

Fejlesztői dokumentáció

Adatbázis menü

A feladatunk az volt, hogy válasszunk ki egy adatbázist a számunkra vonzónak talált opciók közül, majd egy Java FX alkalmazást hozzunk létre a kiválasztott adatbázissal. Az alkalmazásnak különböző menüpontokkal kellett rendelkeznie, és az első lépésben egy CRUD (Create, Read, Update, Delete) funkcionalitást kellett implementálnia.

Az első almenü, az "Olvas", lehetővé tette az adatbázisban található információk kilistázását egy táblázatban. Amennyiben az adatmennyiség túl nagy volt, a listázást lehetőség volt rövidíteni a felhasználó kényelme érdekében.

A következő almenü, az "Olvas2", hasonló volt az előzőhöz, de itt már egy űrlapot is magába foglalt. Ez az űrlap lehetővé tette a felhasználó számára a könnyebb szűrési lehetőségeket, hogy pontosan azokat az elemeket láthassa, amelyek érdeklik.

A harmadik almenü, az "Ír", lehetőséget biztosított a felhasználónak új rekordok hozzáadására az adatbázishoz. A felhasználó itt adhatott meg új információkat a rendszerbe.

A "Módosít" almenü lehetővé tette a már meglévő rekordok szerkesztését az azonosító (ID) alapján. Ezáltal a felhasználó módosíthatta a meglévő adatokat az igényeinek megfelelően.

Az utolsó almenü, a "Töröl", lehetővé tette a felhasználónak, hogy megadott azonosító alapján törölje a kiválasztott rekordot az adatbázisból.

Ez a Java FX alkalmazás tehát teljes körű CRUD műveleteket kínált a választott adatbázissal, a könnyű adatkezelés és szűrés érdekében.

Rest1 menü

A második feladatban egy Restful kliens létrehozása volt a cél, ami kapcsolódott a bemutatott gores.co.in REST szerverhez. A kliensnek implementálnia kellett a Create, Read, és Delete műveleteket, valamint ezekhez kapcsolódó funkcionalitásokat.

A "Create" művelet lehetőséget adott arra, hogy a felhasználó neve, neme, e-mail címe és státusza alapján új felhasználót hozzunk létre. A szerver ezenkívül generált egy egyedi azonosítót (ID-t), amelyet a létrehozott felhasználóhoz rendelt.

A "Read" almenü lehetővé tette a felhasználók kilistázását, beleértve az általunk létrehozottakat is. Ezen keresztül láthattuk a szerver által generált egyedi azonosítókat és az egyes felhasználók adatait.

A "Delete" művelet segítségével egy adott azonosító alapján törölni lehetett egy felhasználót. Ezt követően a "put" módszerrel lehetőség nyílt egy már meglévő felhasználó adatainak módosítására.

Ez a Restful kliens tehát lehetővé tette a teljes CRUD műveletsorozat használatát a gores.co.in REST szerverrel, így hatékonyan kezelhettük az adatok létrehozását, olvasását, módosítását és törlését

SoapKliens menü:

A harmadik feladat során egy SOAP kliens létrehozása volt a cél, amelyet a Magyar Nemzeti Bank webszolgáltatásához kellett kapcsolnunk.

Az "Letöltés" almenü lehetőséget biztosított az összes adat letöltésére a Magyar Nemzeti Bank webszolgáltatásáról, majd ezeket az adatokat a helyi adatbázisunkba mentettük.

A "Letöltés 2" almenü lehetővé tette számunkra, hogy kiválasszuk azokat az adatokat, amelyeket csak le szeretnénk tölteni a webszolgáltatásról, és ezeket ismét a helyi adatbázisunkba mentettük. Ezáltal célzottabban frissíthettük a helyi adatbázisunkat a szükséges információkkal.

A harmadik almenü a "Grafikon" lehetőséget adott arra, hogy a letöltött adatok közül kiválasszuk, melyeket szeretnénk grafikusán megjeleníteni. Ezáltal a felhasználó dönthetett, hogy mely adatokat szeretné vizuálisan reprezentálni, segítve ezzel a könnyebb megértést és elemzést.

Ez a SOAP kliens tehát lehetővé tette a Magyar Nemzeti Bank webszolgáltatásának hatékony használatát, ahol a letöltött adatokat lokális adatbázisunkban tároltuk, majd ezeket grafikusán is megjeleníthettük a kiválasztott adatok alapján.

Adatbányászat menü:

A negyedik feladatunkban a felsorolt adatbázisok segítségével gépi tanuló algoritmusokat kellett alkalmazni és ezek közötti különbségeket kiemelni. A választott algoritmusok közé tartozott az IBk, J48, SMO, NaiveBayes. A feladat része volt egy Döntési fa almenü létrehozása, ahol egy gomb segítségével döntési fát kellett generálnunk, majd az eredményeket egy "döntésifa.txt" állományba kiírni. Az eredmények közé tartoztak a TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative), Correctly Classified Instances, Incorrectly Classified Instances értékek.

A második almenü a "Több algoritmus" lehetőséget biztosította, ahol az összes algoritmus típust lefuttattuk, majd az eredményeket egy "GépiTanulás.txt" állományba mentettük.

A harmadik almenü, a "Több algoritmus almenü-2", egy lenyíló listát tartalmazott 1-től 5-ig, ami a különböző algoritmus típusokat reprezentálta. A felhasználó választása alapján a program létrehozott egy "seged.txt" állományt, majd ebből kiolvasta az értékeket, és megjelenítette azokat a képernyőn.

Ezáltal a feladat célja a gépi tanuló algoritmusok hatékonyságának bemutatása és a különböző típusok közötti összehasonlítás lehetősége volt, a különböző eredmények részletes elemzése és dokumentálása révén.

Egyéb menü:

Az ötödik feladat két fő részből állt. Az első részben egy Párhuzamos almenüt kellett létrehozni, amely segítségével bemutattuk a szálak (thread-ek) működését. Például egy gombra kattintva egy Label szövegét kellett megváltoztatni, úgy hogy egy másodpercenként egy, majd egy másik esetben pedig két másodpercenként változtassa a szöveget. Ezáltal a feladat a párhuzamos feldolgozás és szálak aszinkron működésének gyakorlati példáját mutatta be.

A második részben egy Stream almenüt kellett kialakítani, amely segítségével betöltöttük az adatbázisunkat és különböző szűrők alkalmazásával kilistáztuk az elemeket. Ez a rész a Java Stream API-t használva mutatta be az adatok funkcionális programozási elemekkel történő hatékony kezelését. A Stream lehetővé tette az adatok gyors és hatékony feldolgozását, valamint különböző szűrők alkalmazását az eredmények pontosabb kiválasztása érdekében.

Ezáltal a feladat célja volt a párhuzamos programozás és a funkcionális programozás alapjainak megértése, valamint ezek alkalmazása a gyakorlatban egy Java alkalmazásban.

Forex menü

Az utolsó programhoz kapcsolódó feladatunk során az Oanda API felhasználásával különböző almenük létrehozása volt a cél.

A "Számlainformációk" almenü lehetővé tette a számlainformációk lekérdezését és ezek kiírását egy táblázatba. Ezen keresztül a felhasználó betekintést nyerhetett a számla aktuális állapotába.

Az "Aktuális árak" almenüben lenyíló listából választhatta ki a felhasználó a kívánt devizapárt, majd az aktuális árat kiíratta a képernyőre. Ezáltal a felhasználó könnyen és gyorsan tájékozódhatott a kiválasztott devizapár aktuális helyzetéről.

A "Historikus árak" almenü lehetővé tette a felhasználó számára, hogy lenyíló listákból válassza ki a devizapárt, valamint megadja a kezdő és záró dátumokat. A program ennek alapján egy táblázatban kiírta a két dátum közötti árakat, és további lehetőséget nyújtott ezek grafikonon történő ábrázolására.

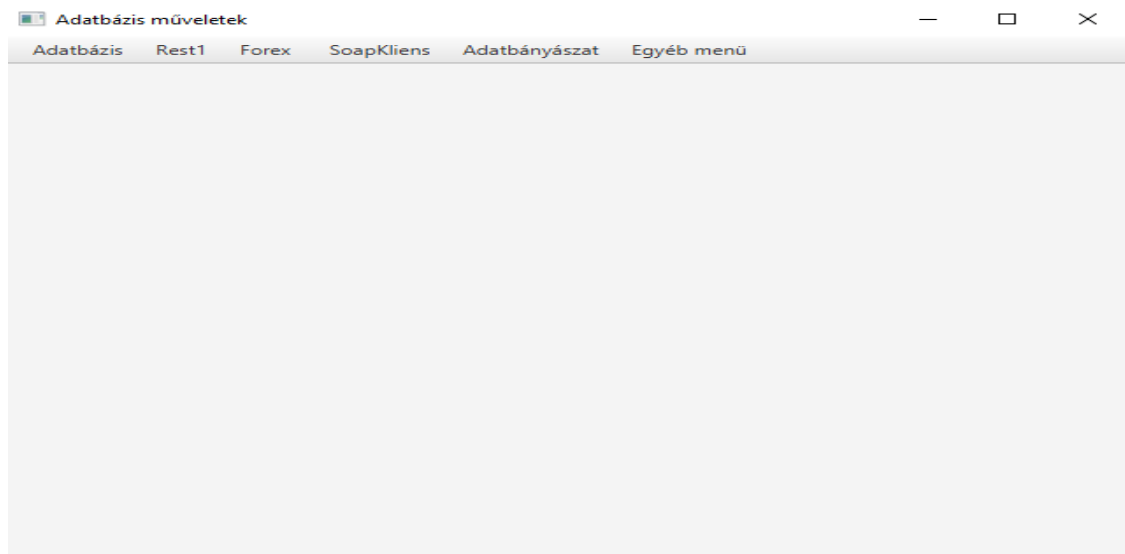
A "Pozíció nyitás" almenü lehetővé tette a devizapár, mennyiség és irány kiválasztását egy lenyíló listából, majd a felhasználó által megadott paraméterek alapján megnyitotta a kívánt pozíciót.

A "Pozíció zárás" almenü lehetőséget biztosított a felhasználónak a pozíció ID-jának megadásával a kiválasztott pozíció zárására.

A "Nyitott pozíciók" almenü egy táblázatba kiírta a nyitott pozíciókat, így a felhasználó könnyen nyomon követhette az aktuális pozícióit.

Ezáltal a feladatok révén a program részletes és interaktív kezelési lehetőségeket kínált az Oanda API segítségével, megjelenítve és kezelve a számlainformációkat, aktuális és historikus árakat, valamint lehetőséget adva a pozíciók nyitására és zárására.

GitHub URL : <https://github.com/GreMre/bead.git>



Továbbfejlesztési Lehetőségek a Jövőben:

1. Felhasználói Felület Fejlesztése:

- További interaktivitás és felhasználói élmény javítása érdekében gondold a grafikus felhasználói felület (GUI) fejlesztésére. Egy jól tervezett, felhasználóbarát interfész növelheti a program használhatóságát.

2. Érthető Grafikonok:

- Az ábrázolt grafikonok továbbfejlesztése, hogy még könnyebben értelmezhetők legyenek. Használj különböző típusú diagramokat vagy hozz létre interaktív grafikonokat, amelyek lehetővé teszik az adatok részletes megjelenítését.

3. Automatizált Tesztek:

- Tesztelési keretrendszer bevezetése a stabilitás és a megbízhatóság növelése érdekében. Automatizált tesztekkel ellenőrizheted, hogy a kódod még a kisebb változtatások után is jól működik.

4. Hibakeresési Képességek:

- A hibakeresést segítő funkcionálisok bevezetése. Logolj fontos információkat, hogy a hibák könnyebben követhetők legyenek, és készíts olyan üzeneteket, amelyek segítenek a hibakeresésben.

5. Adatbiztonság:

- A felhasználók adatainak biztonságának fokozása. Gondold az adatok titkosítására, valamint az API-khoz való kapcsolódás során alkalmazott biztonsági intézkedésekre.

6. Funkciók Bővítése:

- További kényelmi funkciók bevezetése, például az aktuális árak és pozíciók automatikus frissítése egy időzített háttér folyamat segítségével.

7. Szélesebb API Támogatás:

- A jövőre való tekintettel fontold meg más, a pénzügyi adatokat szolgáltató API-k integrálását is, hogy a felhasználóknak szélesebb választékban legyen lehetőségük adatokat lekérni.

8. Felhasználói Beállítások:

- Személyreszabható beállítások bevezetése, hogy a felhasználók testre szabhassák az alkalmazást az egyéni igényeik szerint.

9. Dokumentáció és Segítségrendszer:

- Alapos dokumentáció készítése a felhasználók számára, és egy beépített segítségrendszer bevezetése az alkalmazás használatának megkönnyítése érdekében.

10. Rendszerkarbantartás:

- Rendszeres karbantartási folyamatok bevezetése, például az adatbázis tisztítása, a kód optimalizálása és a szoftverfrissítések rendszeres ellenőrzése.

Ezek a fejlesztési tippek segíthetnek az alkalmazásod további fejlesztésében és kibővítésében, hogy még hatékonyabb és felhasználóbarátabb legyen a jövőben.

Adatbázis műveletek

Adatbázis Rest1 Forex SoapKliens Adatbányászat Egyéb menü

Id	Név	Típus	Díjazott
1	Süni	vegyes	false
2	Gesztenyealagút	vegyes	false
3	Sajtos pogácsa	sós teasütemény	false
4	Diós-mákos	beigli	false
5	Sajttorta (málnás)	torta	false
6	Citrom	torta	false
7	Eszterházy	tortaszelet	false
8	Rákóczi-túrós	pite	false
9	Meggyes kocka	tejszínes sütemény	false
10	Legényfogó	torta	true
11	Alpesi karamell	tortaszelet	false
12	Kókuszcsók édes	teasütemény	false
13	Habos mákos	pite	false

Adatbázis műveletek

Adatbázis Rest1 Forex SoapKliens Adatbányászat Egyéb menü

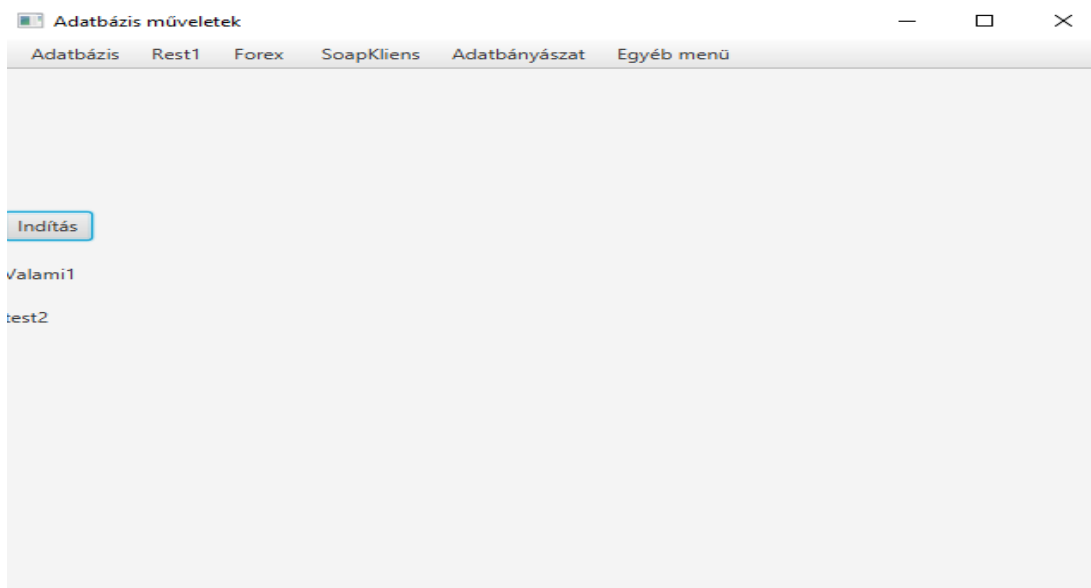
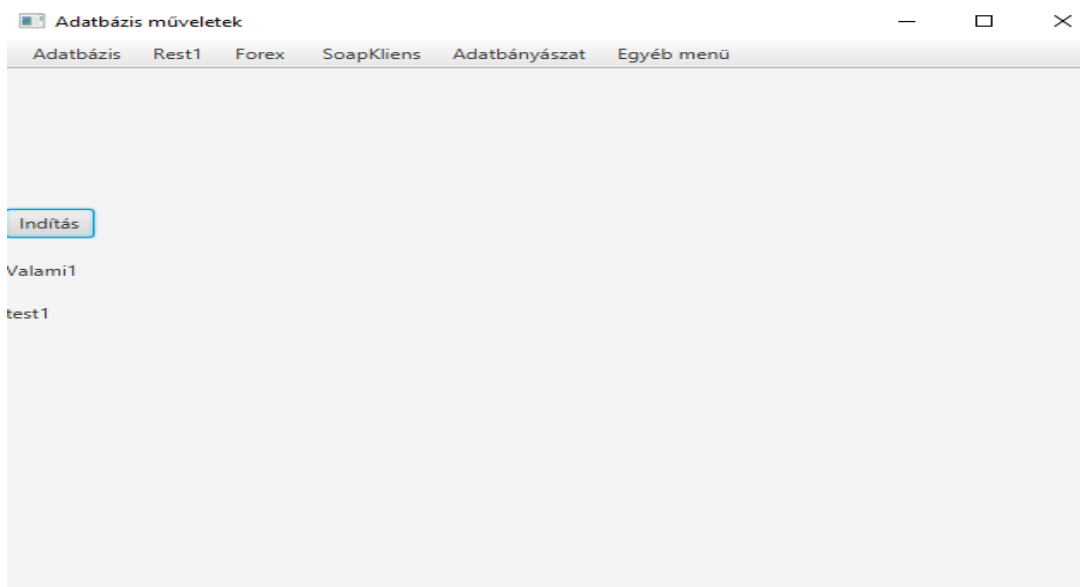
2

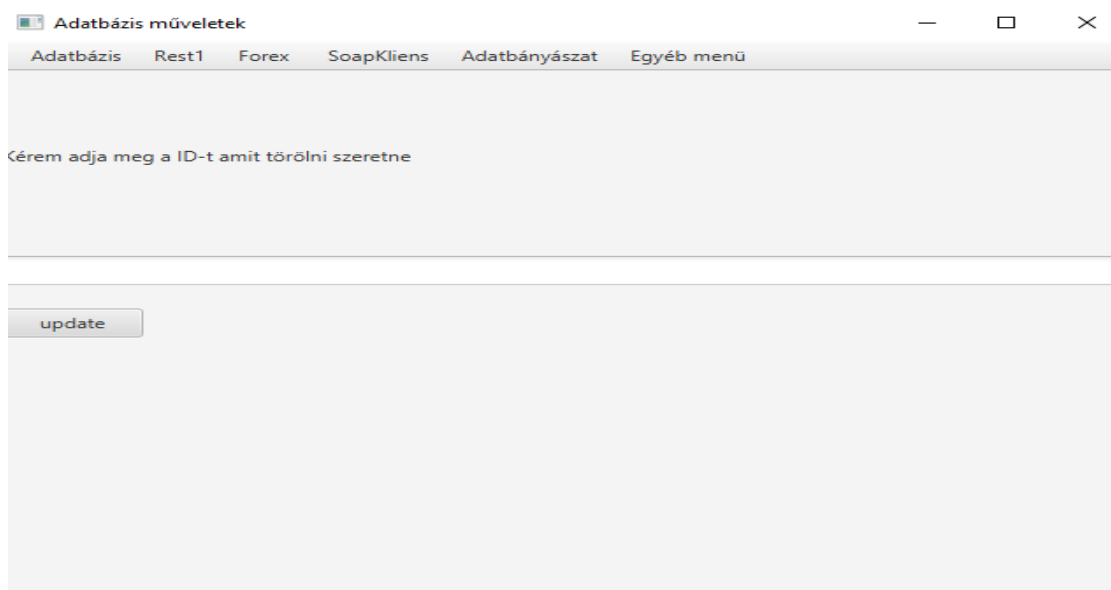
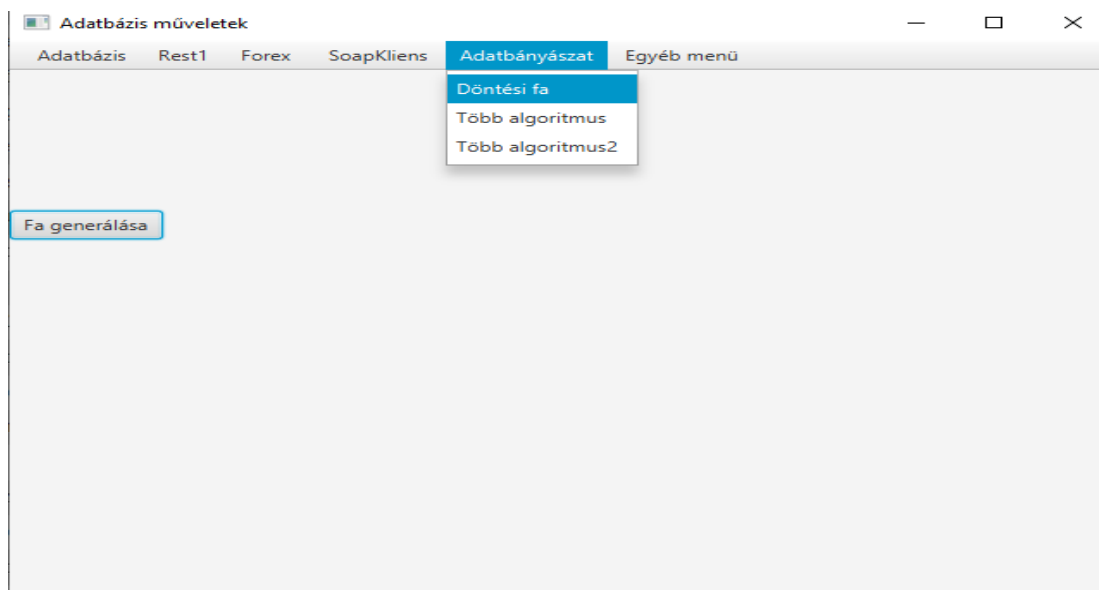
Generálás

Algorithm: SMO

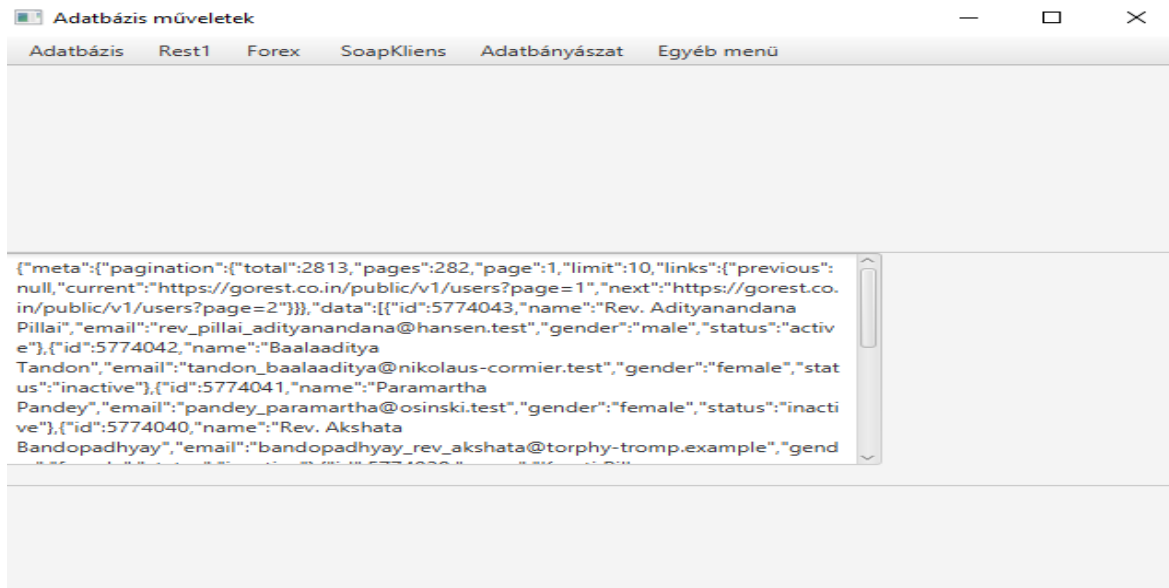
Results

Correctly Classified Instances	51	89.4737 %
Incorrectly Classified Instances	6	10.5263 %
Kappa statistic	0.7635	
Mean absolute error	0.1053	
Root mean squared error	0.3244	
Relative absolute error	23.0111 %	
Root relative squared error	67.9505 %	

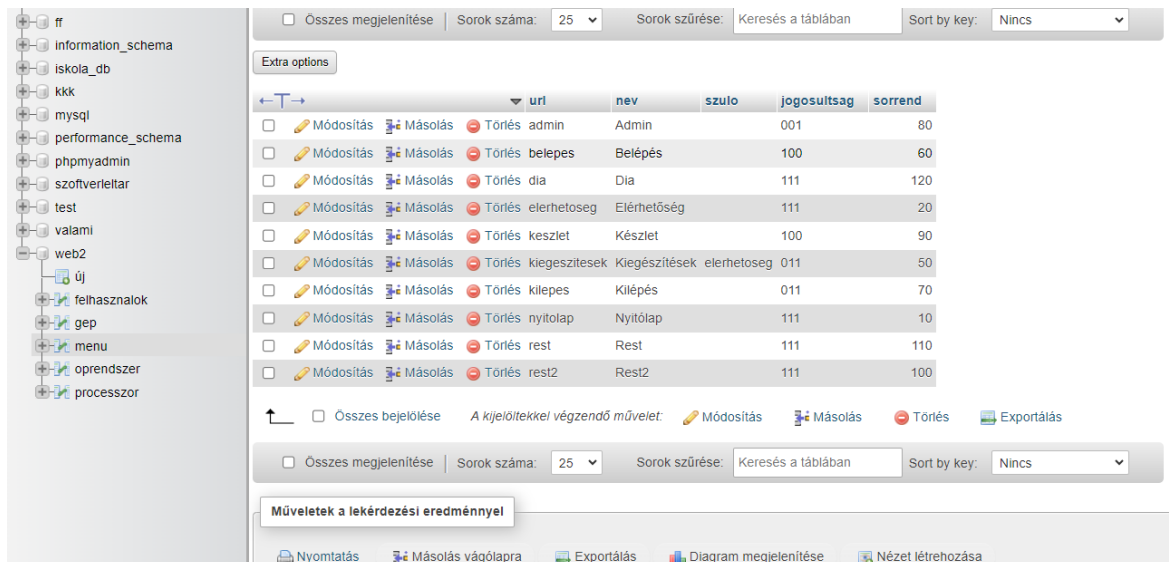














Diagramm



```
{
  "meta": {
    "pagination": {
      "total": 2813,
      "pages": 282,
      "page": 1,
      "limit": 10,
      "links": {
        "previous": null,
        "current": "https://gorest.co.in/public/v1/users?page=1",
        "next": "https://gorest.co.in/public/v1/users?page=2"
      }
    },
    "data": [
      {
        "id": 5774043,
        "name": "Rev. Adityanandana Pillai",
        "email": "rev_pillai_adityanandana@hansen.test",
        "gender": "male",
        "status": "active"
      },
      {
        "id": 5774042,
        "name": "Baalaaditya Tandon",
        "email": "tandon_baalaaditya@nikolaus-cormier.test",
        "gender": "female",
        "status": "inactive"
      },
      {
        "id": 5774041,
        "name": "Paramartha Pandey",
        "email": "pandey_paramartha@osinski.test",
        "gender": "female",
        "status": "inactive"
      },
      {
        "id": 5774040,
        "name": "Rev. Akshata Bandopadhyay",
        "email": "bandopadhyay_rev_akshata@torphy-tromp.example",
        "gender": "female",
        "status": "active"
      }
    ]
  }
}
```



	url	nev	szulo	jogosultsag	sorrend
<input type="checkbox"/> 	admin	Admin		001	80
<input type="checkbox"/> 	belepés	Belépés		100	60
<input type="checkbox"/> 	dia	Dia		111	120
<input type="checkbox"/> 	elérhetoseg	Elérhetőség		111	20
<input type="checkbox"/> 	keszlet	Készlet		100	90
<input type="checkbox"/> 	kiegeszitesek	Kiegészítések elérhetoseg		011	50
<input type="checkbox"/> 	kilepes	Kilépés		011	70
<input type="checkbox"/> 	nyitolap	Nyitólap		111	10
<input type="checkbox"/> 	rest	Rest		111	110
<input type="checkbox"/> 	rest2	Rest2		111	100

Megtörtént és lehetséges hibák a fejlesztés során:

1. API Változások:

- Az Oanda API frissítésekor vagy változtatásakor felmerülhet az a probléma, hogy az alkalmazásnak alkalmazkodnia kell az új változásokhoz. Ez követeli meg az API-dokumentáció rendszeres ellenőrzését és az alkalmazás gyors frissítését.

2. Adatbiztonsági Kockázatok:

- Az API-kon keresztüli adatátvitel során felmerülő biztonsági kockázatok. Kritikus fontosságú az adatok titkosítása és az esetleges biztonsági rések azonnali kezelése.

3. API-k Kiszolgálhatósága:

- Az Oanda API rendszerének elérhetősége ingadozhat, ami problémát okozhat az adatok letöltése és az alkalmazás normál működése szempontjából.

4. Grafikus Felület Stabilitása:

- A GUI tervezése és fejlesztése során felmerülhetnek problémák a különböző felbontások, operációs rendszerek és böngészők kompatibilitásával, amelyeket kezelni kell az egységes és stabil felhasználói élmény érdekében.

5. Szálkezelési Konfliktusok:

- A párhuzamos szálkezelés használata esetén felmerülhetnek szálkonfliktusok, amelyek hibákat és instabilitást okozhatnak. Szükség van megfelelő szinkronizációra és tesztelésre a szálak biztonságos kezelése érdekében.

6. API Túlterhelés:

- Gyakori használat esetén túlterhelhetjük az Oanda API-t, ami a szolgáltatás korlátozásához vagy megszakadásához vezethet. Ezt optimalizálással és az API-hívások megfelelő számának kezelésével kell kezelni.

7. Adatok Helyes Kezelése:

- Az Oanda API-től érkező adatokat helyesen kell értelmezni és kezelni. Hibás adatok vagy azok hiánya problémákat okozhatnak az alkalmazásban, például hibás döntések vagy üres tartalmak.

8. Frissítési és Letöltési Teljesítmény:

- Nagy adatmennyiségek esetén a letöltések és frissítések lassulhatnak. Különböző optimalizációs módszerek és aszinkron folyamatok segíthetnek ezen a területen.

9. Tesztelési Nehézségek:

- Az új funkciók és változtatások bevezetésekor a tesztelés kihívást jelenthet. Szükség van átfogó tesztervekre és a felhasználói élmény folyamatos monitorozására.

10. Dokumentáció Frissítése:

- Az alkalmazás bővítése esetén a dokumentáció frissítése is kulcsfontosságú. A felhasználóknak mindig naprakésznek kell lenniük a funkciókkal és az alkalmazás használatával kapcsolatos információkkal.

Ezek a problémák általában elkerülhetők vagy megoldhatók megfelelő tervezéssel, teszteléssel és karbantartással. A fejlesztés során a rugalmasság és a gyors reagálás segíthet az ilyen típusú kihívások kezelésében.

Mappastruktúra és generált fájlok:

Döntési fa.txt - Jegyzet

Fájl Szerkesztés Formátum Nézet Súgó

j48 pruned tree

```
-----
wage-increase-first-year <= 2.5: bad (14.28/1.28)
wage-increase-first-year > 2.5
|   statutory-holidays <= 10
|   |   vacation = below_average: bad (7.62/1.62)
|   |   vacation = average: bad (0.0)
|   |   vacation = generous: good (2.18)
|   statutory-holidays > 10: good (26.92/1.0)

Number of Leaves :    5
Size of the tree :    8
```

GepiTanulas.txt - Jegyzet

Fájl Szerkesztés Formátum Nézet Súgó

Algorithm: IBk

Results		
Correctly Classified Instances	52	91.2281 %
Incorrectly Classified Instances	5	8.7719 %
Kappa statistic	0.8138	
Mean absolute error	0.2722	
Root mean squared error	0.3281	
Relative absolute error	59.4998 %	
Root relative squared error	68.7093 %	
Total Number of Instances	57	

Correctly Classified Instances: 52 91.2280701754386%

Incorrectly Classified Instances: 5

Algorithm: J48

Results		
Correctly Classified Instances	42	73.6842 %
Incorrectly Classified Instances	15	26.3158 %
Kappa statistic	0.4415	
Mean absolute error	0.3192	
Root mean squared error	0.4669	
Relative absolute error	69.7715 %	
Root relative squared error	97.7888 %	
Total Number of Instances	57	

Correctly Classified Instances: 42 73.6842105263158%

Incorrectly Classified Instances: 15

Algorithm: SMO

Results		
Correctly Classified Instances	51	89.4737 %
Incorrectly Classified Instances	6	10.5263 %
Kappa statistic	0.7635	
Mean absolute error	0.1053	
Root mean squared error	0.3244	
Relative absolute error	23.0111 %	
Root relative squared error	67.9505 %	
Total Number of Instances	57	

Correctly Classified Instances: 51 89.47368421052632%

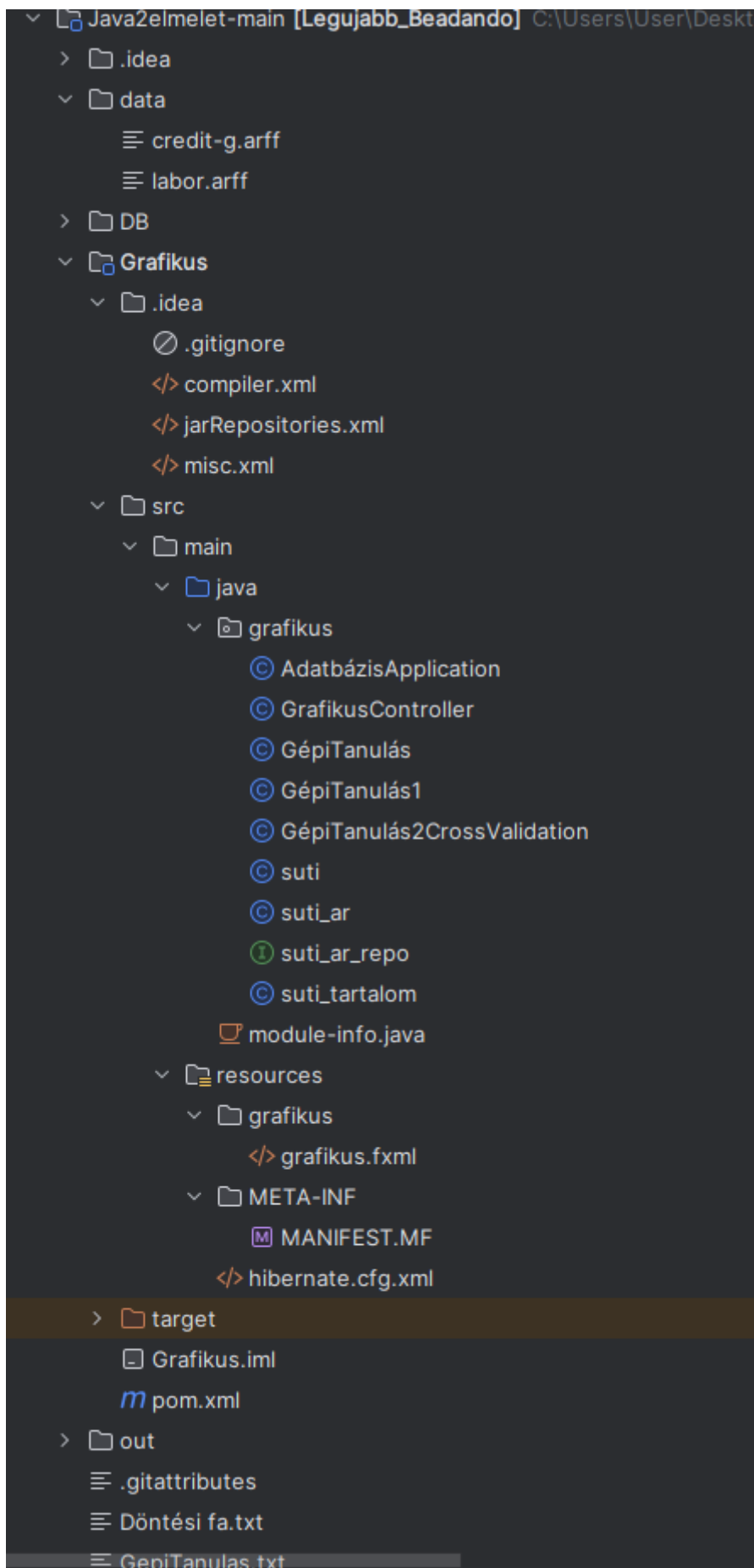
Incorrectly Classified Instances: 6

Algorithm: NaiveBayes

Results		
Correctly Classified Instances	51	89.4737 %
Incorrectly Classified Instances	6	10.5263 %
Kappa statistic	0.7741	
Mean absolute error	0.1042	
Root mean squared error	0.2637	
Relative absolute error	22.7763 %	

Sor: 1. oszl.: 1 100% Unix (LF) UTF-8

Sor: 1. oszl.: 1 100% Windows (CRLF) UTF-8



Tartalom

FEJLESZTŐI DOKUMENTÁCIÓ	3
ADATBÁZIS MENÜ	3
REST1 MENÜ	3
SOAPKLIENS MENÜ:	4
ADATBÁNYÁSZAT MENÜ:	4
EGYÉB MENÜ:	5
FOREX MENÜ	5
TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK A JÖVŐBEN:.....	7
DIAGRAMM	11
MEGTÖRTÉNT ÉS LEHETSÉGES HIBÁK A FEJLESZTÉS SORÁN:	12