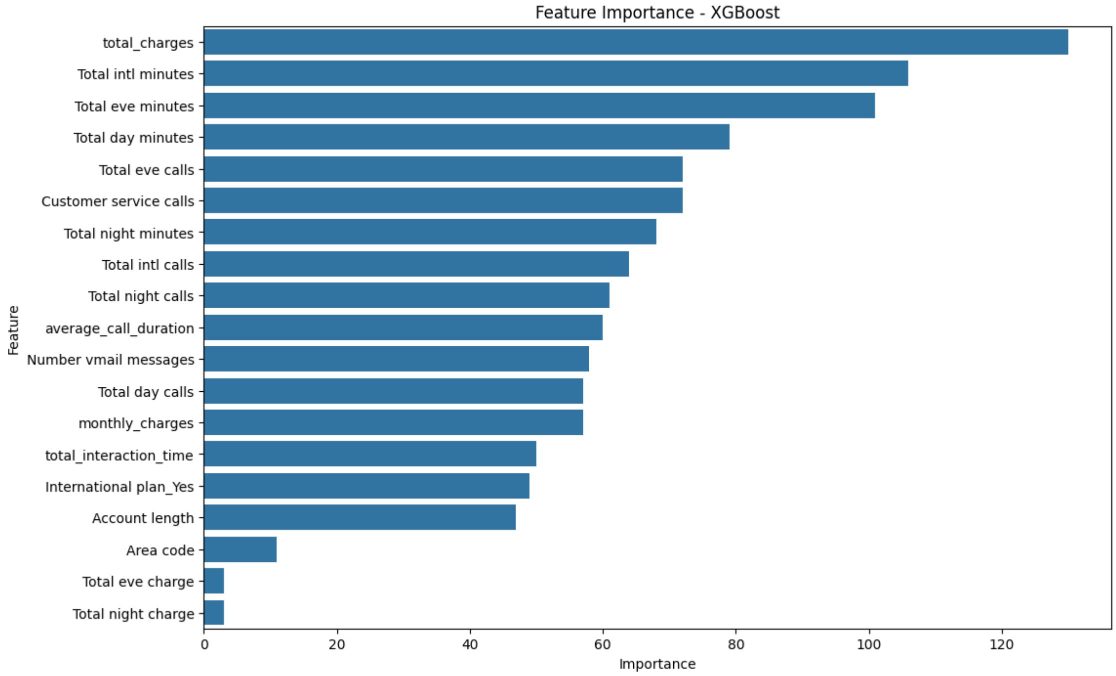
```
    precision = precision_score(y_true, y_pred)
    recall = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)
    roc_auc = roc_auc_score(y_true, y_pred)

    print(f"Model: {model_name}")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1 Score: {f1:.4f}")
    print(f"ROC-AUC: {roc_auc:.4f}")
    print("\n")

    return accuracy, precision, recall, f1, roc_auc

# Evaluate Logistic Regression
log_reg_metrics = evaluate_model(y_val, log_reg_pred, "Logistic Regression")

# Evaluate Random Forest
rf_clf_metrics = evaluate_model(y_val, rf_clf_pred, "Random Forest")

# Evaluate XGBoost
xgb_clf_metrics = evaluate_model(y_val, xgb_clf_pred, "XGBoost")

# Plot ROC Curves
plt.figure(figsize=(10, 8))

models = [log_reg, rf_clf, xgb_clf]
model_names = ["Logistic Regression", "Random Forest", "XGBoost"]
preds = [log_reg_pred, rf_clf_pred, xgb_clf_pred]

for model, name, pred in zip(models, model_names, preds):
    fpr, tpr, _ = roc_curve(y_val, pred)
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

```
Model: Logistic Regression
Accuracy: 0.8642
```

```
Precision: 0.6316
Recall: 0.1690
F1 Score: 0.2667
ROC-AUC: 0.5761


Model: Random Forest
Accuracy: 0.9712
Precision: 0.9831
Recall: 0.8169
F1 Score: 0.8923
ROC-AUC: 0.9072


Model: XGBoost
Accuracy: 0.9774
Precision: 1.0000
Recall: 0.8451
F1 Score: 0.9160
ROC-AUC: 0.9225
```



[17]:
```
# Feature importance for XGBoost
xgb_importance = xgb_clf.get_booster().get_score(importance_type='weight')
sorted_importance = sorted(xgb_importance.items(), key=lambda x: x[1],␣
↪reverse=True)

