

**Bevezetés**

Napjainkban sok helyről hozzáférhetünk a friss hírekhez, rengeteg híroldal működik, és mindegyik arra törekszik, hogy a lehető leggyorsabban értesítse az olvasóit a világban történő eseményekről. Ezeken az oldalakon általában szűrhetjük a híreket téma szerint (politika, sport, közélet stb.), de az olyan híroldal ritka, amely az érzelmi töltet vagy hangulat szerint csoportosítja a tartalmait.

Az alkalmazás célja éppen ezért az, hogy az olvasó hangulat szerint osztályozva lássa a cikkeket, és könnyen választhasson a kategóriák között. Minden cikk rendelkezik egy címkével, amely az adott cikk hangulatát, érzelmi töltetét jelöli, illetve kiválasztható, hogy csak egy bizonyos csoportba tartozó cikkeket jelenítsen meg.

A PletykAI egy olyan webes alkalmazás, amely egy híroldal (www.maszol.ro) főoldalát fésüli át (webscraping), onnan összegyűjti a cikkeket, majd az adatok elemzése és feldolgozása után azokat osztályozva megjeleníti.

Adatok elemzése alatt a következő műveleteket kell érteni:

* releváns HTML tagek tartalmának kinyerése az oldal forráskódjából
* cikkek címének lefordítása angol nyelvre
* a cikkek felcímkézése egy mesterséges intelligencia által az szerint, hogy pozitív, negatív vagy semleges hangvételű

**Irodalmi áttekintés**

A hangulatelemzés (más néven véleménybányászat vagy érzelmi mesterséges intelligencia) természetes nyelvi feldolgozás, szövegelemzés, számítógépes nyelvészet és biometrikus adatok felhasználása az érzelmi állapotok és szubjektív információk szisztematikus azonosítására, kinyerésére, számszerűsítésére és tanulmányozására. A hangulatelemzést széles körben alkalmazzák a vásárlói anyagok, például vélemények és felmérésekre adott válaszok, online és közösségi média, valamint egészségügyi anyagok értelmezésére, amelyek a marketingtől az ügyfélszolgálaton át a klinikai orvoslásig hasznosak lehetnek. A mélynyelvi modellek, például a RoBERTa térnyerésével nehezebb adattartományok is elemezhetők, például olyan hírszövegek, ahol a szerzők jellemzően kevésbé egyértelműen fejezik ki véleményüket.

**Felhasznált technológiák**

Flask

Az alkalmazás alapját egy Flask webalkalmazás képezi, amely egy Pythonra épülő webes keretrendszer. Mikrokeretrendszerként jellemzik, mert igényel különleges eszközöket vegy könyvtárakat. Nincs beépített adatbáziskezelője, sem űrlap validációja, ezért viszonylag egyszerű a megtanulása és gyorsan lehet használható eredményt elérni vele, ugyanakkor ezek a tulajdonságai a negatívumai is lehetnek egyben. Elérhető hozzá sok féle kiegészítő csomag, amelyekkel bővíthetők a funkcionalitások.

Jinja

Ez egy web template engine, amit a Flask használ. Segítségével Python utasításokat adhatunk meg a HTML állományokban.

BeautifulSoup

A BeautifulSoup egy Python-csomag HTML- és XML-dokumentumok elemzésére. Létrehoz egy elemzőfát az elemzett oldalak számára, amely felhasználható az adatok HTML-ből való kinyerésére (webscraping). Segítségével kibányászható a HTML tagek tartalma a weboldalak forráskódjából.

googletrans

A Googletrans egy ingyenes Python csomag, amely a Google Fordító API-ját implementálja pythonba. Segítségével a kódon belül használhatjuk a google fordító minden funkcióját.

VADER (Valence Aware Dictionary and sEntiment Reasoner)

Egy lexikon- és szabályalapú érzéselemző eszköz, amely kifejezetten a közösségi médiában kifejezett érzelmekre van hangolva. Nyílt forráskódú az MIT-licenc alatt.

