Notebook

September 23, 2023

```python
[7]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     %matplotlib inline
     import seaborn as sns
     import scipy as sp
     from tabulate import tabulate
     import random
     import tensorflow as tf
```

```python
[3]: df = pd.read_csv("/content/drive/MyDrive/DataSet02/OnlinePayment/onlinefraud.
       ↪csv")
     df.head()
```

```
[3]:    step      type     amount      nameOrig  oldbalanceOrg  newbalanceOrig  \
     0     1   PAYMENT    9839.64   C1231006815       170136.0       160296.36
     1     1   PAYMENT    1864.28   C1666544295        21249.0        19384.72
     2     1  TRANSFER     181.00   C1305486145          181.0            0.00
     3     1  CASH_OUT     181.00    C840083671          181.0            0.00
     4     1   PAYMENT   11668.14   C2048537720        41554.0        29885.86

            nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
     0  M1979787155             0.0             0.0        0               0
     1  M2044282225             0.0             0.0        0               0
     2   C553264065             0.0             0.0        1               0
     3    C38997010         21182.0             0.0        1               0
     4  M1230701703             0.0             0.0        0               0
```

```python
[4]: df.isnull().sum()
```

```
[4]: step              0
     type              0
     amount            0
     nameOrig          0
     oldbalanceOrg     0
     newbalanceOrig    0
     nameDest          0
```

```
oldbalanceDest      0
newbalanceDest      0
isFraud             0
isFlaggedFraud      0
dtype: int64
```

1 About the dataset

step: represents a unit of time where 1 step equals 1 hour

type: type of online transaction

amount: the amount of the transaction

nameOrig: customer starting the transaction

oldbalanceOrg: balance before the transaction

newbalanceOrig: balance after the transaction

nameDest: recipient of the transaction

oldbalanceDest: initial balance of recipient before the transaction

newbalanceDest: the new balance of recipient after the transaction

isFraud: fraud transaction

[6]: ```df.shape```

[6]: (6362620, 11)

[8]: ```df.info()```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

```
[9]: df.sample(5)
```

```
[9]:          step       type      amount      nameOrig  oldbalanceOrg  \
     4465633   324    CASH_IN    30109.40  C1800129586      273692.74
     4694381   331    PAYMENT    16771.37  C1901092388      103889.00
     4865912   348   CASH_OUT   233049.11  C1837939595           0.00
     5650350   396    PAYMENT    43589.49   C768081401           0.00
     4372461   310    CASH_IN   217352.69   C622621127      392930.32

              newbalanceOrig      nameDest  oldbalanceDest  newbalanceDest  isFraud  \
     4465633        303802.14  C109583746       100660.29        70550.89        0
     4694381         87117.63  M1721743274           0.00            0.00        0
     4865912             0.00  C1763305509      4975075.23      5208124.34        0
     5650350             0.00   M831342188           0.00            0.00        0
     4372461        610283.01  C1678099202       853292.61       635939.92        0

              isFlaggedFraud
     4465633               0
     4694381               0
     4865912               0
     5650350               0
     4372461               0
```

```
[10]: fraud_min_max = [
          ['amount', df.amount.min(), df.amount.max()],
          ['oldbalanceOrg', df.oldbalanceOrg.min(), df.oldbalanceOrg.max()],
          ['newbalanceOrig', df.newbalanceOrig.min(), df.newbalanceOrig.max()],
          ['oldbalanceDest', df.oldbalanceDest.min(), df.oldbalanceDest.max()],
          ['isFraud', df.isFraud.min(), df.isFraud.max()]
      ]

      print(
          tabulate(
              fraud_min_max,
              headers=['columns', 'min value', 'max value'],
              showindex=True,
              tablefmt='github',
              numalign='right'
          )
      )
```

	columns	min value	max value
0	amount	0	9.24455e+07
1	oldbalanceOrg	0	5.9585e+07
2	newbalanceOrig	0	4.9585e+07
3	oldbalanceDest	0	3.56016e+08

```
| 4 | isFraud        |                0 |                1 |
```

```python
[11]: # Downcast numerical columns with smaller dtype
      for col in df.columns:
          if df[col].dtype == 'float64':
              df[col] = pd.to_numeric(df[col], downcast='float')
          if df[col].dtype == 'int64':
              df[col] = pd.to_numeric(df[col], downcast='unsigned')

      # Use category dtype for categorical column
      df['type'] = df['type'].astype('category')
```

```python
[12]: # Check duplicate values
      df.duplicated().sum()
```

```
[12]: 0
```

2   Univariate data visualization

```python
[13]: df['step'].value_counts()
```

```
[13]: 19      51352
      18      49579
      187     49083
      235     47491
      307     46968
              ...
      432         4
      706         4
      693         4
      112         2
      662         2
      Name: step, Length: 743, dtype: int64
```

There are 743 steps, and every step has at least 2 occurrences.

```python
[14]: ax = sns.countplot(x='type', data=df, palette='PuBu')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Count plot of transaction type')
      plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
      plt.ylabel('Number of transactions')
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note
that artists whose label start with an underscore are ignored when legend() is
called with no argument.
```
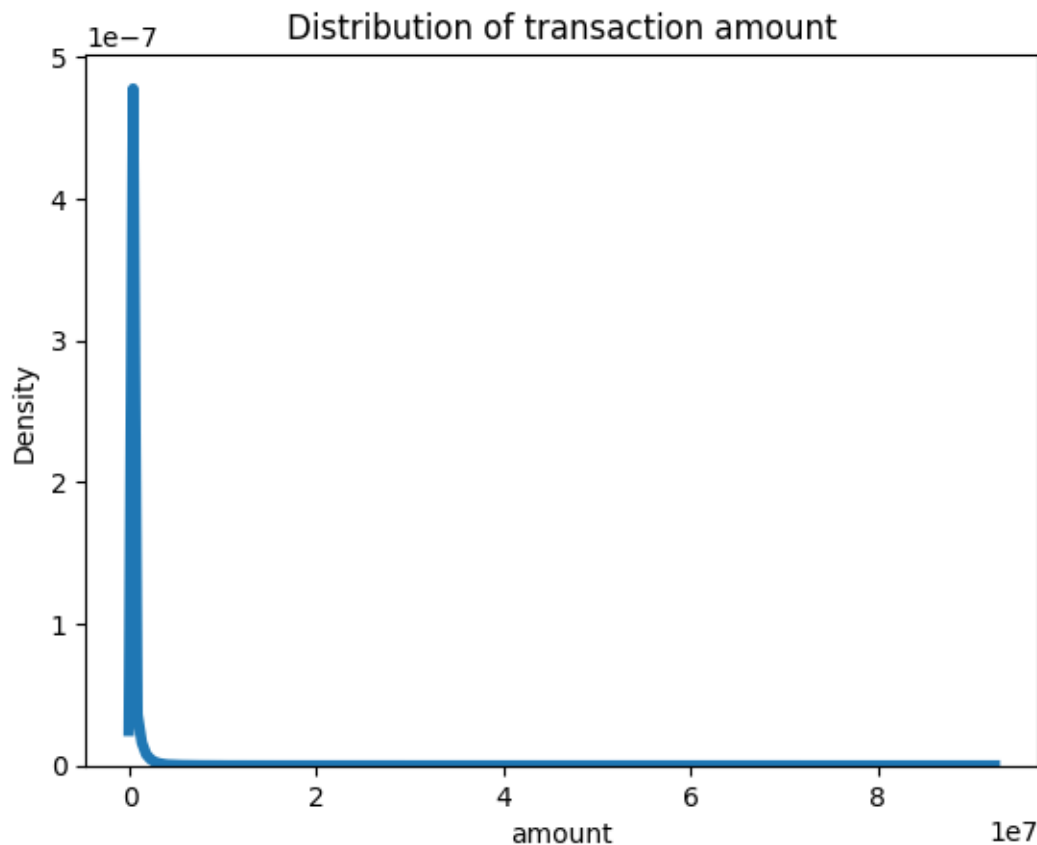
[14]: Text(0, 0.5, 'Number of transactions')

Count plot of transaction type
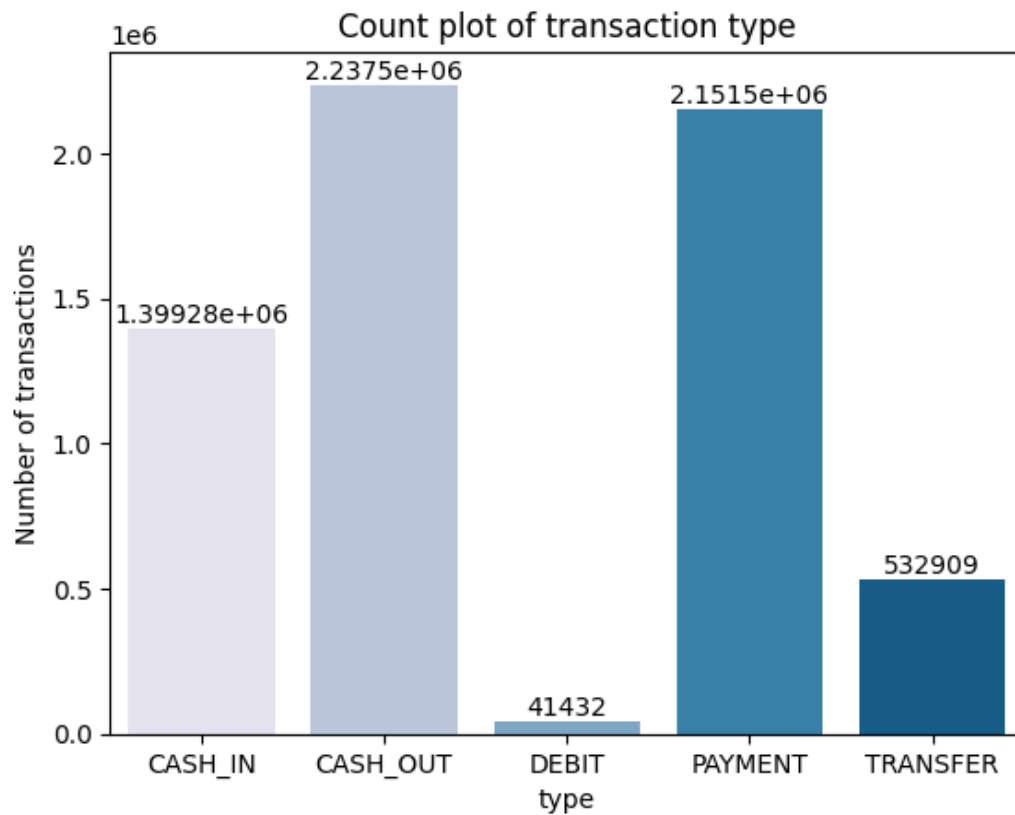
Cash out is the most numerous transaction type, followed by payment, cash in, transfer and debit types.

[15]:
```
sns.kdeplot(df['amount'], linewidth=4)
plt.title('Distribution of transaction amount')
```

[15]: Text(0.5, 1.0, 'Distribution of transaction amount')

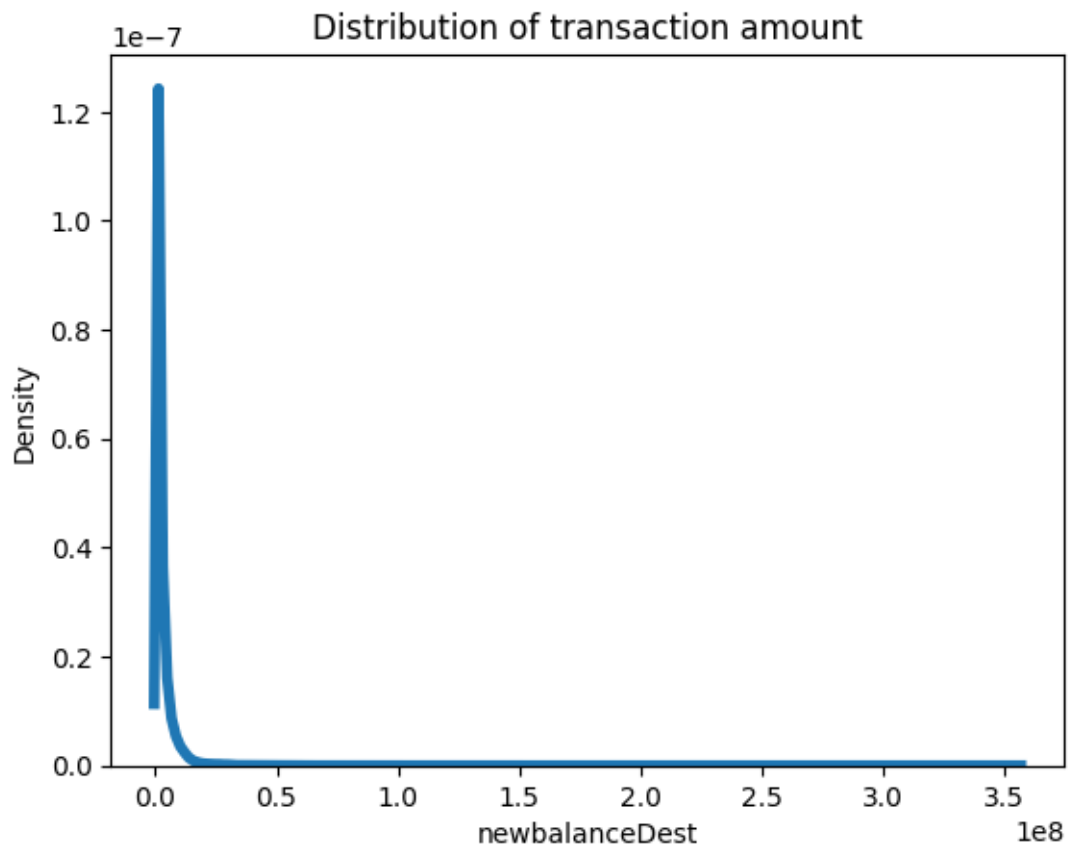The distribution of transaction amounts is right skewed. This indicates that most values are clustered around the left tail of the distribution, with the longer right tail. (mode < median < mean)

```
[16]: df['nameOrig'].value_counts()
```

```
[16]: C1902386530    3
      C363736674     3
      C545315117     3
      C724452879     3
      C1784010646    3
                    ..
      C98968405      1
      C720209255     1
      C1567523029    1
      C644777639     1
      C1280323807    1
      Name: nameOrig, Length: 6353307, dtype: int64
```

There are 2722362 recipients, and every step has at least 1 occurrence.

```
[17]: sns.kdeplot(df['oldbalanceDest'], linewidth=4)
      plt.title('Distribution of transaction amount')
```

[17]: Text(0.5, 1.0, 'Distribution of transaction amount')
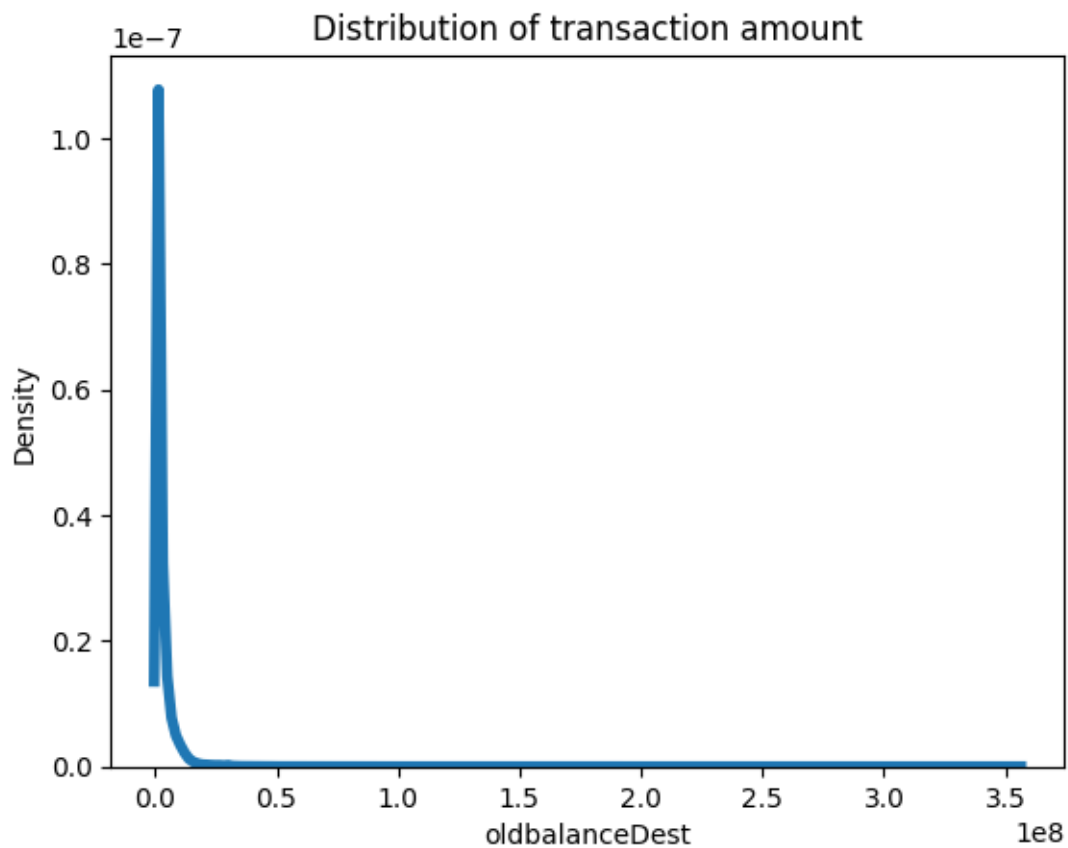
The distribution of pre-transaction balances of the recipient is right skewed.

```
[18]: sns.kdeplot(df['newbalanceDest'], linewidth=4)
      plt.title('Distribution of transaction amount')
```

[18]: Text(0.5, 1.0, 'Distribution of transaction amount')

The distribution of post-transaction balances of the recipient is right skewed.

```
[19]: ax = sns.countplot(x='isFraud', data=df, palette='PuBu')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Count plot of fraud transaction')
      plt.ylabel('Number of transactions')

      del ax
```

There are much more non-fraudulent transactions than fraudulent transactions.

3 Bivariate data visualization

```
[20]: fig, ax = plt.subplots(1,2,figsize=(20,5))

      sns.countplot(x='type', data=df, hue='isFraud', palette='PuBu', ax=ax[0])
      for container in ax[0].containers:
          ax[0].bar_label(container)
      ax[0].set_title('Count plot of transaction type')
      ax[0].legend(loc='best')
      ax[0].set_ylabel('Number of transactions')

      df2 = df.groupby(['type', 'isFraud']).size().unstack()
      df2.apply(lambda x : round(x/sum(x)*100, 2), axis=1).plot(kind='barh',␣
       ↪stacked=True, color=['lightsteelblue', 'steelblue'], ax=ax[1])
      for container in ax[1].containers:
          ax[1].bar_label(container, label_type='center')
      ax[1].set_title('Count plot of transaction type')
```

```
ax[1].legend(bbox_to_anchor=(1.05,1), loc='upper left')
ax[1].set_ylabel('Number of transactions')
ax[1].grid(axis='y')
```
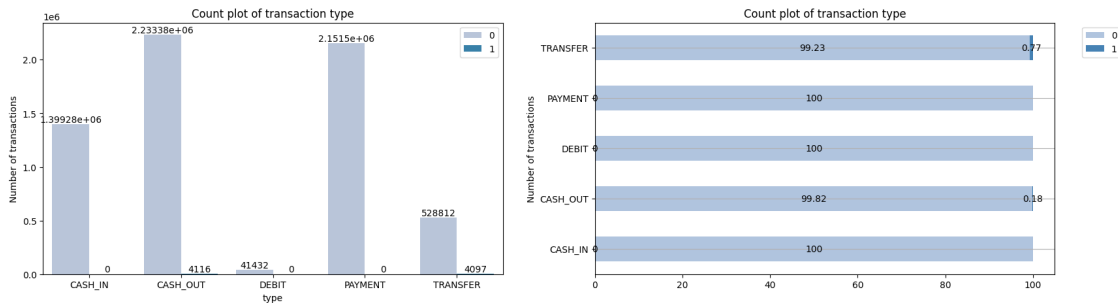

Fraudulent transactions only occur in debit and transfer types.

```
[21]: df['quantity'] = pd.cut(df['amount'], 5, labels=['very low', 'low', 'moderate',␣
      ↪'high', 'very high'])

      ax = sns.countplot(x='quantity', data=df, hue='isFraud', palette='PuBu')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Count plot of amount quantity')
      plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
      plt.ylabel('Number of transactions')
```

```
[21]: Text(0, 0.5, 'Number of transactions')
```

