

Country Index Classifier

برای این امر از دو روش استفاده می‌شود:

تکنیکال: مدل quant classifier که در قسمت quant trader توضیح داده شده است.

فاندامتال: مدل ensemble classifier که از داده‌های فاندامنتال و ترکیب چندین مدل یادگیری ماشین به صورت زیر، قوی شدن و یا ضعف شاخص هر کشور را محاسبه می‌کند.

```
def country_index_classifier(method, initialize, country='USD', TimeFrame='1d',
                             use_haiken_ashi=True, hyper_tune=False, plot_results=False, force_to_train=False, update_fund_data=False):
    """
    Classifies the forecast and probability based on the given method and parameters.

    Parameters:
    - method (str): The classification method to use ('quant' or 'fund').
    - initialize: The initialization parameter for the classifier.
    - country (str): The country currency to consider (default: 'USD').
    - TimeFrame (str): The time frame to use for classification (default: '1d').
    - use_haiken_ashi (bool): Whether to use Haiken Ashi candles for classification (default: True).
    - hyper_tune (bool): Whether to perform hyperparameter tuning (default: False).
    - plot_results (bool): Whether to plot the classification results (default: False).
    - force_to_train (bool): Whether to force the classifier to train (default: False).
    - update_fund_data (bool): Whether to Download and update fundamental data

    Returns:
    - forecast (str): The forecast result ('sell', 'neutral', or 'buy').
    - prob (float): The probability of the forecast result.
    - outputs (dict): Outputs from the currency classifier.

    Raises:
    - ValueError: If an invalid method is specified.
    """
```

Fundamental Model

ابتدا پیش از هرچیز نیاز به جمع‌آوری داده‌ها می‌باشد. داده‌های فاندامنتال از سایت investing.com کراال می‌شود. متأسفانه این سایت API نمی‌دهد و بنابراین کد investing.py برای کراال کردن و به‌روزرسانی داده‌ها با استفاده از Selenium مورد استفاده قرار می‌گیرد.

```

"""
investing.py

Script to download and process investing.com data

Usage:
    - how to update all data (takes 45 minutes)
      update_investing(method='update-all')

    - how to update a country data (takes less than 10 minutes)
      update_investing(method='update-country', country='USD')
      X, y = get_investing(country='USD', timeframe='1w')

    - how to update a single series (takes less than 1 minutes)
      update_investing(method=None, name='US Dollar Index')
      dxy = pd.read_csv("investing_data/US Dollar Index.csv")
      dxy = clean_investing_data(dxy, timeframe)
"""

```

تابع برای محاسبه شاخص هر کشور مطابق با جفت ارز های مربوطه مورد استفاده قرار میگیرد:

```

def get_country_index(country, timeframe='1d'):

    dxy = pd.read_csv("investing_data/US Dollar Index.csv")
    dxy = clean_investing_data(dxy, timeframe)
    if country == 'USD':
        country_index = dxy

    else:
        if country in ['CAD', 'JPY', 'SEK', 'CHF']:
            Ticker = 'USD' + country
            df = pd.read_csv(f"investing_data/{Ticker}.csv")
            df = clean_investing_data(df, timeframe)
            df=df[['Open', 'High', 'Low', 'Close']]
            country_index = dxy/df
        else:
            Ticker = country + 'USD'
            df = pd.read_csv(f"investing_data/{Ticker}.csv")
            df = clean_investing_data(df, timeframe)
            df=df[['Open', 'High', 'Low', 'Close']]
            country_index = df*dxy

    country_index['Mean']=np.mean(pd.concat((country_index['Low'],country_index['High'], country_index['Close']),axis=1),axis=1)
    country_index['diff']=country_index['Mean']-country_index['Mean'].shift(1)

    return country_index.dropna()