-1 és 1 közötti skálán pontozza a mondatokat, a 0 a semlegest jelenti.

A VADER hangulatlexikon érzékeny mind a közösségi média kontextusában kifejezett érzelmek polaritására, mind intenzitására, és általánosan alkalmazható más területeken végzett érzelmek elemzésére is.

szöveges hangulatelemzésre használt modell, amely érzékeny mind az érzelmek polaritására (pozitív/negatív), mind intenzitására (erősségére). A 2014-ben bevezetett VADER szöveges hangulatelemzés emberközpontú megközelítést alkalmaz, ötvözve a kvalitatív elemzést és az empirikus validálást emberi értékelők és a közösség segítségével.

A lexikális megközelítések célja, hogy lexikont vagy „érzelmi szótárat” építsenek a szavak érzelmekre. Ezt a szótárt használhatjuk kifejezések és mondatok hangulatának felmérésére anélkül, hogy bármi másra is kellene figyelni. A hangulat lehet kategorikus – például {negatív, semleges, pozitív} – vagy lehet numerikus is – például intenzitás- vagy pontszámtartomány. A lexikális megközelítések megvizsgálják a mondat minden egyes szavának érzelmi kategóriáját vagy pontszámát, és eldöntik, hogy mi az egész mondat érzelmi kategóriája vagy pontszáma. A lexikális megközelítések ereje abban rejlik, hogy nem kell címkézett adatok felhasználásával modellt betanítanunk, hiszen mindenünk megvan ahhoz, hogy az érzelmek szótárában felmérjük a mondatok hangulatát. A VADER egy példa a lexikális módszerre.

Egy szöveg hangulatpontszámát a szöveg egyes szavainak intenzitásának összegzésével kaphatjuk meg.

A VADER hangulatelemzés (jó, a Python implementációban egyébként) -1 és 1 közötti hangulatpontszámot ad vissza, a legnegatívabbtól a legpozitívabbig.

Egy mondat hangulatpontszámát a VADER-szótárban felsorolt ​​egyes szavak hangulatpontszámainak összegzésével számítják ki a mondatban. Az óvatos olvasók valószínűleg észreveszik, hogy van ellentmondás: az egyes szavak hangulatpontszáma -4 és 4 között van, de egy mondat visszaadott hangulatpontszáma -1 és 1 között van.

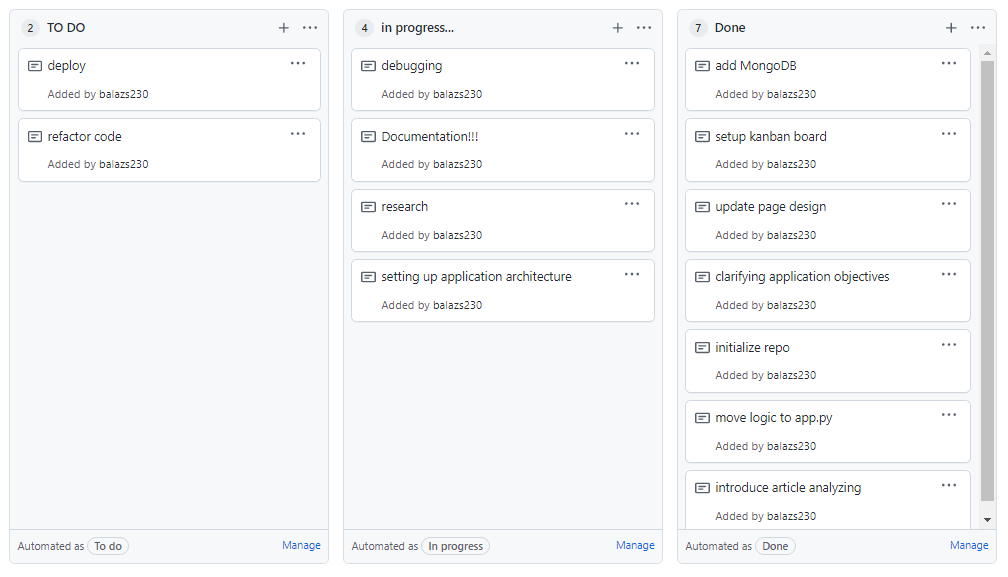
MongoDB

A MongoDB [nyílt forráskódú](https://hu.wikipedia.org/wiki/Ny%C3%ADlt_forr%C3%A1sk%C3%B3d) dokumentumorientált [adatbázis](https://hu.wikipedia.org/wiki/Adatb%C3%A1zis) szoftver, a [NoSQL](https://hu.wikipedia.org/wiki/NoSQL" \o "NoSQL) adatbázisszerverek közé tartozik. A dokumentumokat [JSON](https://hu.wikipedia.org/wiki/JSON)-szerű formátumban tárolja (BSON).

A projektben lokális MongoDB klaszter van alkalmazva.

Projektmenedzsment

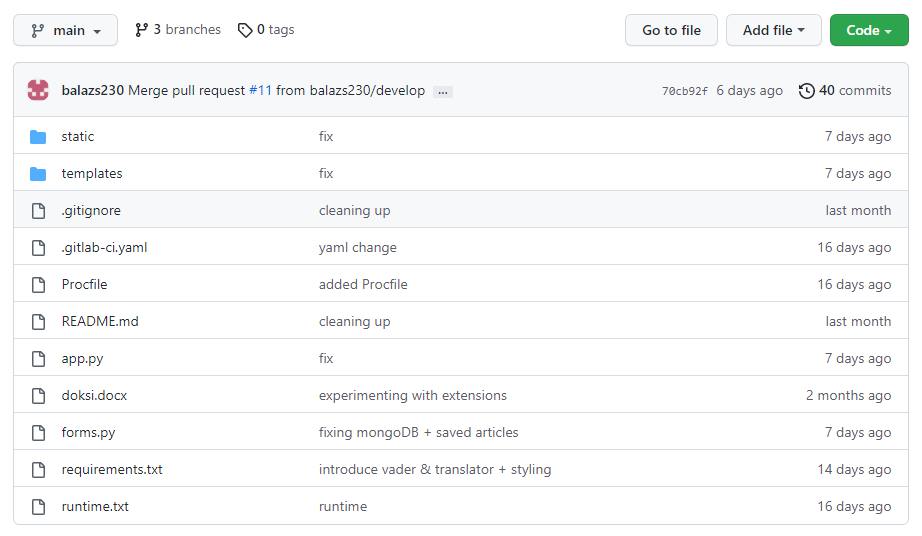
A projekt menedzseléséhez egy Kanban boardot használtam Github-on. Itt láthatók az elvégzett feladatok, implementált fukciók, a még folyamatban lévő feladatok, illetve néhány lehetőség a program továbbfejlesztésére.



Verziókezelés

A project egy nyilvános GitHub repositoryban található. Itt két branchet használtam (developer és main). Minél gyakoribb kommitokkal dolgozatm, annak érdekében, hogy részletesen követhető és visszaállítható legyen.

A fájlrendszer a Flask struktúráját követi.



Rendszerkövetelmények

A program helyi futtatásához arra van szükség, hogy legyen telepítve a Python 3.10+, telepítsük a requirements.txt fájlban levő csomagokat, illetve legyen konfigurélva egy helyi MongoDB adatbázis.

Internetelérés szükséges az alkalmazás használhatához.

Futtatás:

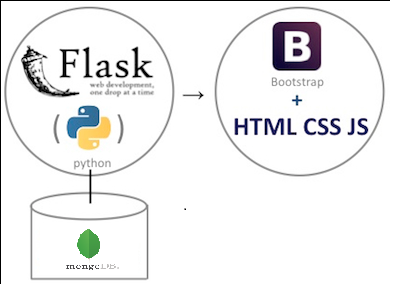
python app.py – a fő állomány futtatása

mongod – az adatbázis elindítása

Funkcionális követelmények

Nem-funkcionális követelmények

Architektúra



Use case diagrammok