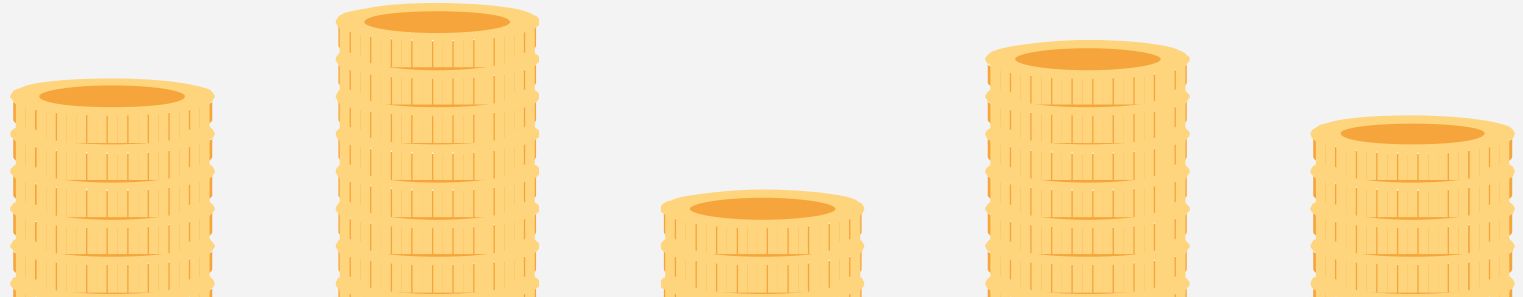


Constructor de Portafolios Sencillo

Julian D. Osorio



Contenidos

Carga de Datos

Pronostico

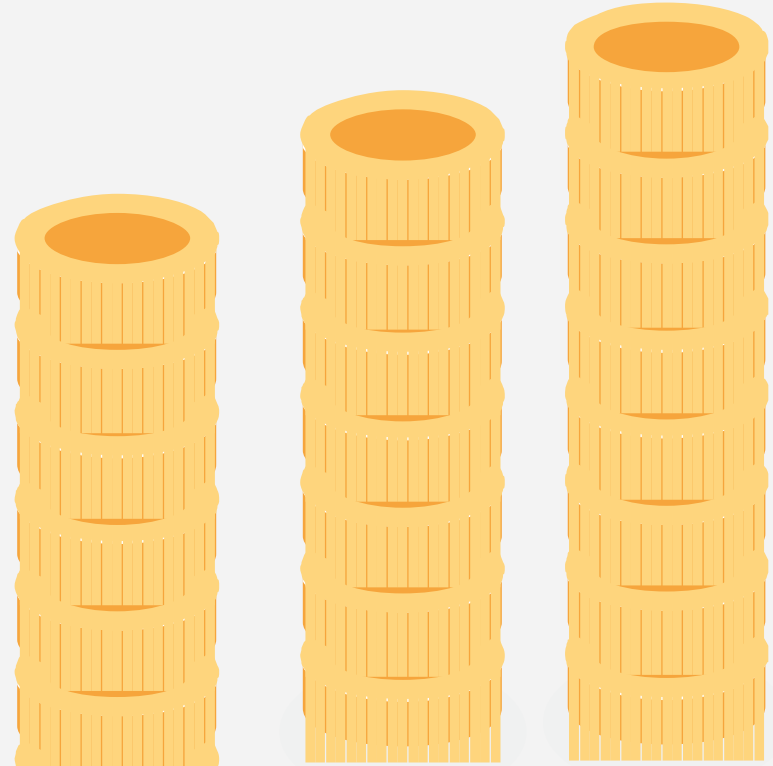
Modelo CPPI



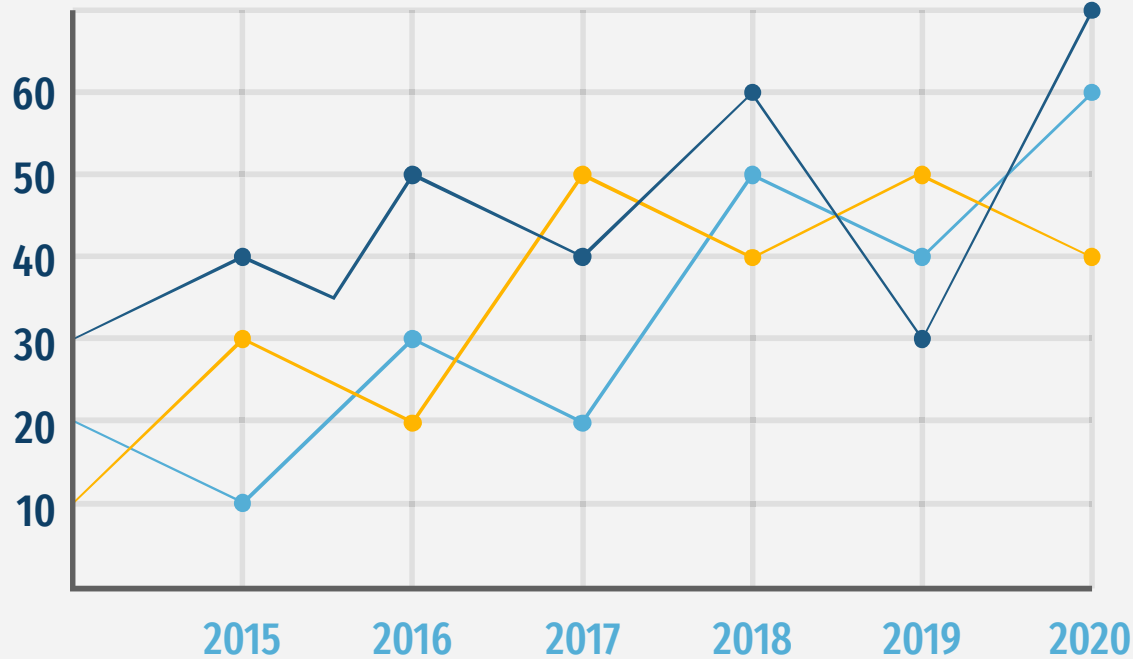
Modelamiento

Modelo de Markowitz

Carga de Datos



Indicadores Tecnicos



Datos Historicos

Extraídos de Yahoo Finance: aperturas, extremos y cierres.

Media Movil

Identificación de Soportes y Resistencias

Índice de Fuerza Relativa

Tendencia a Futuras Correcciones



```
today = date.today().strftime("%Y-%m-%d")

def load( stock = None, start = "2009-01-01", stop = today, delta_target = True, mensual = False ):
    #df = yf.download(stock, start, stop)

    if mensual:
        df = yf.download(stock, "2000-01-01", stop)

        new_indx = pd.to_datetime(df.index)
        df.set_index(new_indx, inplace=True)
        df = df.resample('M').mean()

        df['RSI'] = pda.rsi( df['Close'], length = 2 ) # indicadores de analisis tecnico
        df['FMA'] = pda.ema( df['Close'], length = 2 )
        df['MMA'] = pda.ema( df['Close'], length = 4 )
        df['SMA'] = pda.ema( df['Close'], length = 6 )

    else:
        df = yf.download(stock, start, stop)

        df['RSI'] = pda.rsi( df['Close'], length = 15 ) # indicadores de analisis tecnico
        df['FMA'] = pda.ema( df['Close'], length = 20 )
        df['MMA'] = pda.ema( df['Close'], length = 100 )
        df['SMA'] = pda.ema( df['Close'], length = 150 )

    if delta_target:
        df['Target'] = df['Adj Close'] - df['Open']
        df['Target'] = df['Target'].shift(-1) # tomorrow stock value delta
        df['Categoria'] = [ 1 if df.Target[i] > 0 else 0 for i in range( df.shape[0] ) ]
    else:
        df['Target'] = df['Adj Close'].shift(-1) # tomorrow stock value
        df['Categoria'] = [ 1 if df.Target[i] > df['Adj Close'].iloc[i] else 0 for i in range( df.shape[0] ) ]

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

    df.drop( ['Volume', 'Close'], axis = 1, inplace = True )

    return df
```

load():

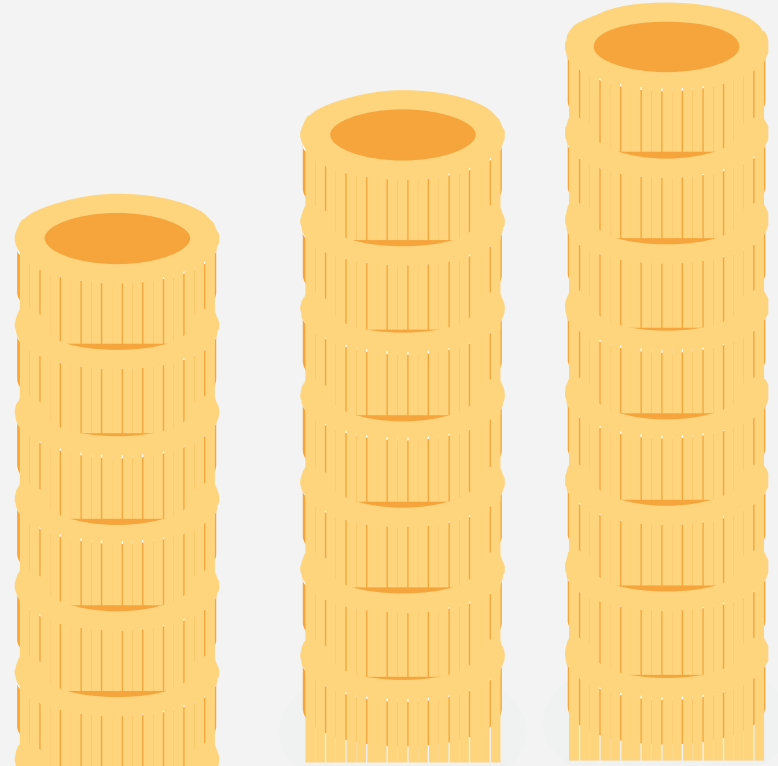
Parametros de Entrada.

- Start:
Cota inferior de los Datos
- Stop:
Cota Superior
- delta_target
Tipo de Target a considerar; precio de Cierre o delta de precio.

Salida.

- Pandas DataFrame

Modelamiento



Modelamiento

Agrupación



Tomar los datos ya limpios, Normalizarlos y organizarlos tales que puedan servir de Input para los modelos

LSTM



Entrenar y evaluar resultados usando una Red Neuronal Recursiva

XGBoost



Contrastar desempeños de ambos modelos.

Comparación de los Modelos

XGBoost



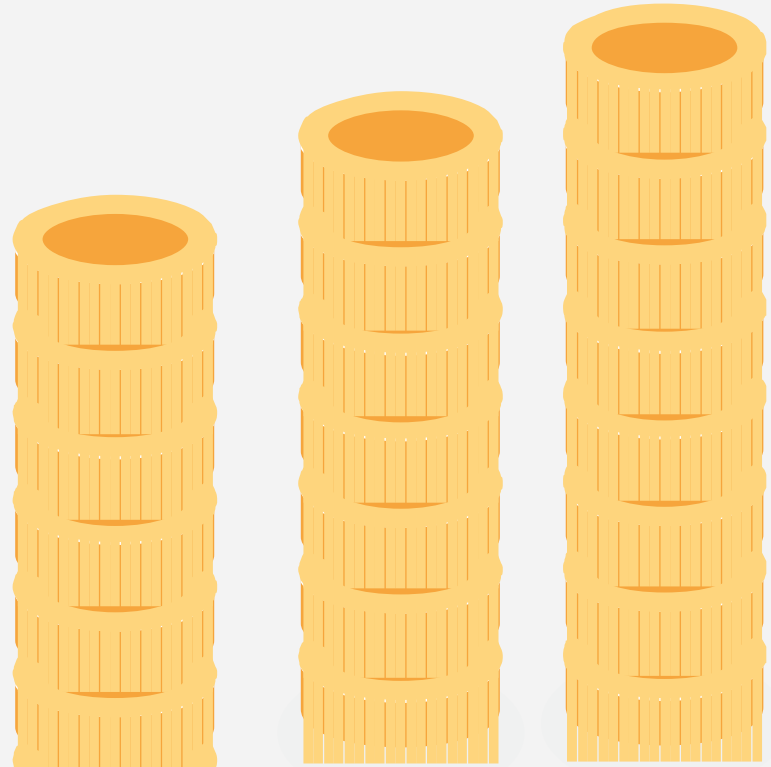
LSTM



Sobre la variable Objetivo



Pronostico



Alimentar el Modelo

01

Cargar el Modelo

```
from keras.models import  
lstm = load_model("VEIEX.h5")
```

02

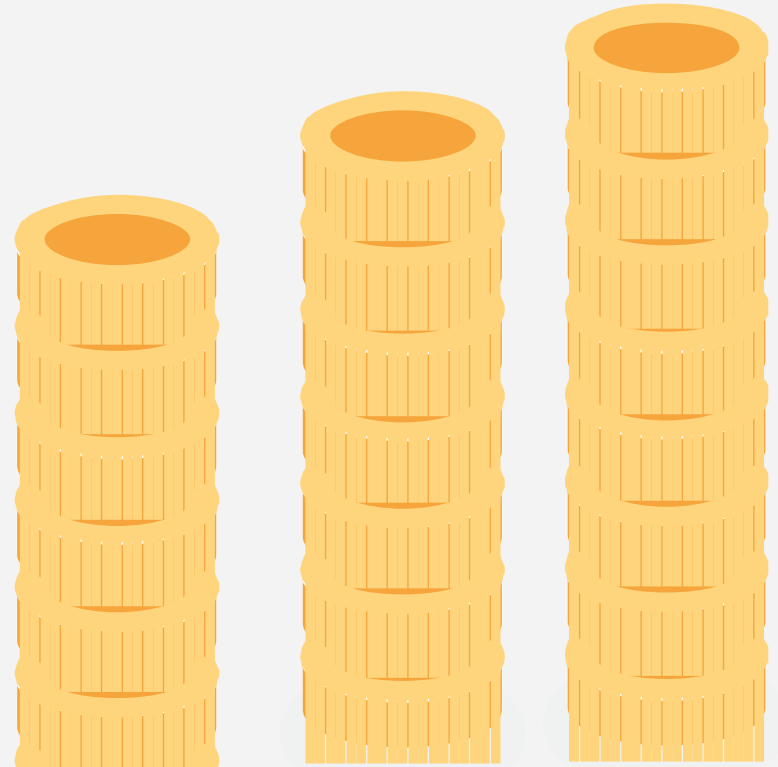
Recolectar Datos

```
Input(1, steps, features)
```



```
def forecast(model):  
    ...  
    Tomar los 'steps' datos más recientes para predecir recursivamente el siguiente  
    ...  
  
    temp = date.today()  
    mes_atras = temp - relativedelta(months=1)  
    mes_atras = mes_atras.strftime("%Y-%m-%d")  
  
    df_forecast = df.copy()  
    df_forecast.drop( [ 'Date', 'Categoria'], axis = 1, inplace = True )  
    df_forecast = df_forecast.iloc[-steps,:]  
  
    x_df = df_forecast.iloc[ :, :8 ]  
  
    sc = MinMaxScaler(feature_range=(0,1))  
    x_df = sc.fit_transform(x_df)  
    #print(df_forecast)  
  
    x_forecast = []  
    x_forecast.append(x_df) ; x_forecast = np.array(x_forecast)  
  
    y_forecast = model.predict(x_forecast)  
    y_forecast = inversa(y_forecast, df_forecast, borrar = False, test = False)  
  
    return y_forecast[0]
```

Modelo de Markowitz



Modelo de Markowitz

Retorno

$$R_p = \sum_i^n \sum_i^n \omega_i R_i = W^T R$$

Volatilidad

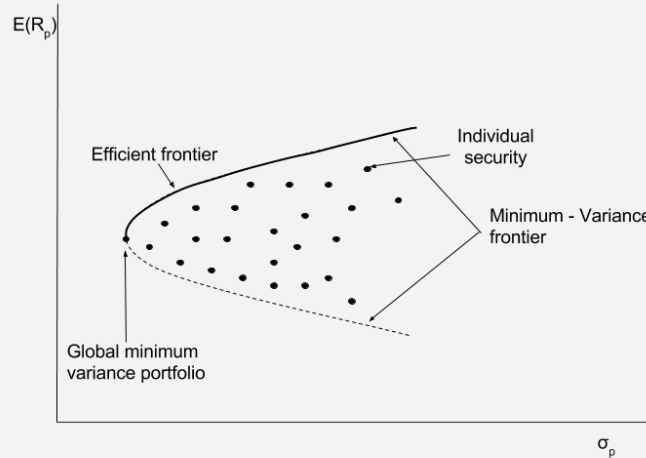
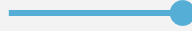
$$\sigma_p^2 = \sum_i^n \sum_j^n \omega_i \omega_j \sigma_{ij}$$

Funcion de
Perdida

$$J(\sigma, \omega) = \frac{1}{2} W^T Cov \cdot W$$



Optimización de Portafolios



$$S_p = \frac{\sum^n \omega_i \mu_i - R_f}{\sqrt{\sum^n \omega_i \omega_j \sigma_{ij}}}$$

Global Minimum Variance Portfolio (GMV)

Portafolio de menor Riesgo en la Frontera de Eficiencia



El modelo de Markowitz sufre en particular de Riesgos de Estimación, donde los Retornos Esperados, así sea por una pequeña desviación, afectan considerablemente la Frontera de Eficiencia. EL GMV es el Portafolio menos afectado por el Riesgo de Estimación.

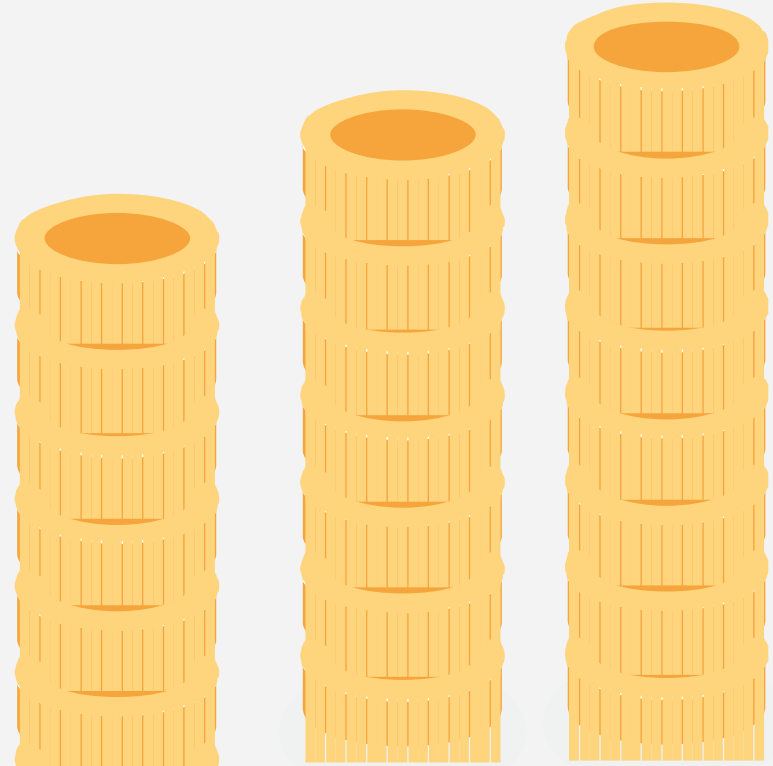
Maximun Sharp Ratio Portfolio (MSR)

Portafolio con mayor Utilidad por unidad de Volatilidad

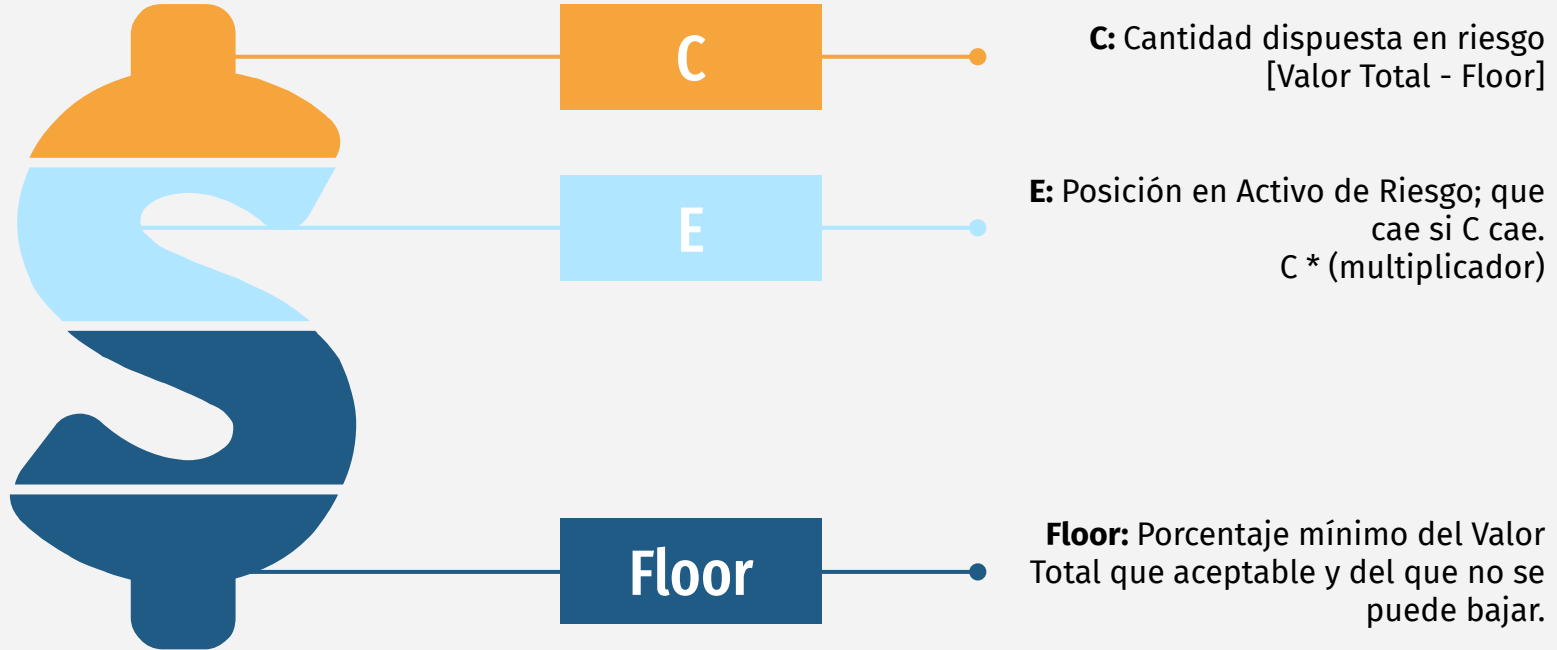


En esencia, el MSR me dice cuanto más retorno puedo tener respecto a una inversión en un activo libre de Riesgo; usualmente Bonos del Tesoro.

Modelo CPPI



Constant Proportion Portfolio

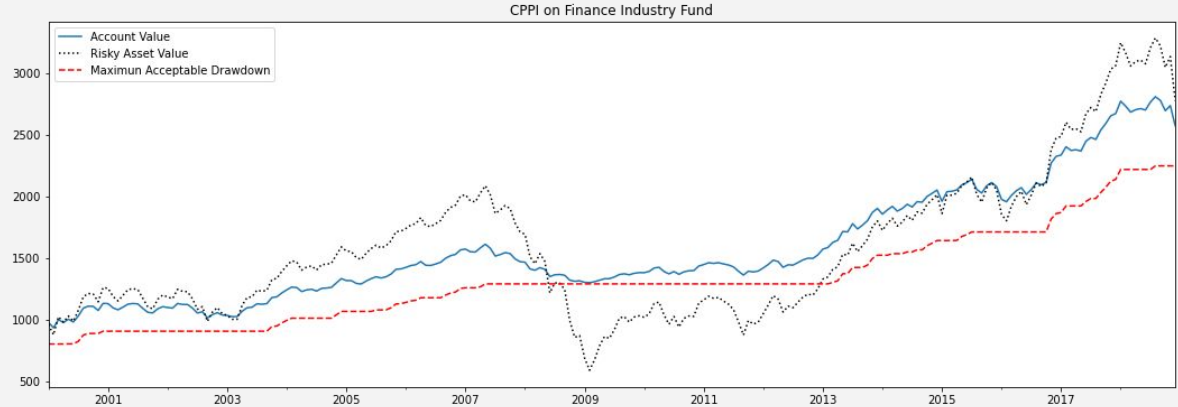


Movimiento Browniano



$$\frac{R_{t+dt} - R_t}{R_t} = \mu dt + \sigma \sqrt{dt} \xi_t$$

Cambio en el retorno esperado del Portafolio influenciado por variables aleatorias (Mov. Browniano)





¡Gracias!