A nyárigyakorlat első két hetében a célkitűzés egy előre megkapott adathalmaz feldolgozása és megjelenítése volt. A feladat megoldása SAP Hana, WEBIde környezetében kellett, hogy történjen. Az első napokban internetes dokumentációkból, leírásokból, videókból ismerkedtem meg az SAP-val és a hozzá köthető fogalmakkal. Az első héten táblák létrehozásával foglalkoztam. A hivatalos oldal nem működött, ezért ott nem sikerült létrehozni a feladathoz szükséges táblákat. A második héten kaptuk meg az elérést az egyetemi szerverhez, amin sikerült létrehoznom a táblát. A nyers táblázatot ezután feldolgoztam kiszedtem belőle a fölösleges információkat, ismétlődéseket. Például nem szükséges, hogy egy régió neve és kódja egyszerre szerepeljen egy sorban. Ezeket a kapcsolatok más, kisebb táblázatban tároltam. Az adatok megjelenítéséhez szükséges volt egy ODATA létrehozására, ami JSON formátomú. Az adatok vizuális megjelenítéséhez Angular-t használtam. Ezekhez mind szükséges volt a javascipt egy bizonyos fokú ismeretére ezért egy napot annak a tanulásával töltöttem. Ezeknek a moduloknak a befejezése után létrehoztam a kalkulációs nézetet WEBIDE-ben. A megjelenítésben eredmény szempontjából nem számított, de gyakorlatnak hasznos volt, hogy a lekérdezéseket hogyan oldom meg. Ezért csináltam példát arra, hogy javascript dolgozza fel az eredeti táblát, és arra is hogy WEBIde-ben hoztam létre a táblát.

Az elkészült munka bemutatása után az SAP PAL könyvtárral, és az adatbányászattal kezdtem el foglalkozni. Utána olvastam annak, hogy mi az adatbányászat, és milyen céljai és eszközei vannak. Elolvastam a PAL dokumentációját és tanulmányoztam a használható eljárásokat. Megpróbáltam a dokumentációban látható módon reprodukálni a procedúrák meghívását, de az sajnos jogosultságok hiánya miatt nem sikerült, és később, mint kiderült nem is az ott olvasottak alapján kell meghívni. Egy a tanár által küldött tutorialon szereplő módszert alkalmazva sajátítottam el a procedúrák meghívását. Az elején igazából úgy se működött, de a script szerver beállítása után megtudtam csinálni és meghívni az ARIMA procedúrát, egy a tutorialban elérhető vásárlási adatokra. Az ARIMA időben sorrendben állítható adatok esetén ad előrejelzést. Egy procedúra meghívása úgy néz ki, hogy a procedúra által szükséges tábla típusokat (table type) és tábla entitiket (entity) létrehozzuk, a táblatípusokkal definiáljuk a függvényt, majd sql kódban megírjuk, hogy hogyan alkalmazza a táblákat az eljárás. A tábláknak a pontos beállítását SAP PAL dokumentációjából olvastam ki. Mivel nem a hivatalos dokumentációban szereplő módszert alkalmaztam, hanem azt kellett átültetni egy másikra ezért kezdetben ez egy lassú folyamat volt.

Az utolsó részében a gyakorlatnak egy a levegőt vizsgáló szenzorok eredményére kellet PAL procedúrát meghívni. Az adatok mennyisége és a táblázat mérete miatt szükséges volt c++ kód írására. Annak a segítségével hoztam létre egy txt fájlt, amiben olyan kód szerepelt, amit bemásolhattam a WEBIde-be, és ezzel létrehozva a táblázatot. Továbbá szükség volt egy ID oszlop létrehozására SQL kód segítségével, mert mindegyik procedúra előfeltétele ez az oszlop. Az eredeti adataink egész éves adatokat tartalmaztak én annak a januári részével foglalkoztam. A procedúrák csak olyan adaton lehet futtatni, ami nem tartalmaz NULL adatot, ezért SQL segítségével kiválasztottam 5 olyan érzékelő adatait, amelyeknek minden sora ki volt töltve. Ehhez az öt szenzorhoz tartozó mérési adatokra, egy két fős csapatban futtattunk le 3 darab procedúrát. Az Affinity Propagation egy klaszterező algoritmus, amit sikeresen lefuttattunk a végeredménye 31 klasztert talált. Az ARIMA procedúra egy outlireket kereső algoritmus, amit egy szenzor hőmérsékleti adataira futtatunk le. A működésének vizsgálata során azt tapasztaltuk, hogy ha kettő vagy annál nagyobb hőmérsékletváltozás történt a szomszédokhoz képest akkor az outlier-nek számított. Az ARIMA algoritmus egy előre jelző algoritmus, amit lefuttattunk több általunk tetszőlegesen választott értékre.