

به نام خدا

پروژه کارشناسی

استاد پروژه: دکتر دهقانی

محمد طراوت

شرح پروژه:

هدف پروژه طراحی سیستم تحلیل اتوماتیک نمودارهای قیمتی بازار سهام و پیدا کردن سهام با توجه به خواسته های مطرح شده

این پروژه دارای چندین مرحله میباشد که به شرح زیر می باشد:

1- گرفتن داده های سهام از بازار بوسی و نمایش آن ها بر روی نمودار

2- پیاده سازی اندیکاتورهای زیر و اعلام موقعیت های خرید یا فروش با توجه به آن

اندیکاتور Aroon

اندیکاتور Bollinger band

اندیکاتور RSI

اندیکاتور میانگین متحرک نمایی

اندیکاتور macd

اندیکاتور alligator

3- رسم اندیکاتور ها بر روی نمودار رسم شده

4- پیدا کردن نقاط حمایت و مقاومت و مشخص کردن آن ها

5- پیدا کردن الگوهای زیر:

خط روند صعودی و نزولی

الگو مثلث

کانال صعودی و نزولی

6- رسم الگوهای مشخص شده بر روی نمودار

7- پیش بینی قیمت با توجه به Linear Regression و SVR

گرفتن داده های سهام از بازار بورسی و نمایش آن ها بر روی نمودار

با توجه به اینکه سازمان بورس ایران api رایگان ندارد و هزینه api بالا بوده و api های ارزان از سال ۹۸ به بعد داده ها را فقط در اختیار ما قرار می دادند مجبور به این شدیم که رباتی بنویسیم که خودش تمامی داده های موجود برای هر سهام را از سایت tsetmc دانلود کند برای این کار نیاز بود web scarping انجام دهیم که این قسمت مربوط به فایل

data mining tsetmc:

می باشد در این فایل توابع زیر موجود است که به توضیح هر تابع میپردازیم

geturl

start_scraping_download

list_csv

sortAndChangeColumns

geturl():

این تابع از داخل قسمت مربوط به سهام بورس نام و لینک مربوط به هر سهم را گرفته و ذخیره میکند

`start_scraping_download(int):`

این تابع که یک ورودی عدد صحیح دارد تعداد سهامی که لازم است داده آن ها را دانلود کند مشخص می کنیم با توجه به اینکه این داده ها روزانه تغییر میکنند و نیاز به تغییر لحظه ای نیست میتوان هر روز از این اسکریپت استفاده کرد و داده ها را به روز کرد

نوع صفحات سهام جوری نمی باشد که امکان استخراج اطلاعات داخل آن وجود داشته باشد ولی میتوان مشخص کرد که بعد از لود شدن صفحه به صورت اتوماتیک روی قسمت مورد نظر که با توجه به ایدی ان مشخص میشود کلیک کند این کار با توجه به سرعت اینترنت زمان گیری باشد که ماهیانه برای آن در نظر گرفته و مسیر دانلود را از قبل برای ان مشخص میکنیم

`list_csv():`

یک فایل به نام list ایجاد کرده که نام شرکت و سهام لینک و اسم فایل دانلود شده ذخیره کرده

`sortAndChangeColumns():`

داده های داندود شده از امروز به گذشته است در نتیجه هنگام رسم نمودار از زمان حال به گشت رفته پس لازم است آن را از گذشته به امروز مرتب کرده همچنین نام ستون ها فرمت دهی خود را دارد که بهتر است تغییر کرده

webapp:

این فایل که از کتابخانه streamlit استفاده میکنند به ما کمک کرده که یک وب اپلیکیشن ایجاد کرده
برای اجرای آن لازم است با دستور زیر آن را اجرا کرده

`streamlit run webapp.py`

در این فایل داده های دانلود شده را خوانده و نمودار آن را رسم کرده که یک نمودار خطی و یک نمودار به
صورت کندل استیک میباشد که برای آن یک دکمه اضافه شده و با کلیک به روی آن وارد نمودار آن
شده

پیاده سازی اندیکاتور های زیر و اعلام موقعیت های خرید یا فروش با توجه به آن

indicators :

این فایل توابع اندیکاتور ها میباشد که شامل:

Aroon

BollingerBands

Rsi

Ema

Macd

Alligator

و BackTest برای هر ۵ تابعی که داریم که از ابتدا تا امروز سیگنال های خرید و فروش را چاپ میکند

برای اندیکاتور های خواسته شده کتابخانه های آماده موجود میباشد که ما از

<https://github.com/mrjbq7/ta-lib>

<https://github.com/dmitriiweb/tapy>

استفاده کرده که اندیکاتور های خواسته شده را دارا میباشند

جهت آشنایی با اندیکاتور های خسته شده میتوان به

[Aroon](#)

[BollingerBands](#)

[Rsi](#)

[Ema](#)

[Macd](#)

[Alligator](#)

رسم اندیکاتور های خواسته شده بر روی نمودار:

یک فایل جدید به نام chart ایجاد کرده این فایل مربوط به تمام قسمتهای مورد نیاز برای رسمی باشد که شامل توابع

drawChart:

این تابع که تابع اصلی میباشد با دریافت ورودی های خود متوجه میشود چه نمودار هایی را باید رسم کند ابتدا از طریق تابع checkSelectForDraw متوجه میشود چند نمودار لازم است و با توجه به آن یک نمودار چند سطری ایجاد کرده که بتوان روی هر سطر نمودار های خواسته شده را پیاده کرد

checkSelectForDraw:

رسم نمودار لازم است یا روی نمودار کندل استیک باشد یا خارج از آن این تابع برای پیدا کردن نمودار های مورد نیاز خارج از نمودار کندل استیک میباشد

CandlestickChart:

برای رسم نمودار کندل استیک میباشد

alligatorChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نمودار کندل استیک آنها را رسم کرده

AroonChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نموداری خارج از کندل استیک آنها را رسم کرده

BollingerBandsChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نمودار کندل استیک آنها را رسم کرده

RsiChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نموداری خارج از کندل استیک آنها را رسم کرده

EmaChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نمودار کندل استیک آنها را رسم کرده

MacdChart:

اندیکاتور مورد نظر را فراخوانده و خروجی های آن را دریافت کرده و بر روی نموداری خارج از کندل استیک آنها را رسم کرده

پیدا کردن نقاط حمایت و مقاومت و مشخص کردن آن ها:

این قسمت شامل یک class میباشد که داخل فایل Sup_Res_Finder میباشد که دارای ۳ تابع

isSupport:

این تابع نقش پیدا کردن حمایت را دارد

isResistance:

این تابع نقش پیدا کردن مقاومت را دارد

find_levels:

این تابع هم حمایت مقاومت ها را داخل نمودار پیدا کرده و ذخیره میکند

برای رسم آن روی نمودار لازم است در قسمت فایل chart تابع زیر را اضافه کرده

def Support_ResistanceChart

این تابع با ایجاد یک شی از کلاس مورد نظر و ارسال داده ها به آن مقاومت ها و حمایت ها را پیدا کرده و ۲۰ تایی آخر آن را رسم کرده (برای جلوگیری از شلوغی نمودار)

پیدا کردن الگوهای زیر:

خط روند صعودی و نزولی

این قسمت مربوط به فایل trendline میباشد که شامل یک تابع با دو ورودی (داده ها و یک عدد که مربوط به اینکه روی چند کندل آخر به جستجو خط روند پردازیم میباشد)

همچنین برای این کار از کتابخانه trendln استفاده کرده ایم

خروجی این تابع نقاط شروع و پایان روند میباشد

برای نمایش خط روندها به فایل chart تابع TrendLineChart را اضافه کرده که به رسم خط روند ها پرداخته

الگو مثلث

این قسمت مربوط به فایل TrianglePatterns میباشد که شامل توابع

lastTriangle:

این تابع تمامی الگوهای مثلث را پیدا کرده و آخرین را باز میگرداند

Pivotid:

این تابع بررسی میکند که هر قسمت یا پیوت (pivot) یا همان نقطه چرخش قیمت است یا خیر

pointpos:

در صورت پیوت بودن قیمت را بر اساس نوع پیوت بر میگرداند

Triangle:

این تابع فراخوان تابع های دیگر است و پردازش مورد نیاز آماده سازی میکند و در آخر اگر مثلثی پیدا شود موقعیت آن را فرستاده در غیر این صورت عدد 1- را بر میگرداند

برای نمایش داخل فایل charts

از تابع TriangleChart استفاده کرده که در صورت وجود مثلث آن را رسم کرده

کانال صعودی و نزولی

این قسمت مربوط به فایل ChannelsDetection میباشد که شامل توابع

find:

که شامل دو ورودی میباشد یکی دیتا و دیگری عددی می باشد که (فرض ۱۰۰) که با این معنا میباشد داخل ۱۰۰ کندل اخر به دنبال یک کانال صعودی یا نزولی باش

برای رسم آن داخل فایل charts تابع ChannelBandsChart را اضافه کرده که با توجه به خروجی تابع find کانال را بر روی نمودار رسم میکند

پیش بینی قیمت با توجه به SVR و Linear Regression

برای این قسمت از فایل machine استفاده کرده همچنین نیاز است که از کتابخانه sklearn استفاده کرده که SVR و Linear Regression را آماده داخل خود دارد .

ML_SVR:

در این تابع ابتدا مقادیر none را از دیتا خارج کرده و با تابع SVR پیش بینی خود را کرده
این تابع شامل ورودی هایی میباشد که میتوان در این [قسمت](#) مطالعه کنید

ML_LR:

این تابع هم همانند تابع قبلی بوده با این تفاوت که از Linear Regression استفاده کرده که جهت
مطالعه ویژگی های آن میتوان به این [لینک](#) مراجعه کنید