# Convert to a DataFrame for better visualization
importance_df = pd.DataFrame(sorted_importance, columns=['Feature',␣
↪'Importance'])

# Plot feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.title('Feature Importance - XGBoost')
plt.show()
```



[18]:

```
from sklearn.model_selection import GridSearchCV

# Define parameter grid
param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.8, 0.9, 1.0],
}

# Initialize GridSearchCV
grid_search = GridSearchCV(estimator=xgb_clf, param_grid=param_grid, cv=5,␣
↪scoring='roc_auc', verbose=2, n_jobs=-1)

# Fit GridSearchCV
grid_search.fit(X_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_
print(f"Best parameters found: {best_params}")

# Train XGBoost with the best parameters
xgb_best = xgb.XGBClassifier(**best_params, use_label_encoder=False,␣
↪eval_metric='logloss')
xgb_best.fit(X_train, y_train)
```

```python
xgb_best_pred = xgb_best.predict(X_val)

# Evaluate the tuned model
evaluate_model(y_val, xgb_best_pred, "XGBoost Tuned")
```

```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
Best parameters found: {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators':
200, 'subsample': 1.0}
Model: XGBoost Tuned
Accuracy: 0.9774
Precision: 1.0000
Recall: 0.8451
F1 Score: 0.9160
ROC-AUC: 0.9225
```

[18]:
```
(0.977366255144033,
 1.0,
 0.8450704225352113,
 0.916030534351145,
 0.9225352112676056)
```

[19]:
```python
# Print the best parameters
best_params = grid_search.best_params_
print(f"Best parameters found: {best_params}")
```

```
Best parameters found: {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators':
200, 'subsample': 1.0}
```

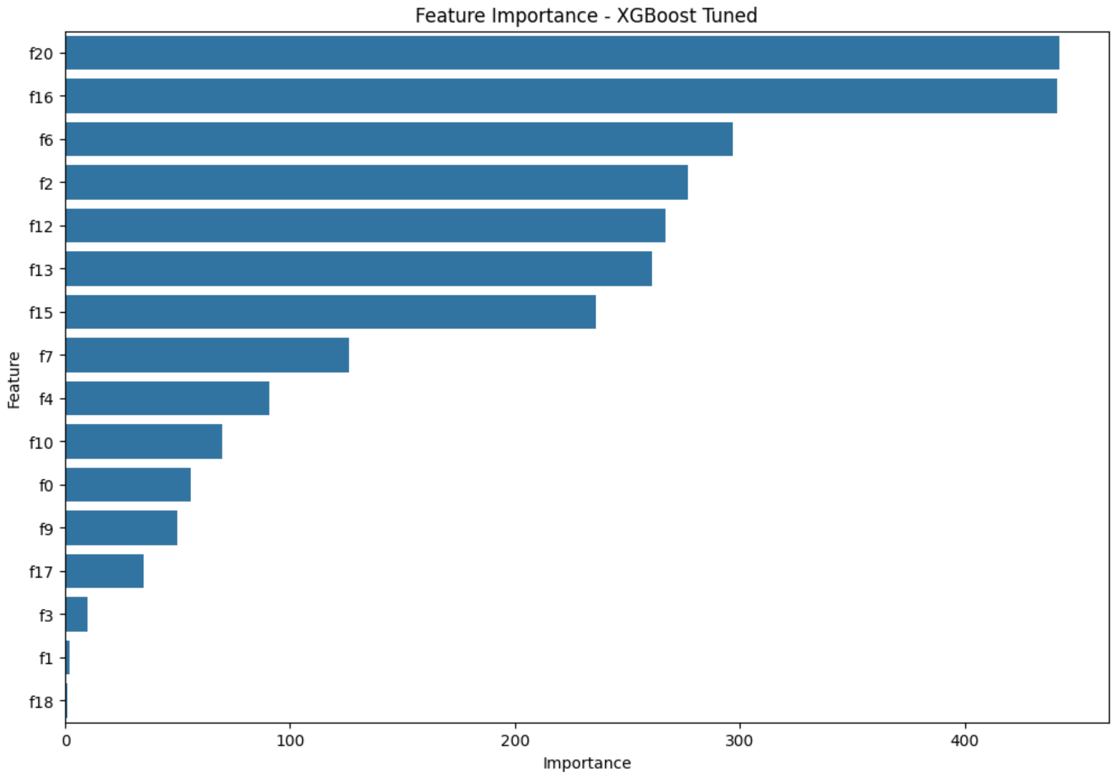
[20]:
```python
# Train XGBoost with the best parameters
xgb_best = xgb.XGBClassifier(
    learning_rate=0.01,
    max_depth=5,
    n_estimators=200,
    subsample=1.0,
    use_label_encoder=False,
    eval_metric='logloss'
)

