```
import numpy as np
import pandas as pd
import matplotlib as mpl
%matplotlib inline
```

```
df = pd.read_csv('Finance_Social_Network.')
```

```
df.head()
```

	Unnamed:	FDR-Lasso-MSE	FDR-Lasso-Accuracy-Score-Test	FDR-Lasso-AUC	FDR-Lasso-F1	FDR-Ridge-MSE	FDR-Ridge-Accuracy-Score-Test	FDR-Ridge-AUC	FDR-Ridge-F1
0	100000	0.01473	0.9853	0.47147	0.9926	0.01477	0.9853	0.47147	0.9926
1	100001	0.01473	0.9853	0.47147	0.9926	0.01473	0.9853	0.47147	0.9926
2	100002	0.01473	0.9853	0.47147	0.9926	0.01473	0.9853	0.47147	0.9926
3	100003	0.01473	0.9853	0.47147	0.9926	0.01473	0.9853	0.47147	0.9926
4	100004	0.01473	0.9853	0.47147	0.9926	0.01473	0.9853	0.47147	0.9926

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 21 columns):
 #   Unnamed:                                 int64
 1   FDR-Lasso-MSE                            float64
 2   FDR-Lasso-Accuracy-Score-Test            float64
 3   FDR-Lasso-AUC                            float64
 4   FDR-Lasso-F1                             float64
 5   FDR-Ridge-MSE                            float64
 6   FDR-Ridge-Accuracy-Score-Test            float64
 7   FDR-Ridge-AUC                            float64
 8   FDR-Ridge-F1                             float64
 9   FDR-Lasso-Accuracy-Score-Train           float64
 10  FDR-Lasso-Precision-Test                 float64
 11  FDR-Lasso-Recall-Test                    float64
 12  FDR-Lasso-Precision-Train                float64
 13  FDR-Lasso-Recall-Train                   float64
 14  FDR-Ridge-Accuracy-Score-Train           float64
 15  FDR-Ridge-Precision-Test                 float64
 16  FDR-Ridge-Recall-Test                    float64
 17  FDR-Ridge-Precision-Train                float64
 18  FDR-Ridge-Recall-Train                   float64
 19  FDR-Lasso-Mean-Cross-Val                 float64
 20  FDR-Ridge-Mean-Cross-Val                 float64
dtypes: int64, 21 float64
memory usage: 1.6 MB
```

```
df.describe()
```

	Unnamed:	FDR-Lasso-MSE	FDR-Lasso-Accuracy-Score-Test	FDR-Lasso-AUC	FDR-Lasso-F1	FDR-Ridge-MSE	FDR-Ridge-Accuracy-Score-Test	FDR-Ridge-AUC	FDR-Ridge-F1
count	10000.0	10000.000000	10.000000	10.000000	10.000000	10000.000000	10.000000	10.000000	10.000000
mean	104999.5	0.014733	0.985300	0.471470	0.992600	0.014733	0.985300	0.471470	0.992600
std	2886.896680	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
min	100000.0	0.014730	0.985300	0.471470	0.992600	0.014730	0.985300	0.471470	0.992600
25%	102499.75	0.014730	0.985300	0.471470	0.992600	0.014730	0.985300	0.471470	0.992600
50%	104999.5	0.014730	0.985300	0.471470	0.992600	0.014730	0.985300	0.471470	0.992600
75%	107499.25	0.014730	0.985300	0.471470	0.992600	0.014730	0.985300	0.471470	0.992600
max	109999.0	0.014730	0.985300	0.471470	0.992600	0.014730	0.985300	0.471470	0.992600

```
df.shape
```

```
(10000, 21)
```

```
df[['Unnamed:']].sample()
```

	Unnamed:
9	100009

```
Name: Unnamed:, dtype: int64
```

```
df['FDR-Lasso-F1'].value_counts()
```

```
0.9926    10
Name: FDR-Lasso-F1, dtype: int64
```

```
plt.pie(df['FDR-Lasso-F1'].value_counts(),autopct='%1f%%',labels=['0.9926'])
plt.legend()
plt.show()
```

```
df.duplicated().sum()
```

```
0
```

```
df.dropna(inplace=True)
```

```
# to see statistics the column of values in each column
df.nunique()
```

```
Unnamed:                                 10000
FDR-Lasso-MSE                                1
FDR-Lasso-Accuracy-Score-Test                1
FDR-Lasso-AUC                                1
FDR-Lasso-F1                                 1
FDR-Ridge-MSE                                1
FDR-Ridge-Accuracy-Score-Test                1
FDR-Ridge-AUC                                1
FDR-Ridge-F1                                 1
FDR-Lasso-Accuracy-Score-Train               1
FDR-Lasso-Precision-Test                     1
FDR-Lasso-Recall-Test                        1
FDR-Lasso-Precision-Train                    1
FDR-Lasso-Recall-Train                       1
FDR-Ridge-Accuracy-Score-Train               1
FDR-Ridge-Precision-Test                     1
FDR-Ridge-Recall-Test                        1
FDR-Ridge-Precision-Train                    1
FDR-Ridge-Recall-Train                       1
FDR-Lasso-Mean-Cross-Val                     1
FDR-Ridge-Mean-Cross-Val                     1
dtype: int64
```

```
# to see the number of values in the dataset
df.nunique().sum()
```

```
10016
```

```
df.dtypes.sample(2)
```

```
FDR-Lasso-Recall-Train    float64
FDR-Ridge-F1              float64
dtype: object
```

```
df.nunique.sum()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
AttributeError: 'DataFrame' object has no attribute 'nunique'
```

```
df.nunique().sample(2)
```

```
FDR-Lasso-Recall-Test    1
FDR-Ridge-Recall-Test    1
dtype: int64
```

```
df.nunique().sample(2).sum()
```

```
2
```

```
df.nunique().sample(2).sum()/df.nunique().sum()*100
```

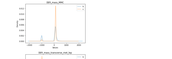
```
0.1998401279031957
```

```
[1] df_sample1[,"PM_ALL_MIN"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[2] df_sample1[,"PM_ALL_MIN_CONTROL"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[3] df_sample1[,"PM_ALL_MIN_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[4] df_sample1[,"PM_ALL_MIN_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[5] df_sample1[,"PM_ALL_MIN_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[6] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[7] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[8] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[9] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[10] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[11] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[12] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[13] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[14] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[15] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[16] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[17] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[18] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[19] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[20] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
```

```
[21] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[22] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[23] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[24] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[25] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[26] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP"],xlab=NULL,ylab=NULL,lwd=0.5,"Clean")
[27] df_sample1[,"PM_ALL_MIN_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP_CP
```

Cum_Profit_Short_Runup

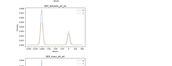
```
# Data for model
X = df[features]
y = df['target']

# Split data into training and test sets
X_train, X_test, y_train, y_test = \
train_test_split(X, y, \
                 test_size=0.2, \
                 random_state=42, \
                 stratify=y)
print("Done")
```


DER_mass_MMC

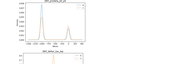
DER_mass_transverse_met_lep

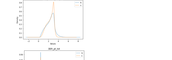
DER_mass_vis

DER_pt_h


DER_deltaeta_jet_jet

DER_mass_jet_jet

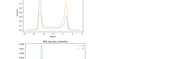

DER_prodeta_jet_jet

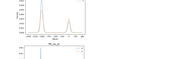

DER_deltar_tau_lep

DER_pt_tot

DER_sum_pt_ch

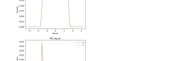
DER_pt_ratio_lep_tau


DER_met_phi_centrality

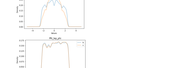

DER_lep_eta_centrality

PRI_tau_pt

PRI_tau_eta


PRI_tau_phi

PRI_lep_pt

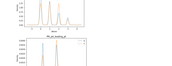

PRI_lep_eta

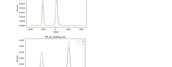
PRI_lep_phi

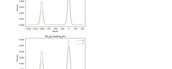
PRI_met

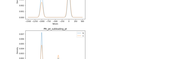
PRI_met_phi

PRI_met_sumet


PRI_jet_num


PRI_jet_leading_pt


PRI_jet_leading_eta


PRI_jet_leading_phi

PRI_jet_subleading_pt

PRI_jet_subleading_eta

PRI_jet_subleading_phi

PRI_jet_all_pt

Weight

```
(0,0,0,0)[0,0,0,
```

	DIR_Ecos_DIRM	DIR_Ecos_Insumos_Ind_Agr	DIR_Ecos_Lab	DIR_p_F	DIR_dEstud_p_F	DIR_Est_p_F	DIR
DIR_Ecos	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ecos_Insumos_Ind_Agr	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ecos_Lab	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
DIR_dEstud_p_F	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
DIR_Est_p_F	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
DIR_dAnal_p_F	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
DIR_dAnalEst_p_F	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_dAnal_Ecos_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_dAnal_Ecos_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind_Ecos_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind_Ecos_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_dAnal_Ecos_Ind_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_dAnal_Ecos_Lab_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_p_F	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_p_F_Ecos_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_p_F_Ecos_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Ind_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Lab_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Ind_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_Ind_Ecos_Lab_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ecos_Ind	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ecos_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ecos_Ind_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ecos_Lab_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind_Ecos_Ind_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind_Ecos_Lab_Agr	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DIR_p_F_Ind_Ecos_Ind_Agr_Lab	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

17 rows x 17 columns