Count plot of amount quantity
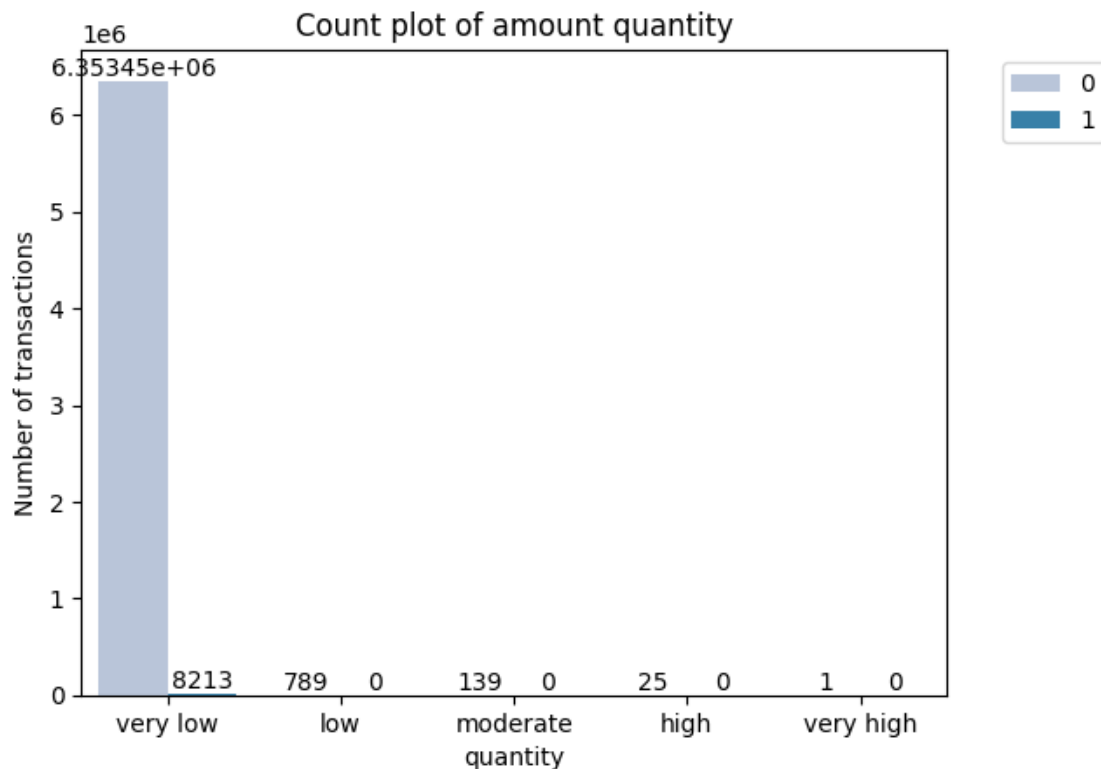
All fraudulent transactions fall into the category of very low amounts. This suggests that in most cases, small transactions are more prone to fraudulent transactions.

```
[22]: df1 = df[df['isFraud']==1]
      df2 = df1['step'].value_counts().head(10)
      ax = df2.plot(kind='bar', color='lightsteelblue')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Top 10 steps that often lead to fraudulent transactions')
      plt.ylabel('Number of fraudulent transactions')
      plt.xlabel('Step')
      plt.grid(axis='x')

      del ax, df2
```

Top 10 steps that often lead to fraudulent transactions
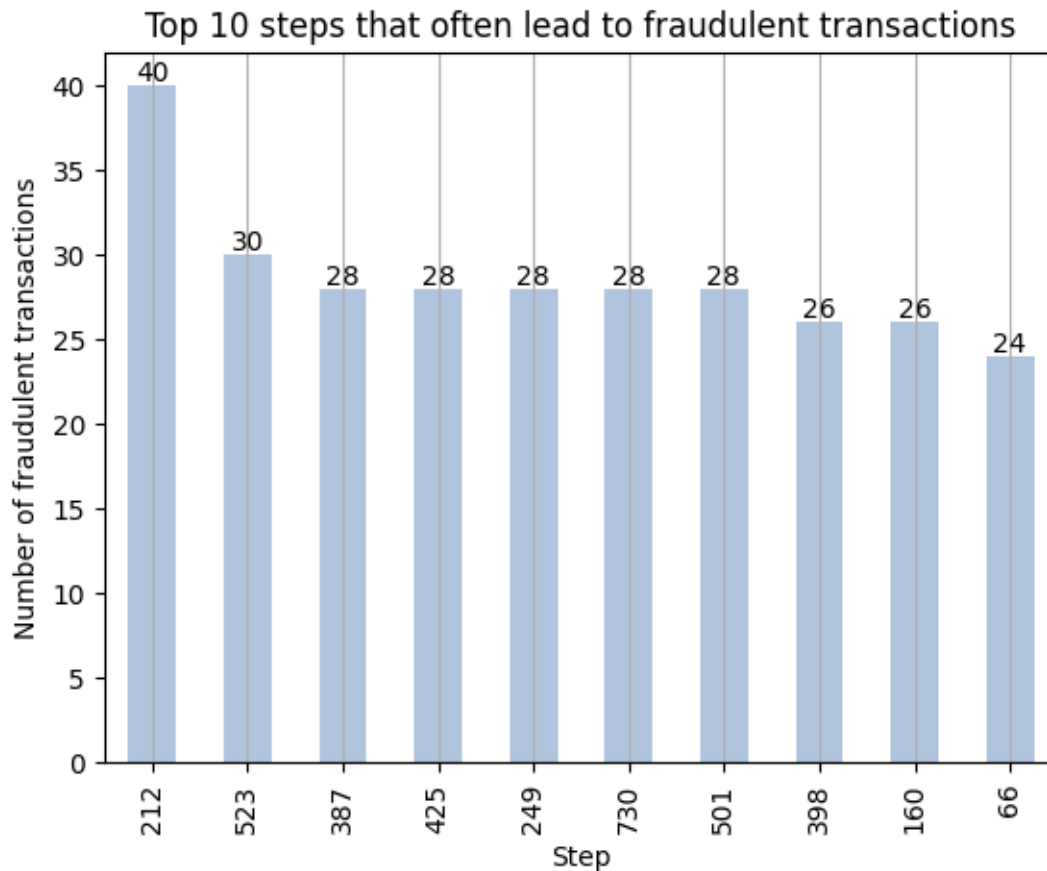
Step 212 has the highest number of fraudulent transactions, 40 cases. This indicates that Step 212 is the step that will most likely lead to fraudulent transactions.

```
[23]: df['oldbalanceOrg_amt'] = pd.cut(df['oldbalanceOrg'], 5, labels=['very low',␣
      ↪'low', 'moderate', 'high', 'very high'])

      ax = sns.countplot(x='oldbalanceOrg_amt', data=df, hue='isFraud',␣
      ↪palette='PuBu')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Count plot of initial customers pre-transaction balance amount')
      plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
      plt.ylabel('Number of transactions')
      plt.xlabel('Initial customers pre-transaction balance amount')
```

```
[23]: Text(0.5, 0, 'Initial customers pre-transaction balance amount')
```
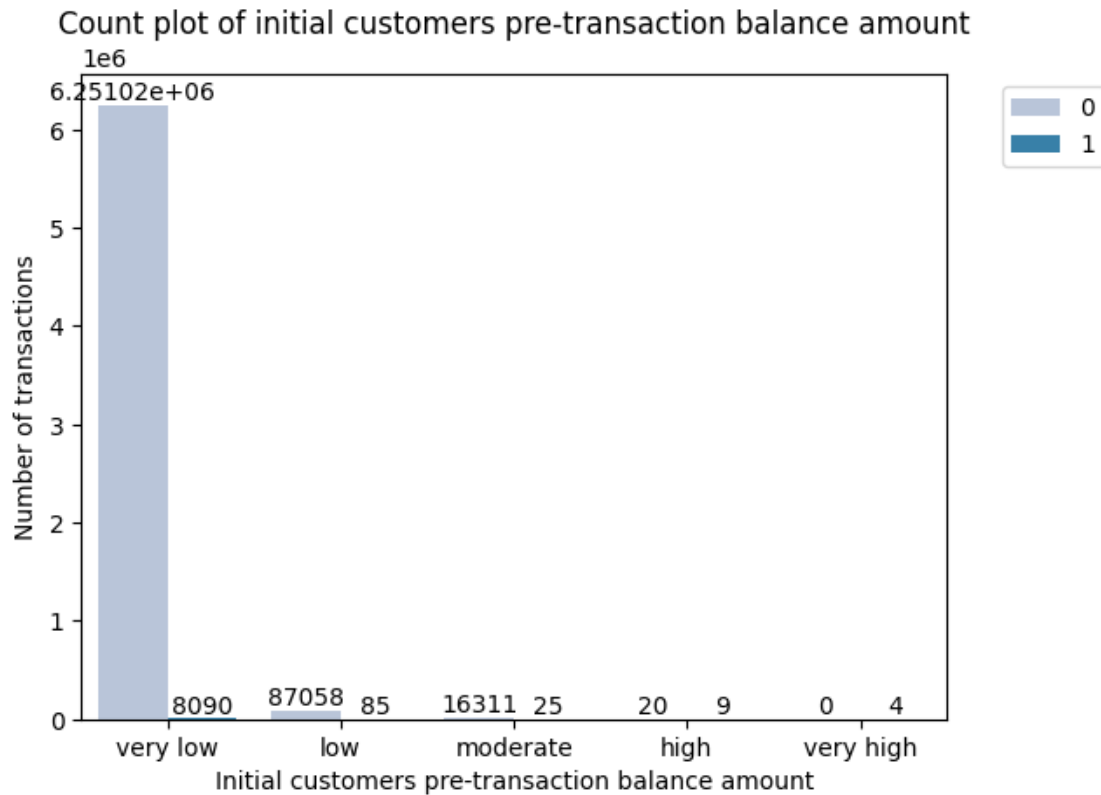
Count plot of initial customers pre-transaction balance amount

Initial customers with very low pre-transaction balances has the highest number of fraudulent transactions. This means that initial customers with very low pre-transaction balances may be more likely to fall for a fraudulent transaction.

```
[24]: df2 = df1['nameOrig'].value_counts().head(10)
      ax = df2.plot(kind='bar', color='lightsteelblue')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Top 10 initial customers fall for a fraudulent transactions')
      plt.ylabel('Number of fraudulent transactions')
      plt.xlabel('Initial customers')
      plt.grid(axis='x')

      del ax, df2
```

Top 10 initial customers fall for a fraudulent transactions
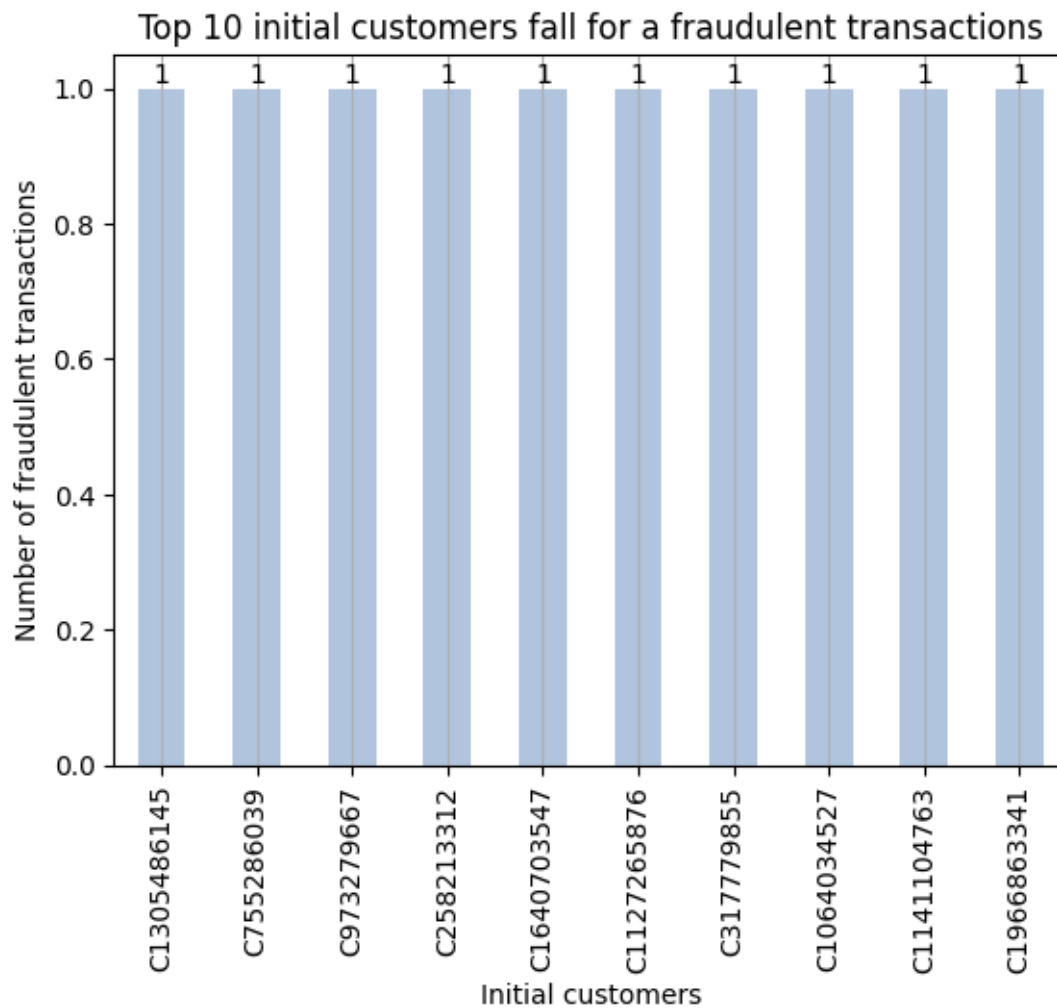
Initial customers are scammed of at most one for a fraudulent transaction.

```
[25]: df2 = df1['nameDest'].value_counts().head(10)
      ax = df2.plot(kind='bar', color='lightsteelblue')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Top 10 recipients highly involved in fraudulent transactions')
      plt.ylabel('Number of fraudulent transactions')
      plt.xlabel('Recipients')
      plt.grid(axis='x')

      del ax, df2
```

