



```
In [162... import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_

import warnings
warnings.filterwarnings("ignore")

sns.set(style="whitegrid")
```

```
In [163... df = pd.read_csv("loan_detection.csv")
df.head()
```

```
Out[163...   age  campaign  pdays  previous  no_previous_contact  not_working  job_admi
```

0	56	1	999	0	1	0
1	57	1	999	0	1	0
2	37	1	999	0	1	0
3	40	1	999	0	1	0
4	56	1	999	0	1	0

5 rows × 60 columns

```
In [164... print("Shape:", df.shape)
print("\nMissing Values:\n", df.isnull().sum())
print("\nData Types:\n")
df.info()
```

Shape: (41188, 60)

Missing Values:

age	0
campaign	0
pdays	0
previous	0
no_previous_contact	0
not_working	0
job_admin.	0
job_blue-collar	0
job_entrepreneur	0
job_housemaid	0
job_management	0
job_retired	0
job_self-employed	0
job_services	0
job_student	0
job_technician	0
job_unemployed	0
job_unknown	0
marital_divorced	0
marital_married	0
marital_single	0
marital_unknown	0
education_basic.4y	0
education_basic.6y	0
education_basic.9y	0
education_high.school	0
education_illiterate	0
education_professional.course	0
education_university.degree	0
education_unknown	0
default_no	0
default_unknown	0
default_yes	0
housing_no	0
housing_unknown	0
housing_yes	0
loan_no	0
loan_unknown	0
loan_yes	0
contact_cellular	0
contact_telephone	0
month_apr	0
month_aug	0
month_dec	0
month_jul	0
month_jun	0
month_mar	0
month_may	0
month_nov	0
month_oct	0
month_sep	0

```

day_of_week_fri      0
day_of_week_mon      0
day_of_week_thu      0
day_of_week_tue      0
day_of_week_wed      0
poutcome_failure     0
poutcome_nonexistent 0
poutcome_success     0
Loan_Status_label    0
dtype: int64

```

Data Types:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 60 columns):

```

#	Column	Non-Null Count	Dtype
0	age	41188 non-null	int64
1	campaign	41188 non-null	int64
2	pdays	41188 non-null	int64
3	previous	41188 non-null	int64
4	no_previous_contact	41188 non-null	int64
5	not_working	41188 non-null	int64
6	job_admin.	41188 non-null	int64
7	job_blue-collar	41188 non-null	int64
8	job_entrepreneur	41188 non-null	int64
9	job_housemaid	41188 non-null	int64
10	job_management	41188 non-null	int64
11	job_retired	41188 non-null	int64
12	job_self-employed	41188 non-null	int64
13	job_services	41188 non-null	int64
14	job_student	41188 non-null	int64
15	job_technician	41188 non-null	int64
16	job_unemployed	41188 non-null	int64
17	job_unknown	41188 non-null	int64
18	marital_divorced	41188 non-null	int64
19	marital_married	41188 non-null	int64
20	marital_single	41188 non-null	int64
21	marital_unknown	41188 non-null	int64
22	education_basic.4y	41188 non-null	int64
23	education_basic.6y	41188 non-null	int64
24	education_basic.9y	41188 non-null	int64
25	education_high.school	41188 non-null	int64
26	education_illiterate	41188 non-null	int64
27	education_professional.course	41188 non-null	int64
28	education_university.degree	41188 non-null	int64
29	education_unknown	41188 non-null	int64
30	default_no	41188 non-null	int64
31	default_unknown	41188 non-null	int64
32	default_yes	41188 non-null	int64
33	housing_no	41188 non-null	int64
34	housing_unknown	41188 non-null	int64
35	housing_yes	41188 non-null	int64