```
# Plotando correlações
corr_matrix = df[cols].corr()
sns.heatmap(corr_matrix, cmap='RdBu_r', annot=True, fmt='.2f', linewidths=0.5, vmin=-1, vmax=1)
```

```
<matplotlib.
```


DIR_Ecos_Insumos_Ind_Agr is highly correlated with DIR_p_F and DIR_dAnal_p_F: 1.000000 Rejected

DIR_Ecos_Lab is highly correlated with DIR_dAnal_p_F: 1.000000 Rejected

DIR_p_F is highly correlated with DIR_Ecos_Insumos_Ind_Agr: 1.000000 Rejected

DIR_dEstud_p_F is highly correlated with DIR_Est_p_F: 1.000000 Rejected

DIR_Est_p_F is highly correlated with DIR_dEstud_p_F: 1.000000 Rejected

DIR_dAnal_p_F is highly correlated with DIR_Ecos_Insumos_Ind_Agr: 1.000000 Rejected

DIR_dAnalEst_p_F is highly correlated with DIR_dAnal_Ecos_Ind: 1.000000 Rejected

DIR_dAnal_Ecos_Ind is highly correlated with DIR_dAnalEst_p_F: 1.000000 Rejected

DIR_dAnal_Ecos_Lab is highly correlated with DIR_p_F: 1.000000 Rejected

DIR_p_F_Ind is highly correlated with DIR_p_F_Ecos_Ind: 1.000000 Rejected

DIR_p_F_Ind_Ecos_Ind is highly correlated with DIR_p_F_Ind: 1.000000 Rejected

DIR_p_F_Ind_Ecos_Lab is highly correlated with DIR_p_F_Ecos_Lab: 1.000000 Rejected

DIR_dAnal_Ecos_Ind_Agr is highly correlated with DIR_Ind_Ecos_Ind_Agr: 1.000000 Rejected

DIR_dAnal_Ecos_Lab_Agr is highly correlated with DIR_Ind_Ecos_Lab_Agr: 1.000000 Rejected

DIR_Ind_p_F is highly correlated with DIR_p_F: 1.000000 Rejected

DIR_Ind_p_F_Ecos_Ind is highly correlated with DIR_p_F_Ecos_Ind: 1.000000 Rejected

DIR_Ind_p_F_Ecos_Lab is highly correlated with DIR_p_F_Ecos_Lab: 1.000000 Rejected

DIR_Ind_Ecos_Ind_Agr is highly correlated with DIR_dAnal_Ecos_Ind_Agr: 1.000000 Rejected

DIR_Ind_Ecos_Lab_Agr is highly correlated with DIR_dAnal_Ecos_Lab_Agr: 1.000000 Rejected

DIR_Ind is highly correlated with DIR_Ind_p_F: 1.000000 Rejected

DIR_Ind_Ecos_Ind is highly correlated with DIR_p_F_Ecos_Ind: 1.000000 Rejected

DIR_Ind_Ecos_Lab is highly correlated with DIR_p_F_Ecos_Lab: 1.000000 Rejected

DIR_Ind_Ecos_Ind_Agr is highly correlated with DIR_dAnal_Ecos_Ind_Agr: 1.000000 Rejected

DIR_Ind_Ecos_Lab_Agr is highly correlated with DIR_dAnal_Ecos_Lab_Agr: 1.000000 Rejected

DIR_p_F_Ecos_Ind is highly correlated with DIR_p_F: 1.000000 Rejected

DIR_p_F_Ecos_Lab is highly correlated with DIR_p_F: 1.000000 Rejected

DIR_p_F_Ecos_Ind_Agr is highly correlated with DIR_dAnal_Ecos_Ind_Agr: 1.000000 Rejected

DIR_p_F_Ecos_Lab_Agr is highly correlated with DIR_dAnal_Ecos_Lab_Agr: 1.000000 Rejected

DIR_p_F_Ind_Ecos_Ind_Agr is highly correlated with DIR_p_F: 1.000000 Rejected

DIR_p_F_Ind_Ecos_Lab_Agr is highly correlated with DIR_p_F: 1.000000 Rejected