# Scaling the data
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)

# Fit the model
xgb_best.fit(X_train_scaled, y_train)
xgb_best_pred = xgb_best.predict(X_val_scaled)
```

```
# Evaluate the tuned model
evaluate_model(y_val, xgb_best_pred, "XGBoost Tuned")
```

```
Model: XGBoost Tuned
Accuracy: 0.9774
Precision: 1.0000
Recall: 0.8451
F1 Score: 0.9160
ROC-AUC: 0.9225
```

[20]:
```
(0.977366255144033,
 1.0,
 0.8450704225352113,
 0.916030534351145,
 0.9225352112676056)
```

[21]:
```
# Feature importance for the tuned XGBoost model
xgb_importance = xgb_best.get_booster().get_score(importance_type='weight')
sorted_importance = sorted(xgb_importance.items(), key=lambda x: x[1],␣
 ↪reverse=True)

# Convert to a DataFrame for better visualization
importance_df = pd.DataFrame(sorted_importance, columns=['Feature',␣
 ↪'Importance'])

# Plot feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.title('Feature Importance - XGBoost Tuned')
plt.show()
```



[22]:

```python
# Combine training and validation sets for final model training
X_final_train = pd.concat([X_train, X_val])
y_final_train = pd.concat([y_train, y_val])

# Scale the combined training set
X_final_train_scaled = scaler.fit_transform(X_final_train)
X_test_scaled = scaler.transform(test_df.drop('Churn', axis=1))

# Train the final XGBoost model
xgb_final = xgb.XGBClassifier(
    learning_rate=0.01,
    max_depth=5,
    n_estimators=200,
    subsample=1.0,
    use_label_encoder=False,
    eval_metric='logloss'
)

# Fit the final model
xgb_final.fit(X_final_train_scaled, y_final_train)
xgb_final_pred = xgb_final.predict(X_test_scaled)
```

```python
# Evaluate the final model
y_test = test_df['Churn'].astype('int')
evaluate_model(y_test, xgb_final_pred, "XGBoost Final")
```

```
Model: XGBoost Final
Accuracy: 0.9802
Precision: 0.9865
Recall: 0.8690
F1 Score: 0.9241
ROC-AUC: 0.9336
```

[22]:
```
(0.9801652892561984,
 0.9864864864864865,
 0.8690476190476191,
 0.9240506329113924,
 0.9335641166255368)
```

```
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.9; total
```

```
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.9; total
```

```
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=1.0; total
```

```
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.9s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=1.0; total
```

```
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=1.0; total
```

```
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.9; total
```

```
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.8; total
```

```
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.8; total
```

```
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.8; total
```

```
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=1.0; total
```

```
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=1.0; total
```

```
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.8; total
```

```
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.9; total
```

```
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=1.0; total
```

```
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.9; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.8; total
```

```
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=200, subsample=0.9; total
time=   0.4s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.6s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.8s
[CV] END learning_rate=0.01, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.8s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=200, subsample=0.9; total
```

```
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=5, n_estimators=300, subsample=1.0; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=100, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.4s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.8; total
time=   0.5s
[CV] END learning_rate=0.1, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=0.8; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=100, subsample=1.0; total
time=   0.1s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=200, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=0.9; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=3, n_estimators=300, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=100, subsample=0.9; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.8; total
time=   0.4s
[CV] END learning_rate=0.2, max_depth=5, n_estimators=300, subsample=0.9; total
```

```
time=   0.4s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=0.8; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1.0; total
time=   0.2s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=0.8; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=200, subsample=1.0; total
time=   0.3s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=0.9; total
time=   0.5s
[CV] END learning_rate=0.2, max_depth=7, n_estimators=300, subsample=1.0; total
time=   0.5s
```

[ ]: