

به نام خدا

یاسین عسکریان

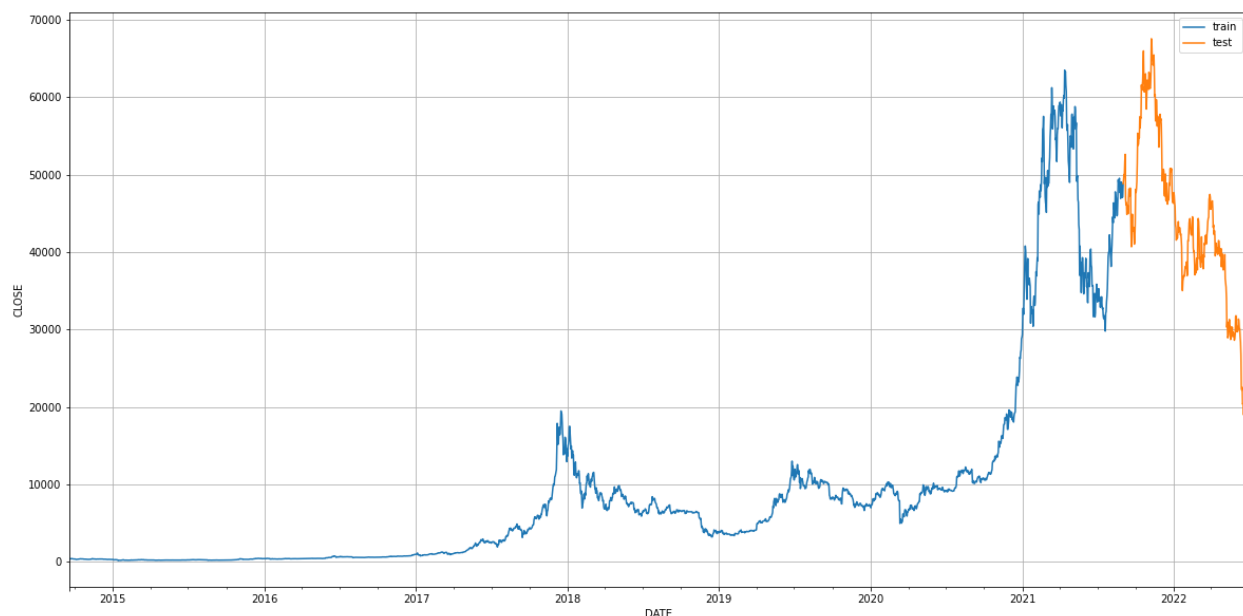
تمرین سری دوازدهم

بخش 1:

طبق خواسته سوال پیش رفتیم و داده های را از طریق yfinance دانلود کردیم که 5 تای اول ان را در تصویر زیر مشاهده می کنید

Date	Open	High	Low	Close	Adj Close	Volume
2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100

قیمت درج شده در ستون close را به عنوان قیمت مرجع میگیریم (برداشت کردم که این قیمت بسته شده بیت کوین در ان روز هست) سپس داده های تست و آموزشی خود را مطابق خواسته سوال جدا کردیم داریم



سپس قیمت های ستون close در داده های آموزش و آزمایش را به وسیله MinMaxScaler به رنج 0 تا 1 تبدیل می کنیم

```

scaler = MinMaxScaler()
all_data = scaler.fit_transform(np.concatenate((train_prices, test_prices)))
train_prices = all_data[:len(train_prices)]
test_prices = all_data[len(train_prices):]

```

سپس نیاز است داده های ورودی و خروجی مدل را آماده کنیم که به اینصورت باشد:

قیمت ها را در یک بازه 60 روزه به مدل بدهیم تا قیمت در روز 61 را پیش بینی کند از قطعه کد زیر برای تبدیل داده های ورودی به بازه های 60 روزه استفاده می کنیم

```

def prepair_data(data:np.array, count:int=60):
    x = []
    y = []
    for i in range(count, data.shape[0]):
        x.append(data[i-count:i,0])
        y.append(data[i,0])

    return np.array(x), np.array(y)

```

خروجی هم به صورت زیر است:

```

train input data shape : (2482, 60, 1) and output data shape is (2482,)
test input data shape: (240, 60, 1) and output data shape is (240,)

```

در نظر داشته باشیم که اولین داده ورودی مدل ما در واقع قیمت 60 روز بعد از اولین داده دانلود شده ماست.

