Két fő iránnyal foglalkoztam a félév során, ezeket szeretném bemutatni:

Először segítő scripteket írtam, ezek azért kellettek, hogy könnyebben és gyorsabban csinálhassak teszteket, könnyebb legyen kiértékelni őket.

Az első script amivel foglalkoztam, az egy batch tester. A MetaTrader 4-et indítja el a megfelelő bemeneti paraméterekkel ahhoz, hogy egy backtest lefusson. A backtest lefutása után a MetaTrader bezáródik, az eredményt a script a megadott mappába másolja. Kitaláltam hozzá egy saját jelölésrendszert, amivel a MetaTrader saját teszt parameter fájlját módosítva több tesztet is tudok futtatni egymás után. Ennek készítettem egy readme-t is, mert ez egy olyan script ami szerintem hasznos lehet másoknak is.

A második script az eredményeket sorolja modelling quality és profit szerint mappákba. Ez egy sokkal kisebb volumenű dolog, de a kiértékelésben sokat segített. Nem kellett megnéznem a hibás, rosszul sikerült teszteket.

A két segítő scriptnek csináltam helpet, ezért a használatukat nem részletezném, az implementáció pedig triviális.

Ezzel a két scripttel kíváncsiságból teszteltem kb. 10-15 letöltött Expert Advisort, ezek közül volt, ami nyereséges volt(!), de a nagyrészükre igaz, hogy csak keveset veszít.

Ezek után főleg olvastam, videókat néztem arról, hogy a standard indikátorok nélkül hogyan lehetne megjósolni az árfolyamot. Itt döntöttem el, hogy csak az EUR/USD chartból szeretnék kiszűrni egy jóslatot, nem fogok tudni a devizapárok közötti korrelációval is foglalkozni. Először megpróbáltam Fourier transzformációval extrapolálni az árfolyamot, ez a nagyfrekvenciájú áringadozások miatt nem adott értékelhető eredményt. Aztán egy kis kutakodás után láttam, hogy ha lesimítom az árakat valamilyen mozgó átlaggal, akkor sokkal jobban meg tudom becsülni a mozgó átlagot. Ezt úgy alakítottam, hogy egy viszonylag kis periódusú (5) EMA-t használtam, amit FFT-vel transzformáltam. Ezzel két baj volt: Fourier transzformációval nagyon „overfitted” lett a kinyert függvény, másfelől pedig a hirtelen nagy változásoknál a függvény előrejelzett és bemeneti értékei kb. szimmetrikusak voltak. Ezt úgy akartam kiküszöbölni, hogy adott varianciaugrásnál ne kössön az algoritmus. Itt hagytam abba a Fourier transzformáció próbálgatását, mert találtam egy jobb megoldást.

Az utolsó verzió a Time Series Decompositionön alapszik. A lényege, hogy egy time series felbontható egy trendre és egy szezonális tényezőre. Itt persze feltételeztem, hogy az árfolyam egy time series, ezt nem tudom sajnos sem bizonyítani, sem cáfolni. A logikát r-ben írtam, ez a legkézenfekvőbb ilyesfajta adatelemzéshez. A kereskedőprogram tehát két részből áll. Van egy szerver, ami a MetaTraderben fut, folyton küldi a hisztorikus adatokat az r-scriptnek, ami a kliens. A kliens az adatok alapján előrevetít egy becslést az árról és eldönti, hogy venni, vagy eladni érdemesebb. A két programot ZeroMQ TCP kommunikációval kötöttem össze, ezzel volt a legnagyobb probléma, mert sajnos a dokumentáció sem az MQL4 sem az r nyelven nem teljes.

A logika működése nem túl bonyolult: a hisztorikus adatokat (100 bar) STL dekompozícióval szétbontom a trend, seasonal és remainder részekre, ezeket extrapolálom, majd újra összeadom, ezzel megkapom az előrejelzést 12 bar távolságig. Ha az előrejelzésben nincs a megadott Stop Loss értéknél kisebb, viszont van a megadott Take Profit értéknél nagyobb, akkor egy buy signal keletkezik, fordítva egy sell signal, a maradék esetben pedig a javaslat, hogy ne tegyünk kötést. A szerver egy Timer eseményre végzi a dolgát, a timer intervalluma alapján állítható a válaszideje a rendszernek. A kliens kéréseire reagálva vagy a hisztorikus adatokat küldi, vagy azt reagálja, hogy nincs változás az előző kéréshez képest. Az Expert Advisor (szerver) backtestelésénél a timer eventek nem váltódnak ki, ezért kiegészítettem az OnTick() metódust egy explicit OnTimer() hívással, így a program backtestelhető.

Az eredmények sajnos mind a Fourier transzformáció felhasználásával, mind az STL dekompozícióval elmaradnak a várttól. A Fourier transzformáció írásakor csináltam egy Indicatort, ez az előzőekhez hasonló eredményt hozott. Az STL dekompozíciós megoldást viszont szerintem érdemes továbbgondolni, sajnos csak az utolsó pár hétben kezdtem ismerkedni vele így nem tudom kijavítani, hiába látom, hogy nem jó az eredmény.

Az egész projekt és a források elérhetők a githubon: <https://github.com/bendicsekb/expert_advisors>