```

36 loan_no 41188 non-null int64
37 loan_unknown 41188 non-null int64
38 loan_yes 41188 non-null int64
39 contact_cellular 41188 non-null int64
40 contact_telephone 41188 non-null int64
41 month_apr 41188 non-null int64
42 month_aug 41188 non-null int64
43 month_dec 41188 non-null int64
44 month_jul 41188 non-null int64
45 month_jun 41188 non-null int64
46 month_mar 41188 non-null int64
47 month_may 41188 non-null int64
48 month_nov 41188 non-null int64
49 month_oct 41188 non-null int64
50 month_sep 41188 non-null int64
51 day_of_week_fri 41188 non-null int64
52 day_of_week_mon 41188 non-null int64
53 day_of_week_thu 41188 non-null int64
54 day_of_week_tue 41188 non-null int64
55 day_of_week_wed 41188 non-null int64
56 poutcome_failure 41188 non-null int64
57 poutcome_nonexistent 41188 non-null int64
58 poutcome_success 41188 non-null int64
59 Loan_Status_label 41188 non-null int64
dtypes: int64(60)
memory usage: 18.9 MB

```

```

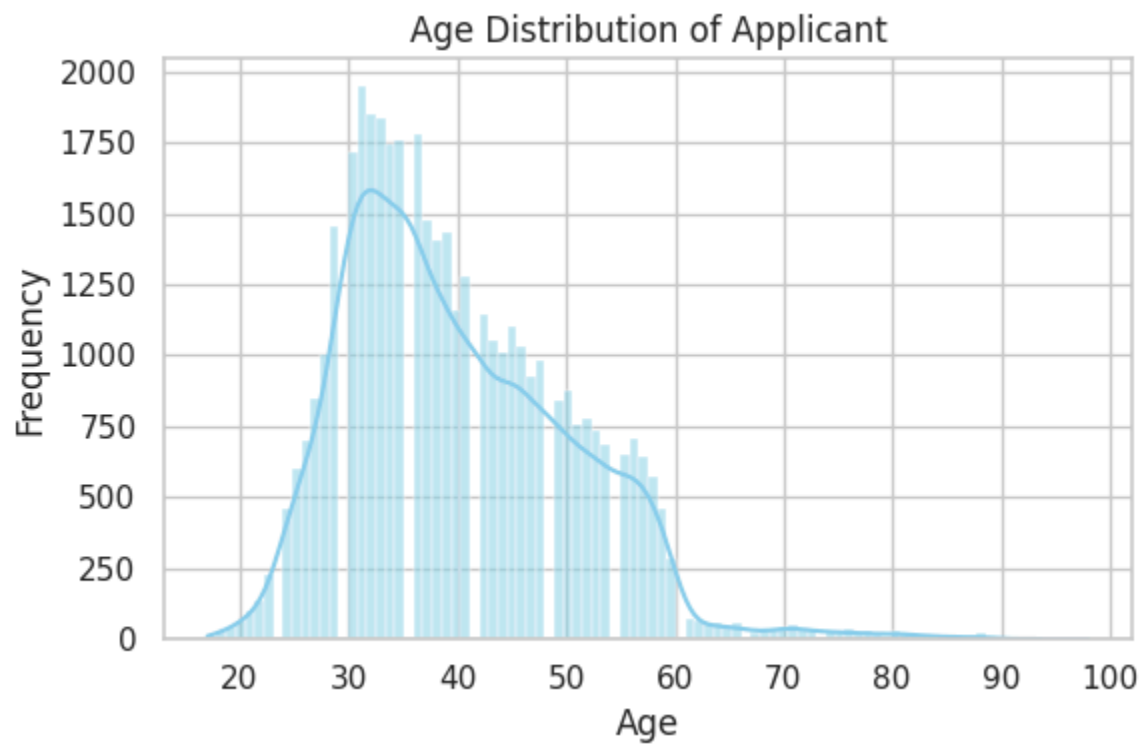
In [165... plt.figure(figsize=(6,4))
sns.countplot(x='Loan_Status_label', data=df, palette='viridis')
plt.title("Loan Status Distributions")
plt.xlabel("Loan Ststus")
plt.ylabel("Count")
plt.tight_layout()
plt.show()