## Top 10 recipients highly involved in fraudulent transactions
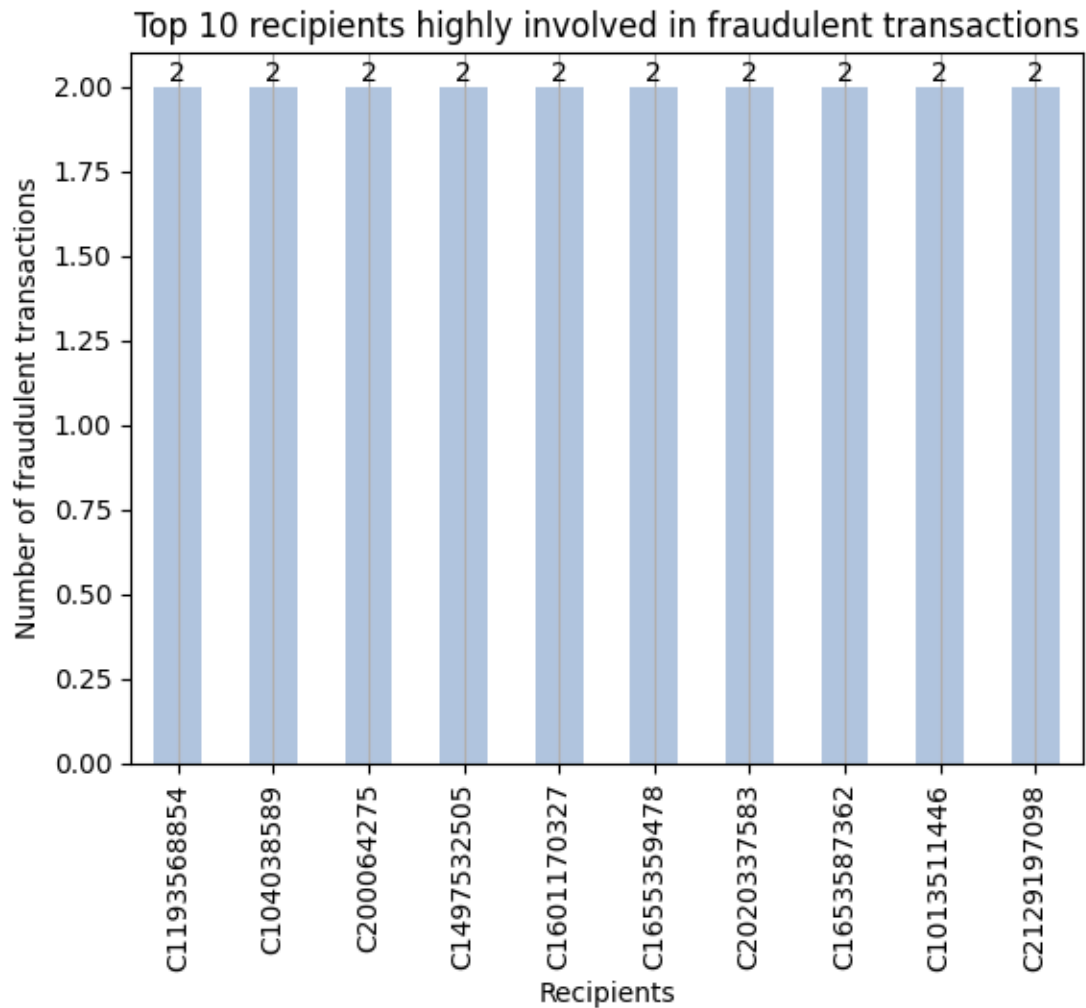
Suspicious recipients only have a maximum of 2 involved fraudulent transactions.

```
[26]: df['oldbalanceDest_amt'] = pd.cut(df['oldbalanceDest'], 5, labels=['very low',
      →'low', 'moderate', 'high', 'very high'])

      ax = sns.countplot(x='oldbalanceDest_amt', data=df, hue='isFraud',
      →palette='PuBu')
      for container in ax.containers:
          ax.bar_label(container)
      plt.title('Count plot of recipients pre-transaction balance amount')
      plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
      plt.ylabel('Number of transactions')
      plt.xlabel('Recipient pre-transaction balance amount')
```

```
[26]: Text(0.5, 0, 'Recipient pre-transaction balance amount')
```
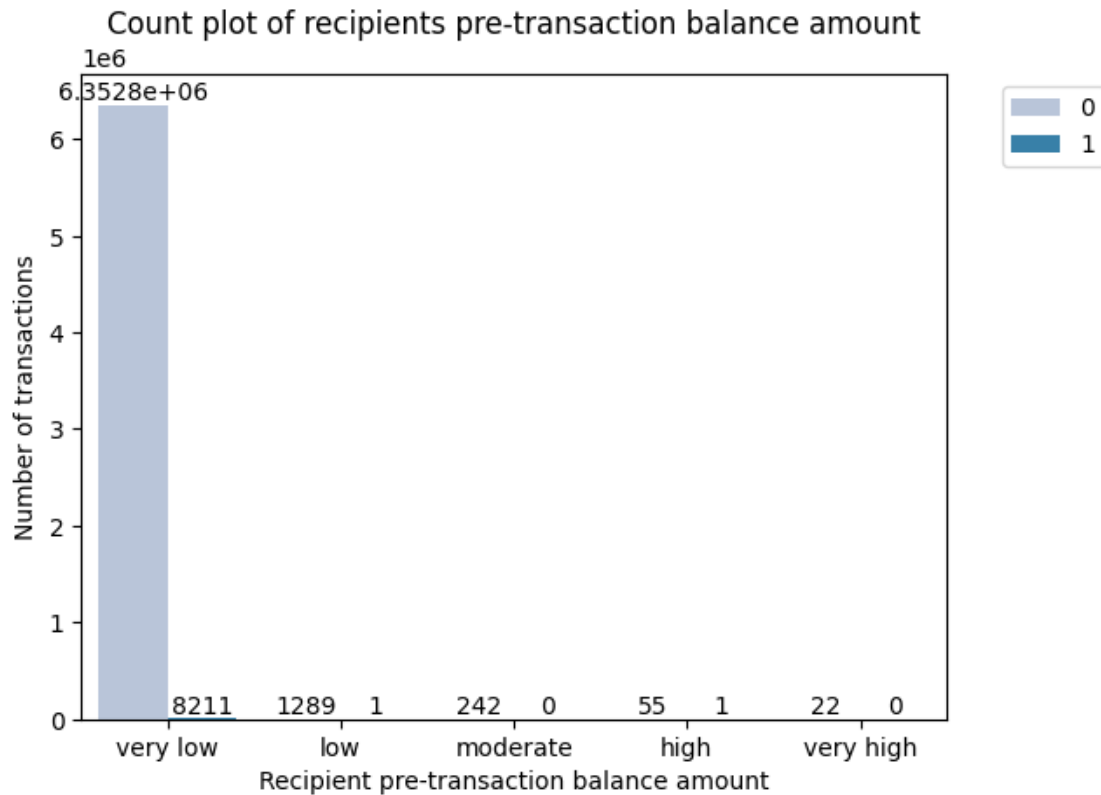
Count plot of recipients pre-transaction balance amount

Recipients with very low pre-transaction balances has the highest number of fraudulent transactions. This implies that recipients with very low pre-transaction balances may be more susceptible to fraudulent transactions.

4 Multivariate data visualization

```
[27]: corr_matrix = df.corr('spearman')
      sns.heatmap(corr_matrix, cbar=True, annot=True, mask = np.triu(np.
       ↪ones_like(corr_matrix, dtype = bool)), fmt='.3f', cmap='PuBu')
      plt.title('Correlation')
```

```
<ipython-input-27-f0e0de9688db>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr_matrix = df.corr('spearman')
```

```
[27]: Text(0.5, 1.0, 'Correlation')
```
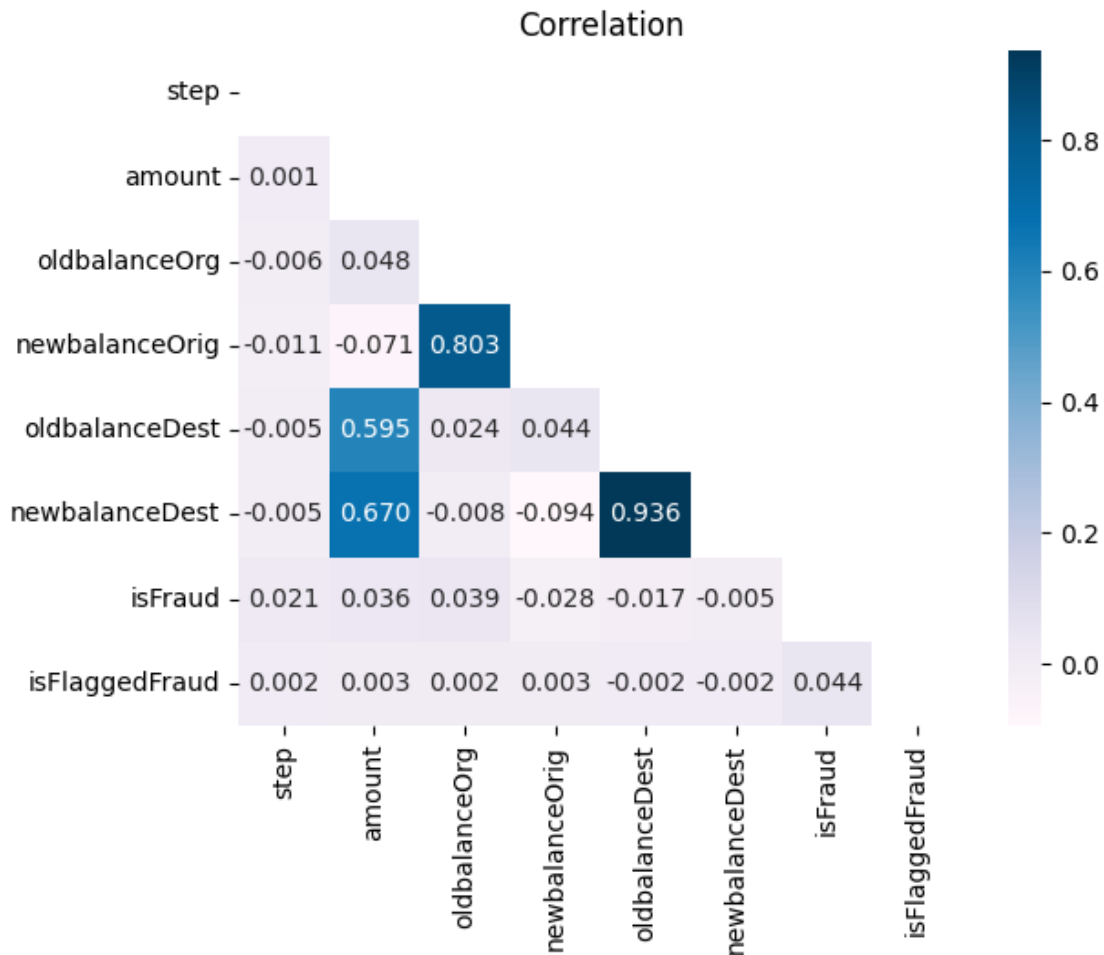
Correlation

oldbalanceOrg and newbalanceOrig has strong positive relationship.

oldbalanceDest and newbalanceDest has strong positive relationship.

oldbalanceOrg and amount has weak positive relationship.

newbalanceOrig and amount has moderate positive relationship.

5 Model Building

```
[28]:  # Data preprocessing
       df['type'] = df['type'].map({'PAYMENT':0, 'CASH_IN':1, 'DEBIT':2, 'CASH_OUT':3,
        ↪'TRANSFER':4})
```

```
[29]:  from sklearn.model_selection import StratifiedKFold
       from sklearn.preprocessing import StandardScaler
       from sklearn.ensemble import RandomForestClassifier
       from sklearn.linear_model import LogisticRegression
```

```python
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report, roc_curve, auc,␣
 ↪ConfusionMatrixDisplay


seed = 42
np.random.seed(seed)
random.seed(seed)
tf.random.set_seed(seed)


X = df.copy()
X.drop(['nameOrig', 'newbalanceOrig', 'nameDest', 'newbalanceDest', 'quantity',␣
 ↪'oldbalanceOrg_amt', 'oldbalanceDest_amt'], axis=1, inplace=True)
y = X.pop('isFraud')


# Stratified train-test split
skfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)
for train_idx, test_idx in skfold.split(X,y):
    X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
    y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]


sc = StandardScaler()
scaled_train = sc.fit_transform(X_train)
scaled_test = sc.transform(X_test)
X_train = pd.DataFrame(scaled_train, index=X_train.index, columns=X_train.
 ↪columns)
X_test = pd.DataFrame(scaled_test, index=X_test.index, columns=X_test.columns)


X_train, y_train = RandomUnderSampler(sampling_strategy='majority').
 ↪fit_resample(X_train, y_train)
```

```python
[30]: def model_comparison_evaluate(classifiers, X, y):
          print('K-Fold Cross-Validation:\n')
          for name, model in classifiers.items():
              print('{}:'.format(name))

              scoring = ['accuracy', 'precision', 'recall', 'f1', 'roc_auc']

              for score in scoring:
                  scores = cross_val_score(model, X, y, scoring=score, cv=skfold,␣
 ↪n_jobs=-1)
```

```
            print('Mean {} score: {:.3f} ({:.3f})'.format(score, scores.mean(),
     ↪scores.std()))


         print('\n')
```

[31]:
```
classifiers = { 'Random Forest Classifier':
     ↪RandomForestClassifier(class_weight='balanced', random_state=seed),
                 'Logistic Regression':
     ↪LogisticRegression(class_weight='balanced', random_state=seed)
                 }
```

