**تجارت الگوریتمی - الگوریتم تجارت ایچیموکو**

**معرفی**

**هنگام معامله در بازارهای مالی، معامله گران می خواهند از هر ابزاری که می توانند برای انتخاب اطلاعات مهم از حجم عظیم داده های پر سر و صدا استفاده کنند. بسیاری از شاخص های فنی در بازارهای مالی به عنوان راهی برای تشخیص سیگنال های تجاری سودآور و همچنین ورود و خروج استفاده می شوند.**

**نمونه‌هایی از این شاخص‌ها از میانگین‌های متحرک شناخته‌شده تا شاخص‌های کمتر شناخته‌شده و تاریخی‌تر مانند اصلاحات فیبوناچی و چرخه‌های زمانی گان متغیر است.**

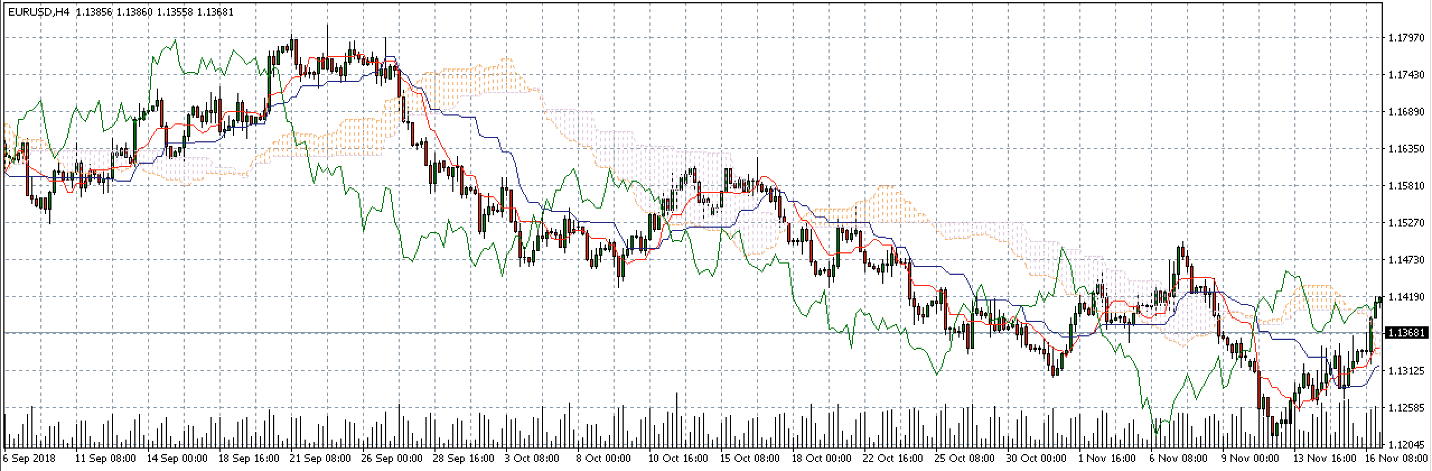
**هیچ شاخص فنی جام مقدس نیست، اما Ichimoku - نام کامل Ichimoku Kinko Hyo - به خودی خود بسیار خوب است.**

**در این پست، اندیکاتور ایچیموکو، اجزای آن را بررسی می کنیم و الگوریتم معاملاتی را می نویسیم و کدنویسی می‌کنیم که از سیگنال های ایچیموکو برای معامله استفاده می کند.**

**ایچیموکو کینکو هیو**

**Ichimoku Kinko Hyo به زبان ژاپنی تقریباً به معنای *نمودار تعادل یک نگاه* یک نشانگر فنی است که در دهه 1930 در ژاپن اختراع شد. این توسط فردی به نام گویچی هوسودا که جالب توجه بود، یک روزنامه نگار توسعه داده شد. ظاهراً توسعه این شاخص 30 سال طول کشید و در طول دهه 1960 برای عموم منتشر شد.**

**در اینجا تصویری از نشانگر Ichimoku در حال اجرا بر روی EUR/USD است.**

****

**در نگاه اول، این شاخص به طرز مضحکی کثیف و غیرضروری پیچیده به نظر می رسد، مانند یک کتاب درسی ریاضی. هر چند وقتی به نگاه کردن به آن عادت کنید، بسیار ساده تر و شهودی تر می شود. چندین بخش متحرک وجود دارد، اما می توان استدلال کرد که پیچیدگی نسبی آن باعث موفقیت آن بیشتر از چیزی مانند میانگین متحرک ساده می شود.**

**بیایید این شاخص را قسمت به قسمت تجزیه کنیم. Ichimoku Kinko Hyo دارای 6 جزء است:**

* **Tenkan-sen (خط قرمز): همچنین به عنوان *خط تبدیل* شناخته می شود ، این با اضافه کردن حداکثر قیمت 9 دوره گذشته به حداقل قیمت 9 دوره گذشته و تقسیم این مقدار بر 2 محاسبه می شود. از نظر میانگین متحرک. ، این را می توان به عنوان یک خط کوتاه مدت و سریعتر در نظر گرفت.**
* **Kijun-sen (خط آبی): که به عنوان *خط پایه* نیز شناخته می شود ، با افزودن حداکثر قیمت 26 دوره گذشته به حداقل قیمت 9 دوره گذشته و تقسیم این مقدار بر 2 محاسبه می شود. به عنوان یک نسخه کندتر و طولانی مدت Tenkan-sen.**
* **دهانه چیکو (خط سبز): همچنین به عنوان *خط عقب مانده* شناخته می شود ، این به سادگی قیمت هایی است که 26 دوره عقب تر از قیمت واقعی ترسیم شده اند. بنابراین، در یک نمودار روزانه، قیمت 26 نوامبر در نمودار در 1 نوامبر نشان داده می‌شود. نگاه کردن به این نمودار کمی گیج‌کننده‌تر است، اما زمانی که شروع به مطابقت با جایی که شاخص عقب است، آسان‌تر می‌شود.**
* **دهانه Senkou A (خط نارنجی مایل به رنگ): این با اضافه کردن تنکان سن و کیجون سن و تقسیم بر 2 محاسبه می شود. خط 26 دوره جلوتر ترسیم می شود، در جهت مخالف دهانه چیکو.**
* **دهانه سنکو B (خط مایل به ارغوانی): این با جمع کردن بالاترین بالاترین 52 دوره گذشته و کمترین پایین ترین دوره 52 دوره گذشته و تقسیم بر 2 محاسبه می شود. مانند دهانه سنکو A، 26 دوره قبل نیز ترسیم می شود. .**
* **دهانه Senkou A و B با هم یک ناحیه ابر شکل می‌سازند که پر می‌شود. این ناحیه به عنوان Kumo شناخته می‌شود . کومو به عنوان شاخص حمایت و مقاومت عمل می کند.**

**همچنین اعداد 9، 26 و 52 در سنگ تنظیم نشده اند. می‌توانید آن‌ها را به هر چیزی که مناسب می‌دانید تغییر دهید، اما من فکر می‌کنم بهتر است آنچه را که Goichi Hosoda پس از 30 سال توسعه انتخاب کرد، حفظ کنیم. دلیلی برای این اعداد خاص ارائه نشده است، یا حداقل من نتوانستم آن را پیدا کنم. با این حال، من هنوز آنها را خودسرانه نمی نامم. اگر مرد آن اعداد را انتخاب می کرد، باید به چیزی توجه می کرد.**

**سیگنال هایی که می توانیم تشخیص دهیم**

**اکنون که قطعات تشکیل دهنده نشانگر Ichimoku را درک کرده ایم، می توانیم ببینیم که چگونه می تواند سیگنال خرید یا فروش را نشان دهد.**

**هسته شاخص Ichimoku دو خط اصلی Tenkan-sen و Kijun-sen است. حرکت آهسته تر و کاهش واکنش پذیری به حرکت بازار از کیجون سن (از این رو خط پایه نامیده می شود) با حرکت سریع تر و واکنش پذیری بالا به بازار از Tenkan-sen (خط تبدیل) در تضاد است. این تفاوت در رفتار به دلیل تفاوت در دوره زمانی است که آنها در آن محاسبه می شوند (9 در مقابل 26 دوره). هنگامی که قیمت به بالا یا پایین می رود، این دو خط از روی یکدیگر عبور می کنند.**

**علاوه بر این، دهانه های Senkou محدوده خوبی برای افزایش یا کاهش قیمت به احتمال زیاد ارائه می دهند. البته، هیچ چیز قطعی نیست، و قطعاً نه در مورد بازارهای مالی، اما محاسبات با استفاده از 52 روز گذشته به نظر می رسد که پای خود را حفظ کند.**

**با استفاده از این اطلاعات، می توانیم چند ایده سیگنال را استنباط کنیم.**

**خرید سیگنال :**

**عبور Tenkan-sen از Kijun-sen نشان دهنده افزایش قیمت در کوتاه مدت است که نشانه خرید است.**

**علاوه بر این، اگر آنها بالاتر از Kumo باشند (به این معنی که عبور از بالای ابر اتفاق افتاده است)، این بدان معنی است که قیمت از بالاترین سطح 52 روز گذشته و به طور قابل توجهی به سمت بالا حرکت می کند. این آن را به یک سیگنال خرید قوی تبدیل می کند.**

**فروش سیگنال :**

**از سوی دیگر، عبور Tenkan-sen از Kijun-sen نشان می دهد که قیمت در کوتاه مدت به سمت پایین حرکت کرده است که سیگنالی برای فروش است.**

**به طور مشابه، اگر این متقاطع در زیر Kumo اتفاق بیفتد، به این معنی است که قیمت به طور قابل توجهی کاهش یافته است و سیگنال فروش قوی خواهد بود.**

**چیزهای بیشتری وجود دارد که می توانید با استفاده از این نشانگر به دست آورید، اما ما امروز خیلی از آن سوراخ خرگوش نمی گذریم.**

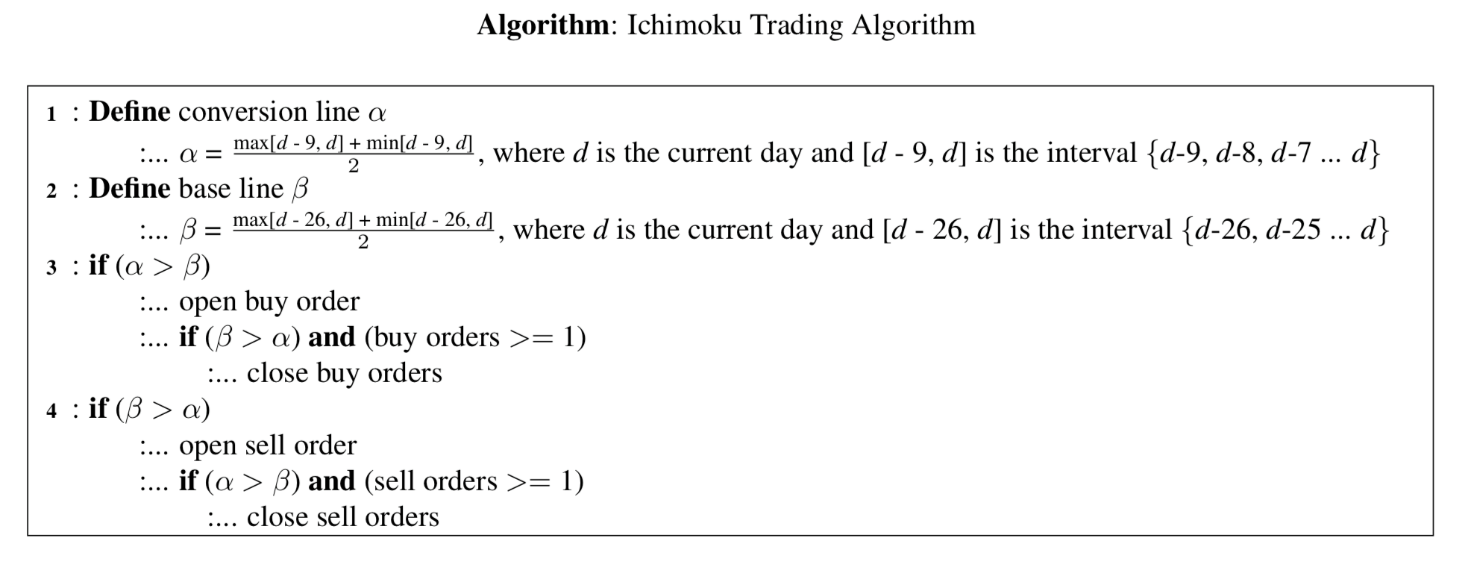
**الگوریتم**

**برای این آموزش، ما یک ربات تجاری بسیار پیچیده ایجاد نمی کنیم. بلکه بسیار ساده خواهد بود و از ابتدایی ترین سیگنال های ایچیموکو برای تجارت استفاده می کند. ایده این است که سادگی اما اثربخشی شاخص را برجسته کنیم و نشان دهیم که می توان از آن به صورت الگوریتمی استفاده کرد.**