حال مدل را می سازیم

```
def build_model(count=60):
    model = models.Sequential()
    model.add(layers.Input((count, 1)))
    model.add(layers.LSTM(units=50, return_sequences=True))
    model.add(layers.Dropout(0.2))
    model.add(layers.LSTM(units=50, return_sequences=True))
    model.add(layers.Dropout(0.2))
    model.add(layers.LSTM(units=50, return_sequences=True))
    model.add(layers.Dropout(0.2))
    model.add(layers.LSTM(units=50, return_sequences=False))
    model.add(layers.Dense(1))

    return model

model = build_model()
model.summary()
```

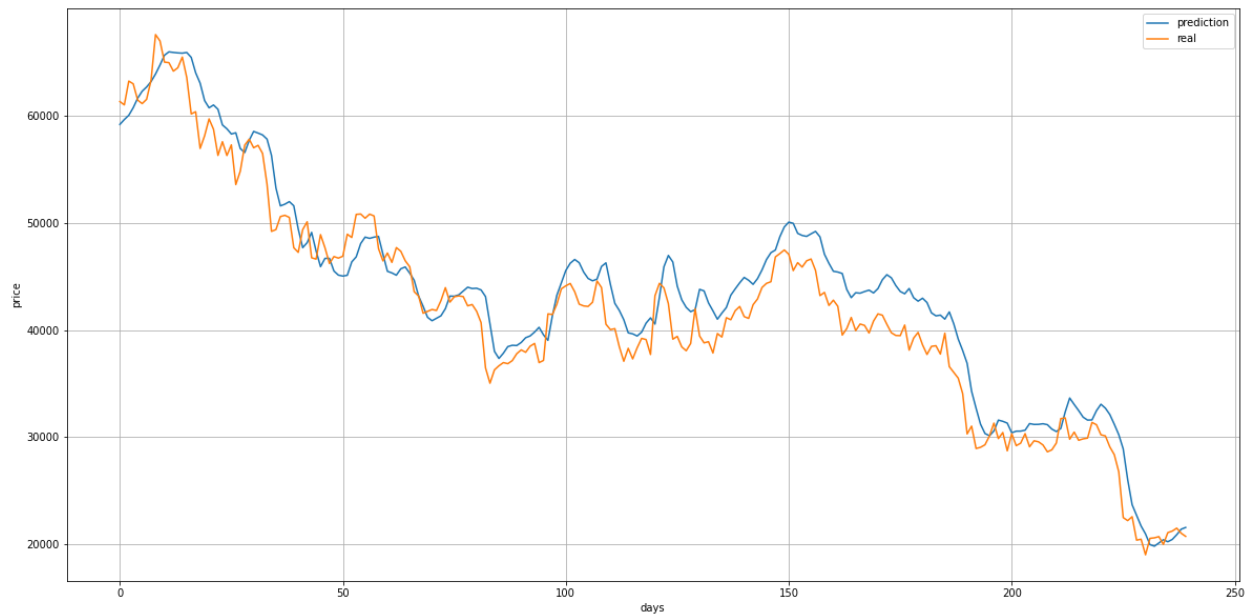
سپس مدل را بهینه ساز adam و تابع ضرر mse compile می کنیم و عملیات آموزش داده های آموزشی را شروع می کنیم

```
model.compile(loss = 'mean_squared_error', optimizer = 'adam')
```

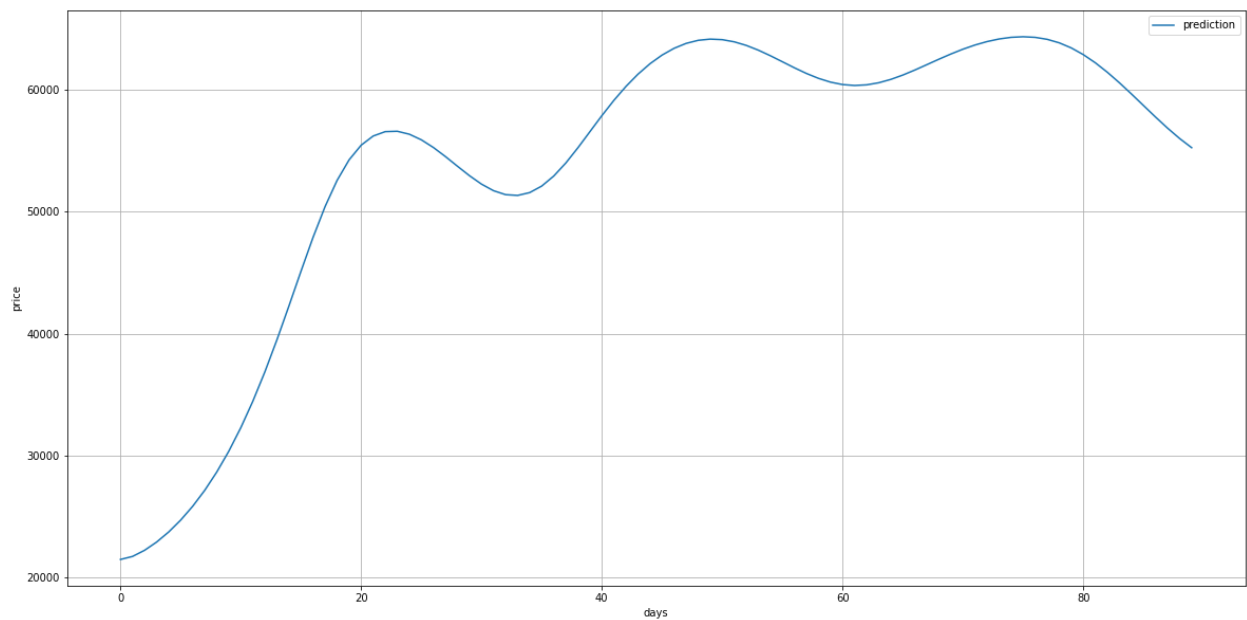
```
model.fit(x_train,y_train,batch_size=32,epochs=100)
```

حال نیاز است که داده های آزمایشی خودمان را پیشبینی کنیم

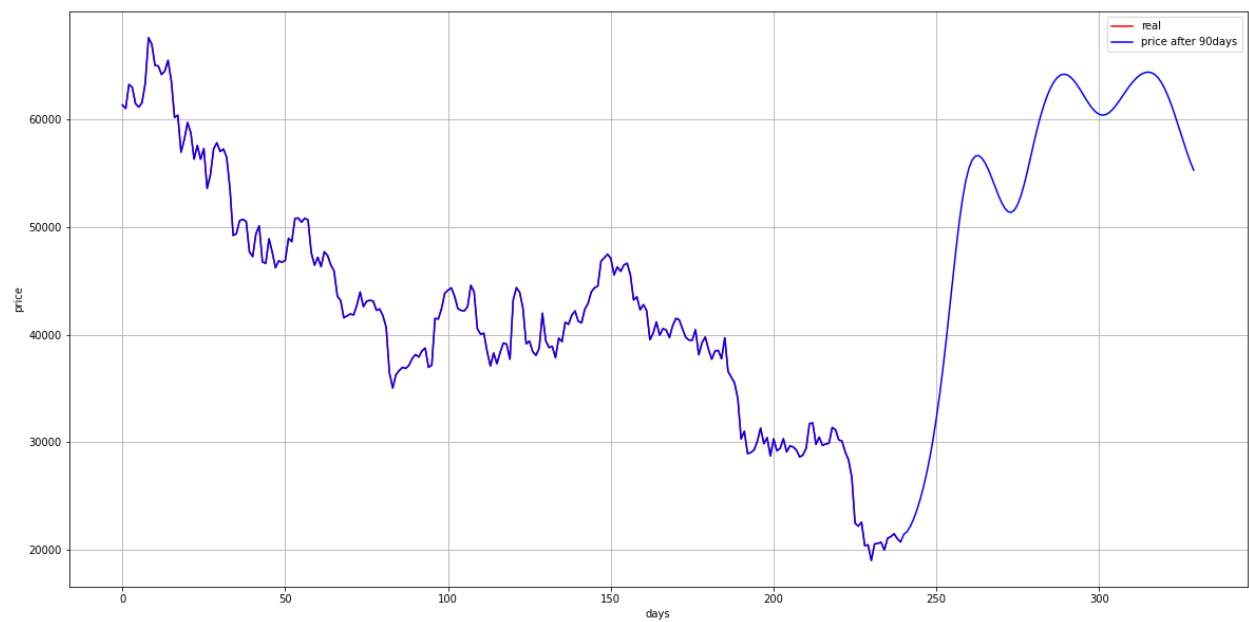
تفاوت پیش بینی بدست آمده با داده واقعی را در نمودار زیر مشاهده میکنید



حال نیاز است طبق خواسته سوال 90 روز آینده را پیش بینی کنیم پیش بینی 90 روز آینده را به اینصورت انجام شده است که ادامه y_{test} انجام شده است دیتاهای y_{test} 240 تا هست که از آخرین روز 60 تا 60 تا به جلو می رویم و هر مقدار جدید که پیش بینی می شود به داده y_{test} اضافه می شود درواقع پیشبینی 90 روز آینده برپایه پیشبینی مدل خودمان انجام شده است



این نمودار پیشبینی 90 روزه است



این نمودار هم از ادامه داده های تست است.

مقداری که ما 60 تعیین کرده ایم بسیار مهم است چرا که اگر خیلی زیاد باشد مدل دچار `gradient vanishing` می شود و اطلاعات داده های اولی از بین می رود و اگر کم باشد پیش بینی دقیقی نخواهیم داشت این متغیر در واقع حافظه مدل ما را تعیین میکند .