[32]:
```
model_comparison_evaluate(classifiers, X_train, y_train)
```

```
K-Fold Cross-Validation:

Random Forest Classifier:
Mean accuracy score: 0.985 (0.003)
Mean precision score: 0.977 (0.006)
Mean recall score: 0.995 (0.002)
Mean f1 score: 0.986 (0.003)
Mean roc_auc score: 0.998 (0.001)


Logistic Regression:
Mean accuracy score: 0.848 (0.007)
Mean precision score: 0.843 (0.008)
Mean recall score: 0.856 (0.005)
Mean f1 score: 0.849 (0.006)
Mean roc_auc score: 0.927 (0.004)
```

[33]:
```
model = RandomForestClassifier(class_weight='balanced', random_state=seed)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_pred_score = model.predict_proba(X_test)[:,1]
print('Random Forest Classifier:')
print(classification_report(y_pred, y_test, labels=[0,1],
  ↪target_names=['Non-Fraud [0]', 'Fraud [1]']), '\n')



fig, ax = plt.subplots(1, 2, figsize=(20,5))
ax[0].set_title('Confusion Matrix of Random Forest Model:')
ConfusionMatrixDisplay.from_predictions(y_test, y_pred, colorbar=False,
  ↪values_format='', cmap='crest', ax=ax[0])
ax[0].grid(False)
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_score)
roc_auc = auc(fpr, tpr)
ax[1].set_title('ROC Curve - Random Forest Classifier')
ax[1].plot(fpr, tpr, label = 'AUC = %0.3f' % roc_auc, c='steelblue')
ax[1].plot([0,1],[0,1],'--', c='lightsteelblue')
ax[1].legend(loc='lower right')
ax[1].set_ylabel('True Positive Rate')
ax[1].set_xlabel('False Positive Rate')
```

```
Random Forest Classifier:
               precision    recall   f1-score    support

Non-Fraud [0]       0.97      1.00       0.99    1238601
    Fraud [1]       1.00      0.05       0.09      33923

     accuracy                            0.97    1272524
    macro avg       0.99      0.52       0.54    1272524
 weighted avg       0.98      0.97       0.96    1272524
```

[33]: Text(0.5, 0, 'False Positive Rate')
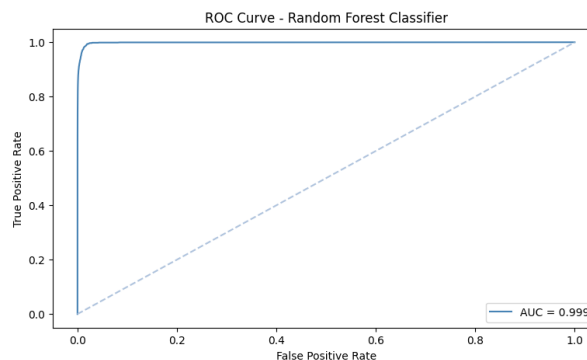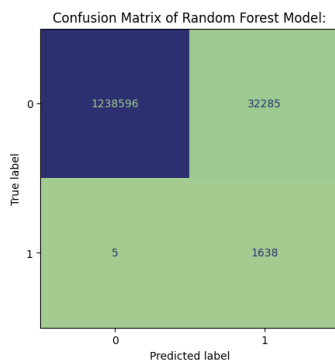
From the confusion matrix, 1,238596 were correctly classified as non-fraudulent payments, and 32,285 people were misclassified as non-fraudulent payments. According to the confusion matrix, 1,638 payments were incorrectly labelled as fraud while 5 payments were correctly identified as fraud.

6 Conclusion

Random Forest obtains the highest score of all using K-fold cross-validation. The best performing model is Random Forest for identifying fraudulent and non-fraudulent payments, as the AUC is 0.999, which is close to 1. This means it has a good separability measure, and the model has an 99.9% chance of being able to distinguish between positive and negative classes..

[35]: `!apt-get install -qq -y texlive-xetex`

```
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 120895 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.4_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.4) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.4_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.4) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.1_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package libwoff1:amd64.
```

```
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/17-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/18-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/19-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/20-libptexenc1_2021.20210626.59705-1ubuntu0.1_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/21-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/22-ruby3.0_3.0.2-7ubuntu2.4_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.4) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/23-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/24-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/25-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/26-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
```

```
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/27-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/28-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/29-libruby3.0_3.0.2-7ubuntu2.4_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.4) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/30-libsynctex2_2021.20210626.59705-1ubuntu0.1_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/31-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/32-libtexlua53_2021.20210626.59705-1ubuntu0.1_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/33-libtexluajit2_2021.20210626.59705-1ubuntu0.1_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/34-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/35-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/36-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/37-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/38-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/39-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/40-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/41-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
```

```
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/42-texlive-
binaries_2021.20210626.59705-1ubuntu0.1_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.1) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/43-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/44-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/45-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/46-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/47-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/48-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/49-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/50-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/51-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/52-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/53-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.1) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.1) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
```

```
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.1) …
Setting up ruby-webrick (1.7.0-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.1) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.4) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.4) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.1) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.1) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
```

```
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.4) …
Setting up ruby3.0 (3.0.2-7ubuntu2.4) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.1) …
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time… done.
```

```python
from nbconvert import PDFExporter
import nbformat

# Specify the path to your Colab notebook file
notebook_path = '/content/drive/MyDrive/Colab Notebooks/OnlinePaymentFraud.
  ↪ipynb'

# Load the notebook
with open(notebook_path) as f:
    nb = nbformat.read(f, as_version=4)
```

```python
# Initialize the PDF exporter
pdf_exporter = PDFExporter()

# Generate the PDF
pdf_data, resources = pdf_exporter.from_notebook_node(nb)

# Specify the output PDF file path
output_pdf_path = '/content/drive/MyDrive/DataSet02/OnlinePayment/your-notebook.
 ↪pdf'

# Save the PDF to the specified path
with open(output_pdf_path, 'wb') as f:
    f.write(pdf_data)
```

[NbConvertApp] WARNING | pattern
'/content/drive/MyDrive/ColabNotebooks/OnlinePaymentFraud.ipynb' matched no
files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.

Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True
--TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the
system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is only useful
for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]

```
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides', 'webpdf']
            or a dotted object name that represents the import path for an
            ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed
    as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized.This
    should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
```

```
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                to output to the directory of each notebook.
To recover
                                previous default behaviour (outputting to the
current
                                working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb --to html

            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides', 'webpdf'].

            > jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes
'base', 'article' and 'report'. HTML includes 'basic', 'lab' and
'classic'. You can specify the flavor of the format used.

> jupyter nbconvert --to html --template lab mynotebook.ipynb

You can also pipe the output to stdout, rather than a file

> jupyter nbconvert mynotebook.ipynb --stdout

PDF is generated via latex

> jupyter nbconvert mynotebook.ipynb --to pdf

You can get (and serve) a Reveal.js-powered slideshow

> jupyter nbconvert myslides.ipynb --to slides --post serve

Multiple notebooks can be given at the command line in a couple of
different ways:

> jupyter nbconvert notebook*.ipynb
> jupyter nbconvert notebook1.ipynb notebook2.ipynb

or you can specify the notebooks list in a config file, containing::

    c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

> jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.

```
[ ]: !xelatex ImplantFailure.tex
```

```
[ ]: from google.colab import files
     files.download('ImplantFailure.pdf')
```