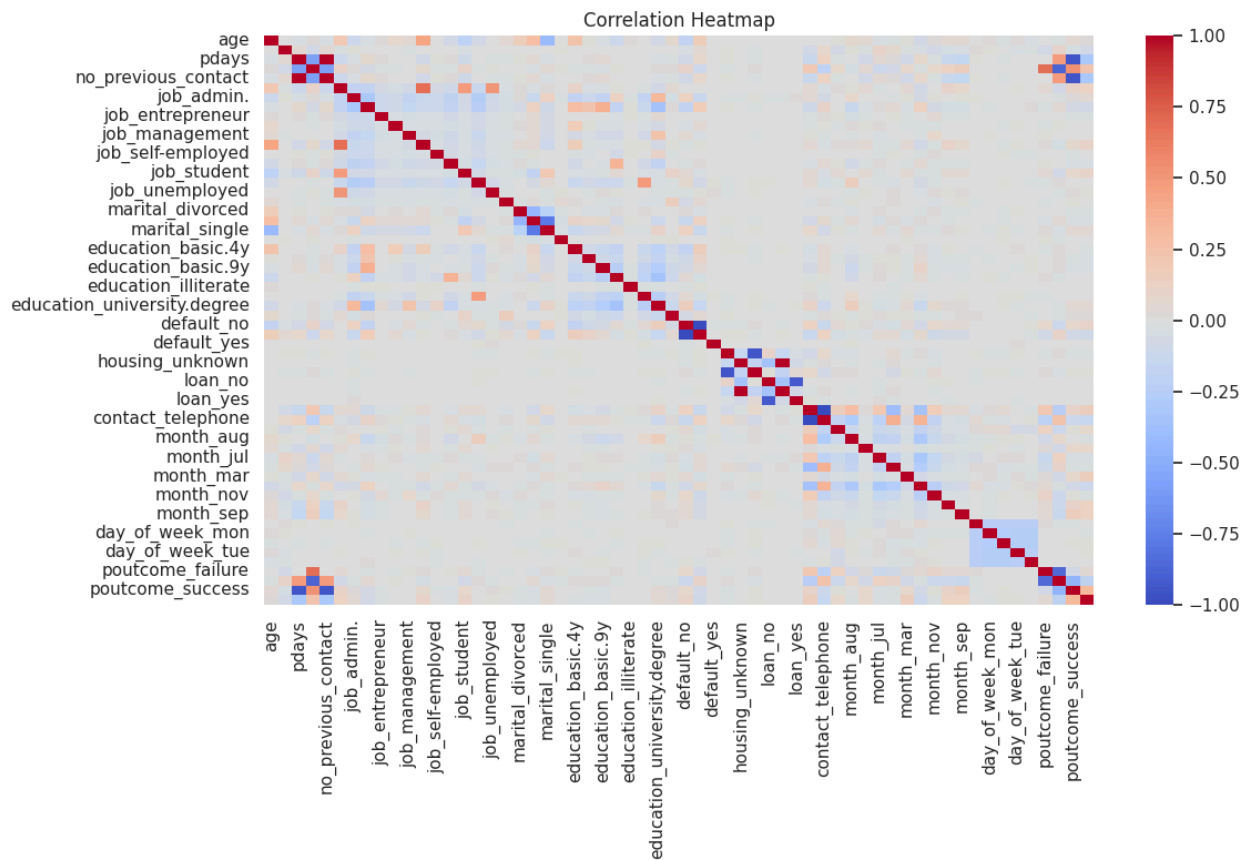
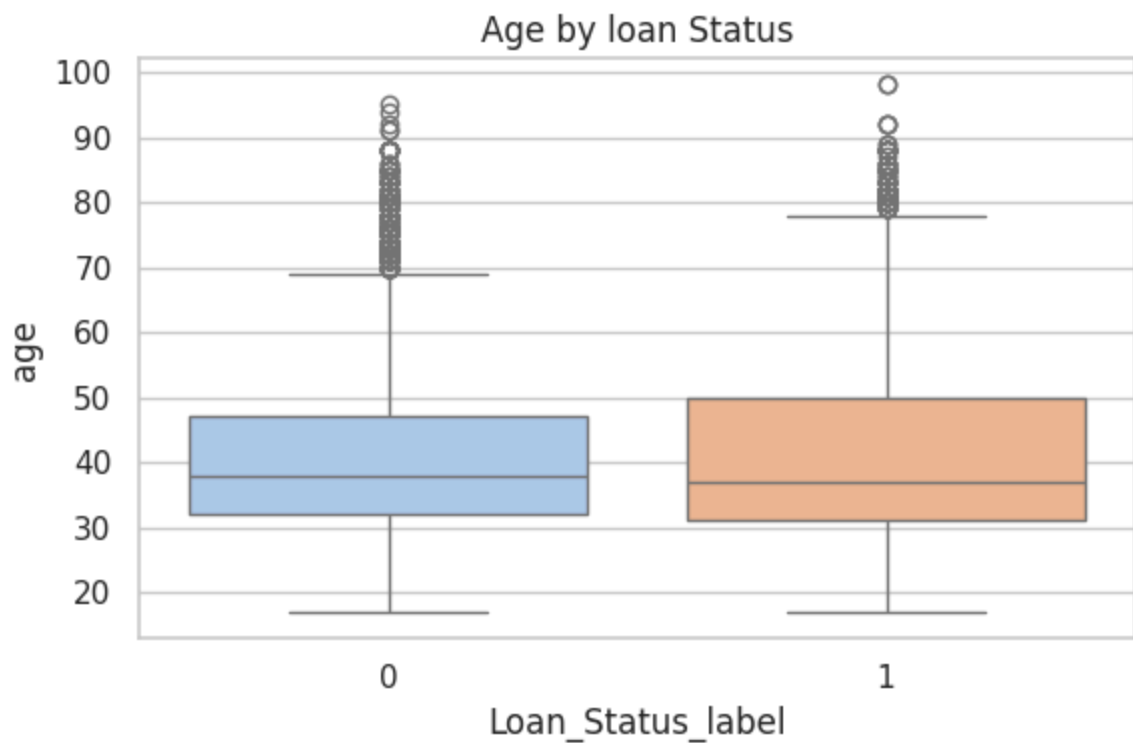
plt.figure(figsize=(6,4))
sns.histplot(df['age'], kde=True, color='skyblue')
plt.title("Age Distribution of Applicant")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()