**ما از کراس اوور اصلی تنکان سن و کیجون سن استفاده خواهیم کرد. وقتی خط تبدیل از خط پایه عبور کرد، ما خرید می کنیم و وقتی خط تبدیل از زیر خط پایه عبور کرد، می فروشیم.**

**برای بستن معاملات، دو گزینه داریم: یا می توانیم سود برداشت را تعیین کنیم - به این معنی که وقتی مقدار مشخصی پول به دست آوردیم، معامله را به طور خودکار می بندیم - یا می توانیم منتظر بمانیم تا خط تبدیل از خط پایه بازگردد. برعکس اگرچه ممکن است سود برداشت از نظر اقتصادی رضایت‌بخش‌تر باشد، اما انتخاب سودی که سود احتمالی ما را به حداکثر می‌رساند، بسیار طولانی خواهد بود و به تحلیل‌های آماری زیادی نیاز دارد. بنابراین، برای اختصار و به منظور نشان دادن نشانگر، ما به سادگی از متقاطع دوم به عنوان سیگنال نزدیک خود استفاده می کنیم.**

**بنابراین، به طور خلاصه، در اینجا الگوریتم قرار داده شده است:**

****

**تبدیل به کد**

**ما از MQL4 برای کدگذاری الگوریتم معاملاتی خود استفاده خواهیم کرد. MQL4**[**1**](https://osghaffar-github-io.translate.goog/Ichimoku-Algo-Trading/?_x_tr_sl=auto&_x_tr_tl=fa&_x_tr_hl=en-US&_x_tr_pto=nui#footnote1)**زبانی است که برای تجارت طراحی شده است، اما مبتنی بر C++ است و استفاده از آن چندان دشوار نیست، به خصوص اگر قبلاً با C++ آشنا هستید.**

**خوشبختانه، چندین توابع داخلی در MQL4 وجود دارد که به ما این امکان را می‌دهد تا از تایپ معادلات تبدیل و خط مبنا صرفنظر کنیم. با این حال، در صورتی که از زبانی استفاده می‌کنید که این زبان‌ها را ندارد، خوب است که آنها را در دسترس داشته باشید.**

**در اینجا الگوریتم اصلی داخل یک تابع آمده است:**

**void start()**

**{**

**int ticket = 0;**

**int order;**

**int orderclose;**

**int OrderTotal1 = OrderCounter(1111);**

**int OrderTotal2 = OrderCounter(2222);**

**double convline[9];**

**double baseline[26];**

**double conv = CopyClose(Symbol(), 0, 0, 9, convline);**

**double base = CopyClose(Symbol(), 0, 0, 26, baseline);**

**double maxOfConversionLine = maximum(convline, 9);**

**double minOfConversionLine = minimum(convline, 9);**

**double maxOfBaseLine = maximum(baseline, 26);**

**double minOfBaseLine = minimum(baseline, 26);**

**double conversionLine = (maxOfConversionLine + minOfConversionLine) / 2;**

**double baseLine = (maxOfBaseLine + minOfBaseLine) / 2;**

**if((conversionLine > baseLine) && (OrderTotal1 <= 2)){**

**ticket = OrderSend(Symbol(),OP\_BUY,LotSize,Ask,3,0,0,NULL,1111,0,Green);**

**for(int c = 0; c < OrdersTotal(); c++){**

**order = OrderSelect(c, SELECT\_BY\_POS);**

**if(OrderType() == OP\_SELL){**

**orderclose = OrderClose(OrderTicket(), OrderLots(), Ask, 3, Red);**

**}**

**}**

**}**

**if((baseLine > conversionLine) && (OrderTotal2 <= 2)){**

**ticket = OrderSend(Symbol(),OP\_SELL,LotSize,Bid,3,0,0,NULL,2222,0,Red);**

**for(int c = 0; c < OrdersTotal(); c++){**

**order = OrderSelect(c, SELECT\_BY\_POS);**

**if(OrderType() == OP\_BUY){**

**orderclose = OrderClose(OrderTicket(), OrderLots(), Bid, 3, Green);**

**}**

**}**

**}**

**}**

**کد نسبتاً ساده است، بنابراین می‌توانیم بدون مقدمه زیاد شروع به تجزیه آن کنیم.**

**چند خط اول کد بیشتر برای کمک به صاف کردن عملکرد ضروری است:**

**int ticket = 0;**

**int order;**

**int orderclose;**

**int OrderTotal1 = OrderCounter(1111);**

**int OrderTotal2 = OrderCounter(2222);**

**int ticket به سادگی راه را برای MQL4 برای پیگیری معاملات خود را است. هر معامله بعدی را که انجام می‌دهید شماره گذاری می کند، به عنوان مثال تجارت شماره 1، تجارت شماره 2 و غیره.**

**متغیرهای ما order و ما orderclose برای زمانی که توابع سفارش خود را بعداً قرار می دهیم در جای خود هستند. آنها نقش بسیار مهمی ندارند، بلکه یک تفاوت ظریف خاص MQL4 است.**

**OrderTotal بخشی از تابع دیگری است؛ به طور خلاصه، این فقط راهی برای پیگیری تعداد معاملات باز شده است.**

**در ادامه، این چند خط را داریم:**

**double convline[9];**

**double baseline[26];**

**double conv = CopyClose(Symbol(), 0, 0, 9, convline);**

**double base = CopyClose(Symbol(), 0, 0, 26, baseline);**

**در اینجا ما خطوط تبدیل (Tenkan-sen) و پایه (Kijun-sen) خود را تعریف می کنیم. همانطور که قبلاً گفتم، MQL4 دارای چندین توابع داخلی است که محاسبات را برای ما انجام می دهد. با این حال، ما نحوه پیاده سازی الگوریتم را به صورت دستی بررسی خواهیم کرد.**

**ما 2 آرایه ایجاد خواهیم کرد، یکی برای خط تبدیل و دیگری برای خط مبنا. از آنجایی که می دانیم در هر آرایه چند عنصر (9 و 26) وجود دارد، می توانیم هنگام تعریف آنها را به صورت ایستا تخصیص دهیم.**

**راه پر کردن آرایه ها از طریق تابعی به نام است CopyClose که آخرین x قیمت ها را در آرایه ما برای ما کپی می کند.**

**با توجه به اسناد**[**2**](https://osghaffar-github-io.translate.goog/Ichimoku-Algo-Trading/?_x_tr_sl=auto&_x_tr_tl=fa&_x_tr_hl=en-US&_x_tr_pto=nui#footnote2)**، پارامترها به شرح زیر است:**

**int CopyClose(string symbol\_name, int timeframe, int start\_pos, int count, close\_array[]);**

**با استفاده از این، می‌توانیم آن را با اطلاعاتی که داریم پر کنیم - هیچ نماد یا جفت ارز خاصی نمی‌خواهیم، ​​بنابراین می‌توانیم آن را به عنوان NULL بگذاریم. بازه زمانی ما تایم فریم فعلی است، بنابراین آن را به عنوان 0 می گذاریم. مقدار شروع ما جدیدترین مقدار است، به این معنی که 0 را نیز قرار می دهیم. تعداد ما 9 و 26 است، بنابراین برای یک نمونه از تابع، 9 و برای ما 26 را قرار می دهیم. برای آرایه نزدیک خود، نام آرایه خود را به سادگی می گذاریم - خط تبدیل یا خط پایه.**

**حالا این قسمت را داریم:**

**double maxOfConversionLine = maximum(convline, 9);**

**double minOfConversionLine = minimum(convline, 9);**

**double maxOfBaseLine = maximum(baseline, 26);**

**double minOfBaseLine = minimum(baseline, 26);**

**double conversionLine = (maxOfConversionLine + minOfConversionLine) / 2;**

**double baseLine = (maxOfBaseLine + minOfBaseLine) / 2;**

**در اینجا ما آرایه های خود را به توابع ساده ماکزیمم و مینیمم منتقل می کنیم تا مقادیر حداکثر و حداقل مورد نیاز برای الگوریتم را بدست آوریم. من توابع را در اینجا وارد نمی کنم، اما آنها تکرار کننده اصلی شما در آرایه برای یافتن مقادیر حداکثر و حداقل هستند.**

**پس از آن، معادله خود را می نویسیم و مقادیر حداکثر و حداقل خود را وصل می کنیم.**

**علاوه بر این، یک راه ساده تر و داخلی برای انجام این کار وجود دارد - می توانیم از تابع iIchimoku استفاده کنیم:**

**با نگاهی به مستندات**[**3**](https://osghaffar-github-io.translate.goog/Ichimoku-Algo-Trading/?_x_tr_sl=auto&_x_tr_tl=fa&_x_tr_hl=en-US&_x_tr_pto=nui#footnote3)**، پارامترها به شرح زیر است:**

**iIchimoku(Symbol, Timeframe, Tenkan-sen, Kijun-sen, Senkou Span B period, Data Source, Time Shift)**

**با استفاده از این، می‌توانیم آن را با اطلاعاتی که داریم پر کنیم - هیچ نماد یا جفت ارز خاصی نمی‌خواهیم، ​​بنابراین می‌توانیم آن را به عنوان NULL بگذاریم. بازه زمانی ما زمان فعلی است، بنابراین آن را 0 می گذاریم. سپس، سه مقدار بعدی را به ترتیب با 9، 26 و 52 جایگزین می کنیم. منبع داده این است که کدام خط را به عنوان مقدار بازگشتی می خواهید - خط تبدیل، خط پایه یا یکی از خطوط Senkou Span. برای این کار 1 را در خط تبدیل و 2 را برای خط پایه قرار می دهیم. در نهایت، ما به هیچ وجه نمی خواهیم زمان را تغییر دهیم، بنابراین آن را 0 نگه می داریم.**

**این ما را نتیجه خواهد داد**

**double conversionLine = iIchimoku(NULL, 0, 9, 26, 52, 1, 0);**

**double baseLine = iIchimoku(NULL, 0, 9, 26, 52, 2, 0);**

**به جای این همه کار خوبه، نه؟**

**این قسمت آخر کد نیز نسبتاً ساده است:**

**if((conversionLine > baseLine) && (OrderTotal1 <= 2)){**

**ticket = OrderSend(Symbol(),OP\_BUY,LotSize,Ask,3,0,0,NULL,1111,0,Green);**

**for(int c = 0; c < OrdersTotal(); c++){**

**order = OrderSelect(c, SELECT\_BY\_POS);**

**if(OrderType() == OP\_SELL){**

**orderclose = OrderClose(OrderTicket(), OrderLots(), Ask, 3, Red);**

**}**

**}**

**}**

**if((baseLine > conversionLine) && (OrderTotal2 <= 2)){**

**ticket = OrderSend(Symbol(),OP\_SELL,LotSize,Bid,3,0,0,NULL,2222,0,Red);**

**for(int c = 0; c < OrdersTotal(); c++){**

**order = OrderSelect(c, SELECT\_BY\_POS);**

**if(OrderType() == OP\_BUY){**

**orderclose = OrderClose(OrderTicket(), OrderLots(), Bid, 3, Green);**

**}**

**}**

**}**

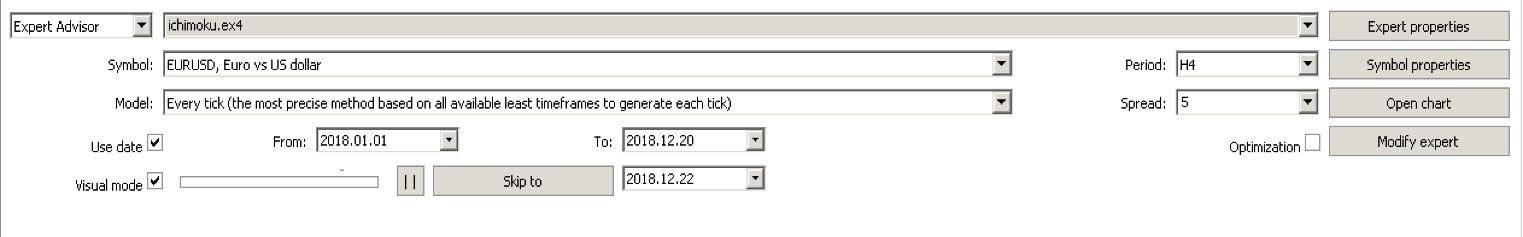
**اگر خط تبدیل بیشتر از خط مبنا باشد، خرید می کنیم. اگر خط پایه بیشتر باشد، می فروشیم. مجموع سفارش تعداد معاملاتی را که قبلاً باز کرده‌ایم ردیابی می‌کند تا از باز کردن مبلغ غیرضروری جلوگیری کند.**

**سپس تمام سفارش‌های باز فعلی را مرور می‌کند و اگر یک سیگنال خرید باشد، هر سفارش فروش را می‌بندد، و اگر سفارش فروش باشد، هر سفارش خرید را می‌بندد.**

**نتایج**

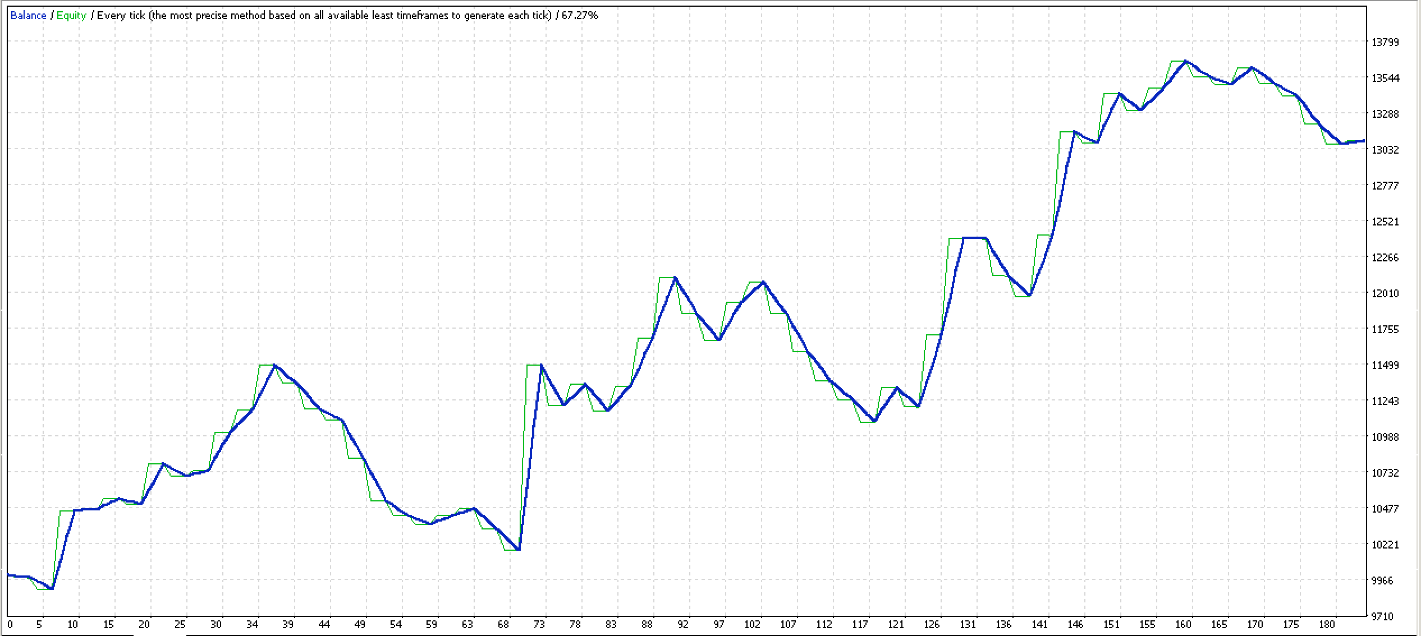
**اکنون که الگوریتم خود را در کد پیاده سازی کرده ایم، بیایید یک بک تست انجام دهیم تا ببینیم معامله گر Ichimoku ما چگونه عمل می کند. شما می توانید**[**کد کامل را در github من مشاهده کنید.**](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://github.com/osghaffar/Algo-Trading/tree/master/Ichimoku%2520Trading%2520Algorithm)

**در اینجا شرایطی وجود دارد که من برای آزمایش الگوریتم استفاده می کنم:**

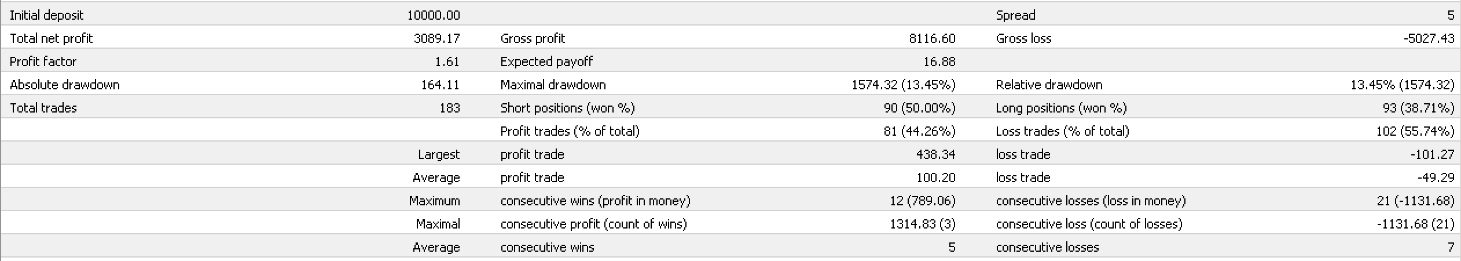
****

**من از EUR/USD در نمودار 4 ساعته استفاده می کنم (بنابراین تمام دوره های ما که ایچیموکو برای محاسبه استفاده می کند 4 ساعت خواهد بود) و از 1 ژانویه 2018 تا 20 دسامبر 2018 آن را آزمایش می کنم. شروعی دارد. موجودی 10000 دلار است و از اندازه زیادی 0.1 استفاده می کند. من از اسپرد 5 هم استفاده می کنم، فقط برای اینکه زیاد در عملکرد الگوریتم ما اختلال ایجاد نکند. در حال حاضر، به جای استحکام بیشتر به عملکرد آن در بازار اهمیت می دهم.**

**در اینجا نتایج آمده است:**

****

**و در اینجا گزارش مفصل تر است:**

****

**می بینیم که الگوریتم سود خالص 3089 دلار را به دست آورد که حدود 31٪ بازدهی در سال است. اصلا بد نیست! اما مطمئناً پیشرفت هایی وجود دارد که می توان انجام داد.**

**یک مثبت قطعی این است که حداکثر برداشت 13.45٪ بوده است و در نمودار سود می توانید ببینید که حقوق صاحبان سهام (خط سبز) هرگز به زیر مانده حساب خیلی نزول نکرده است. این به این معنی است که ما معامله بزرگی با زیان انجام ندادیم.**

**علاوه بر این، معاملات کوتاه ما فقط در 50٪ مواقع موفق بودند و معاملات طولانی ما فقط در 39٪ موارد موفق بودند. این واقعیت که ما هنوز از این اعداد سود می‌بردیم، تعجب‌آور است و به قدرت سیستم ایچیموکو اشاره می‌کند، زیرا بزرگترین تجارت سود ما (438 دلار) تقریباً 4 برابر بیشتر از بزرگترین معامله با ضرر ما (-101 دلار) بود.**

**نتیجه**

**برای جمع بندی باید بگویم که این سیستم قطعا موثر است و پتانسیل بالایی دارد. به نظر می رسد که 9، 26 و 52 یک نقطه شیرین قطعی هستند، زیرا آنها به سیستم اجازه می دهند تا نسبت به بازار واکنش نشان دهد - نه خیلی واکنشی، جایی که وقتی نوسان وجود دارد معامله را آغاز کند، و نه *به* اندازه کافی واکنش نشان می دهد. جایی که شما معاملات سودآور را از دست می دهید.**

**همچنین تا آنجا پیش می‌روم که بگویم این یکی از تنها سیستم‌های معاملاتی است که می‌تواند به خودی خود سودآور باشد، بدون اینکه شاخص‌های دیگری در آن دخالت داشته باشند. به نظر می رسد**[**این آزمایش انجام شده توسط BabyPips**](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://www.babypips.com/learn/forex/what-is-the-most-profitable-indicator)**این موضوع را تأیید می‌کند، زیرا آنها سیستم معاملاتی بسیار مشابهی را انجام دادند که ما در این پست وبلاگ انجام دادیم.**

**من همچنین نمی دانم که چه ترفندهایی را می توان برای بهبود سیستم انجام داد - به ویژه، چگونه می توانیم به طور موثر از ابر Kumo به عنوان یک سیگنال تجاری استفاده کنیم. امکانات زیادی وجود دارد و ایچیموکو مطمئناً یک شاخص امیدوارکننده است.**

**شما چی فکر میکنید؟**

**منابع**

1. [**مستندات MQL4**](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://docs.mql4.com)
2. [**اسناد MQL4 CopyClose[]**](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://docs.mql4.com/series/copyclose)
3. [**اسناد MQL4 Ichimoku**](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://docs.mql4.com/indicators/iichimoku)
4. **[بهترین شاخص فنی چیست؟ - بیبی پیپس](https://translate.google.com/website?sl=auto&tl=fa&nui=1&u=https://www.babypips.com/learn/forex/what-is-the-most-profitable-indicator)**