```

با یک تست های انجام شده، داده های مهم مطابق با شاخص هر کشور به صورت زیر بدست آمده است:

```
def get_features():
    features=dict()

    features['USD']=['USD_bond_2Y', 'USD_bond_5Y', 'USD_bond_10Y',
                    'EURUSD', 'NZDUSD', 'GBPUSD', 'USDCHF',
                    'NASDAQ', 'VIX', 'T-Note', 'US 30 Cash',
                    'Silver', 'Gold', 'Copper', 'CRB']

    features['CAD']=['CAD_bond_7Y', 'CAD_bond_5Y', 'CAD_bond_4Y', 'CAD_bond_3Y', 'CAD_bond_2Y', 'CAD_bond_10Y',
                    'CADCHF', 'GBPCAD', 'CADJPY', 'EURCAD', 'VIX', 'T-Note', 'US Wheat', 'Heating Oil', 'CRB']

    features['AUD']=['AUD_bond_10Y', 'AUD_bond_4Y',
                    'AUDCHF', 'AUDJPY', 'AUDCAD', 'VIX', 'NASDAQ', 'USD_bond_10Y', 'Lumber', 'Brent Oil', 'Copper', 'CRB']

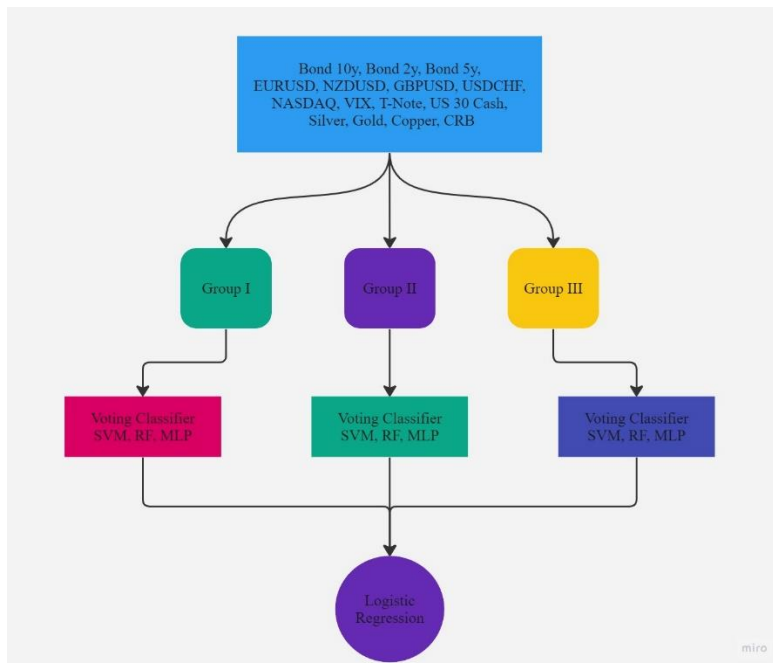
    features['NZD']=['NZD_bond_6M', 'VIX', 'NASDAQ', 'Silver', 'Brent Oil', 'Copper',
                    'EURNZD', 'NZDUSD', 'GBPNZD']

    features['JPY']=['JPY_bond_8Y', 'JPY_bond_10Y', 'JPY_bond_30Y',
                    'EURJPY', 'GBPJPY', 'CADJPY', 'VIX', 'NASDAQ', 'Heating Oil', 'US Dollar Index']

    features['GBP']=['GBP_bond_3Y', 'GBP_bond_1M', 'GBP_bond_6M',
                    'GBPUSD', 'GBPJPY', 'GBPCHF', 'EURGBP', 'Heating Oil']

    features['EUR']=['Germany 10-Year_Bond', 'Germany 5-Year_Bond', 'France 10-Year_Bond', 'US Dollar Index',
                    'EURGBP', 'EURUSD', 'EURJPY', 'EURCAD', 'NASDAQ', 'Brent Oil', 'Silver', 'CRB', 'Natural Gas']
```

حال با استفاده از cluster کردن ویژگی ها هر یک به همراه تاخیرهای بهینه بدست آمده از تابع FindBestLag وارد یک مدل Voting Classifier متشکل از (SVM, RF, MLP) می شود و در نهایت خروجی این مدل ها وارد مدل Logistic Regression تا تصمیم گیری نهایی انجام شود.



```
def fund_model(country , timeframe, features=None, method='clustering', subset_cols=None,
               use_haiken_ashi=True, hyper_tune=False, plot_results=False, update_fund_data=False):

    if features==None: features=get_features()[country]

    if update_fund_data: update_investing(method='update-country', country=country)
    X, y = get_investing(country=country, timeframe=timeframe)
```

```

if method=='user_defined':

    # subset_cols=[
    # ['EURUSD', 'NZDUSD', 'GBPUSD', 'USDCHF'],
    # ['Silver', 'Gold', 'Copper', 'CRB'],
    # ['NASDAQ','VIX','T-Note','US 30 Cash', 'bond10y', 'bond2y', 'bond5y']
    # ]

    subsets=subset_dataframes(X, features, method='user_defined', subset_cols=subset_cols)
else:
    subsets=subset_dataframes(X, features, method='clustering', n_clusters=3, subset_cols=None)

```

```

Targets=y
if use_haiken_ashi: Targets=find_heikin_ashi_candlestick(Targets)

Targets['label'] = Targets['diff'].apply(lambda x: 1 if x > 0 else 0)

Inputs = []
Forecast_Inputs=[]
Labels = []
Index=[]
for subset in subsets:
    dataset = pd.concat((subset, pd.DataFrame(Targets['diff']), pd.DataFrame(Targets['label'])), axis=1).dropna()
    X, y = dataset[dataset.columns[:-2]], dataset[dataset.columns[-2:]]

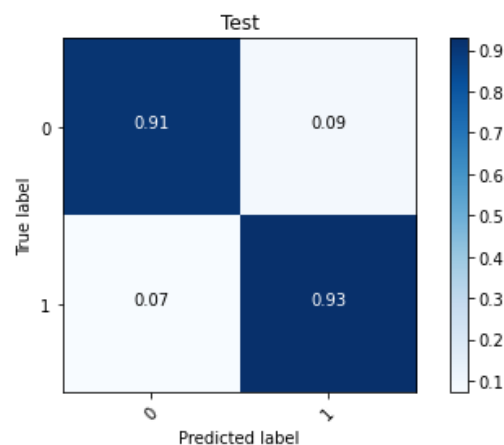
    X_scaled,_ = standardization(X.to_numpy())

    Best_Inputs, Forecast_Input=FindBestLags(X_scaled, y['diff'].to_numpy(), lagmax=200, nlags=10)
    Inputs.append(Best_Inputs.reshape((-1, Best_Inputs.shape[1]*Best_Inputs.shape[2])))
    Forecast_Inputs.append(Forecast_Input.reshape((-1, Forecast_Input.shape[1]*Forecast_Input.shape[2])))
    Labels.append(y['label'].to_numpy()[~len(Best_Inputs):])
    Index.append(dataset.index[~len(Best_Inputs):])





forecast, y_pred, prob= ensemble_classifier(Inputs, Targets, Forecast_Inputs, Labels, Index, hyper_tune=hyper_tune, plotResults=plot_results)
return forecast, y_pred, prob

```

Confusion Matrix این مدل برای شاخص دلار در تایم فریم روزانه برای دادگان تست به صورت زیر است:



خروجی نهایی هردو مدل به صورت زیر ارائه می شود:

Model Type	Haiken ashi	normal	Haiken ashi	normal
Fund				
Quant	