plt.figure(figsize=(6,4))
sns.boxplot(x='Loan_Status_label', y='age', data=df, palette='pastel')
plt.title("Age by loan Status")
plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), cmap='coolwarm', annot=False)
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()

```





```
In [166... target_col = 'Loan_Status_label'
X = df.drop(columns=[target_col])
y = df[target_col]
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

```
In [167... scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
In [168... rf = RandomForestClassifier(random_state=42)
rf.fit(X_train_scaled, y_train)
y_pred_rf = rf.predict(X_test_scaled)

lr = LogisticRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)
```

```
In [169... def evaluate_model(name, y_test, y_pred):
    print(f"\n--- {name} ---")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Classifications Report:\n", classification_report(y_test, y_pred))
    print("Confusion Matrix :\n", confusion_matrix(y_test, y_pred))

evaluate_model("Random Forest", y_test, y_pred_rf)
evaluate_model("Logistic Regressions", y_test, y_pred_lr)
```

```

--- Random Forest ---
Accuracy: 0.8868657441126487
Classifications Report:

```

	precision	recall	f1-score	support
0	0.91	0.97	0.94	7303
1	0.50	0.25	0.34	935
accuracy			0.89	8238
macro avg	0.71	0.61	0.64	8238
weighted avg	0.86	0.89	0.87	8238

```

Confusion Matrix :
[[7070  233]
 [ 699 236]]

```

```

--- Logistic Regressions ---
Accuracy: 0.8957271182325808
Classifications Report:

```

	precision	recall	f1-score	support
0	0.90	0.99	0.94	7303
1	0.64	0.19	0.29	935
accuracy			0.90	8238
macro avg	0.77	0.59	0.62	8238
weighted avg	0.87	0.90	0.87	8238

```

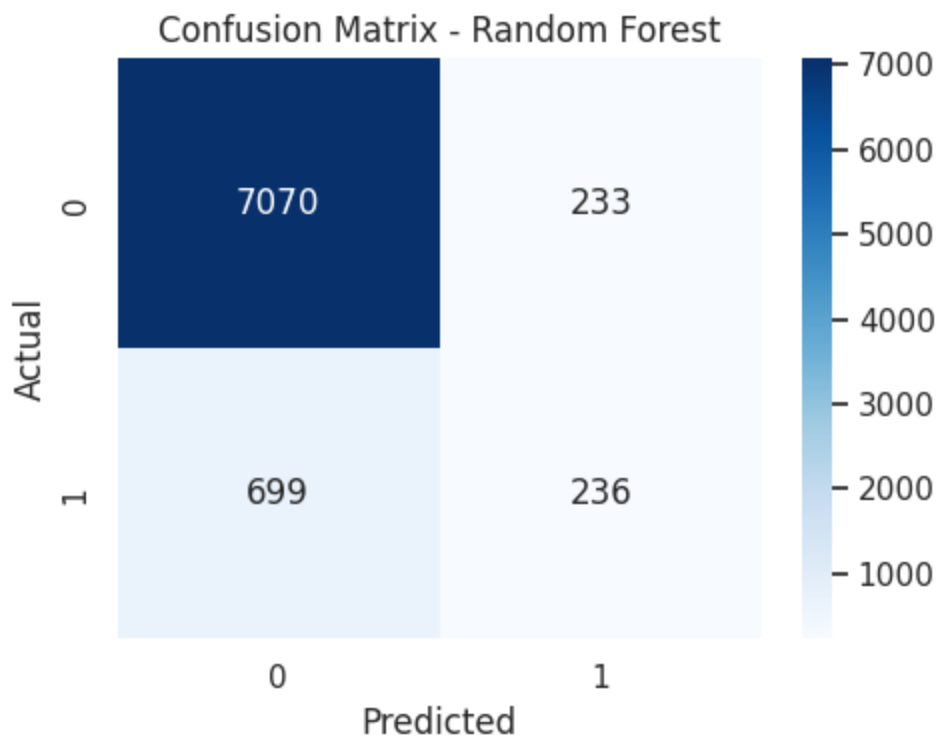
Confusion Matrix :
[[7203  100]
 [ 759 176]]

```

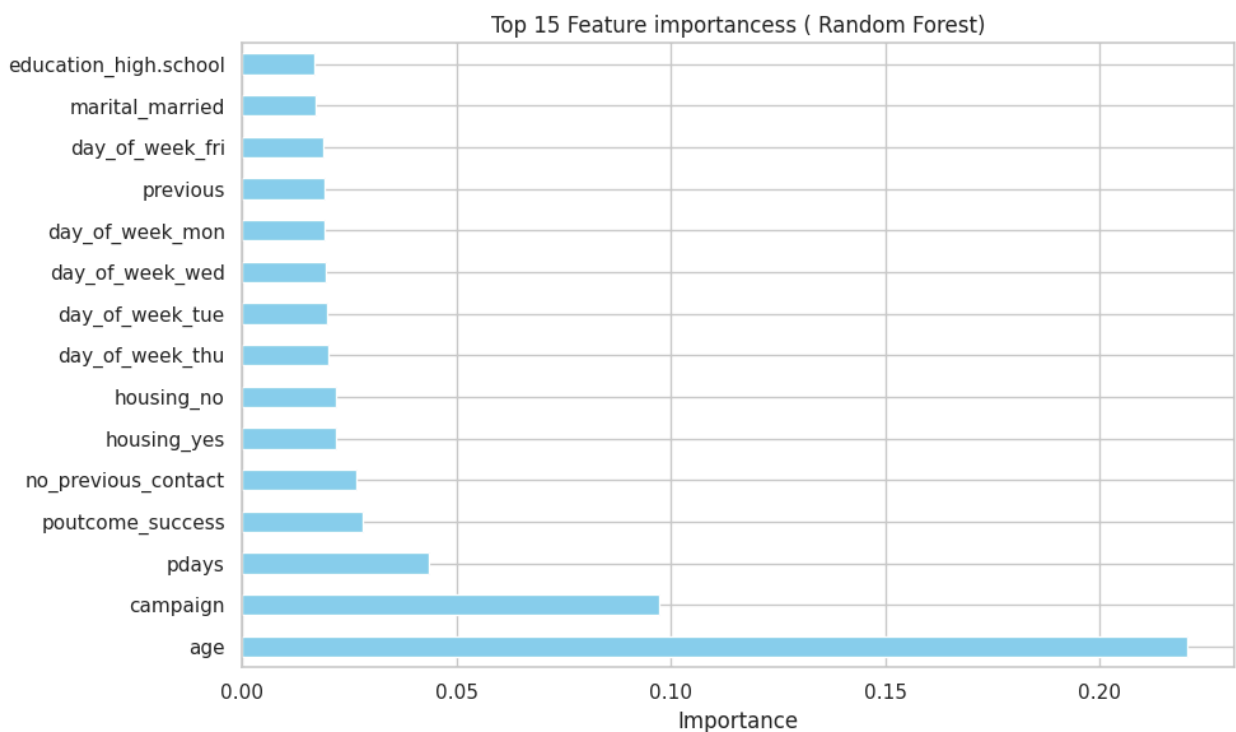
```

In [170... plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred_rf), annot=True, fmt='d', cmap='Bl
plt.title("Confusion Matrix - Random Forest")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.show()

```

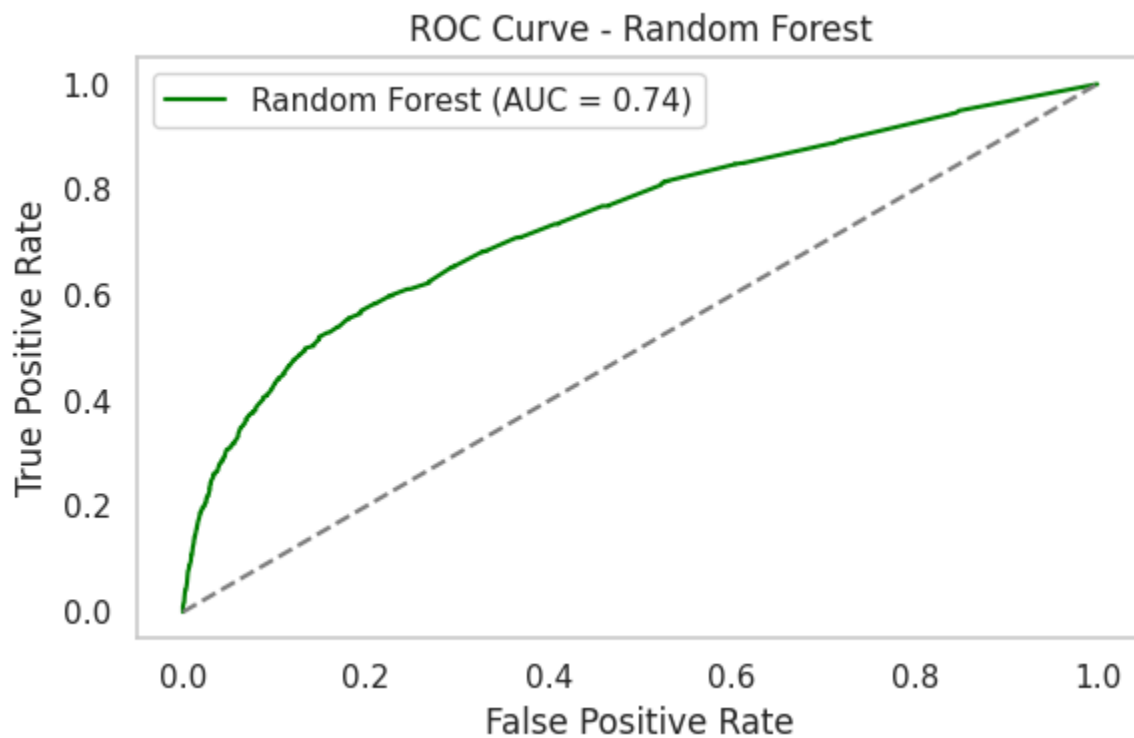



```
In [171]: importances = pd.Series(rf.feature_importances_, index=X.columns)
plt.figure(figsize=(10,6))
importances.nlargest(15).plot(kind='barh', color='skyblue')
plt.title("Top 15 Feature importances ( Random Forest)")
plt.xlabel("Importance")
plt.tight_layout()
plt.show()
```



```
In [172... y_proba = rf.predict_proba(X_test_scaled)[: , 1]
fpr, tpr, thresholds = roc_curve(y_test, y_proba)

plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, label='Random Forest (AUC = {:.2f})'.format(roc_auc_score(y
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Random Forest')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```



In []:

In []:

In []:

In []: