Big data

Microsoft Azure

2016.

# Bevezetés

## Célkitűzés

A labor során egy egyszerű gépi tanulás alkalmazást fogunk elkészíteni az Azure Machine Learning (ML) ingyenes verziójával. Az alkalmazásban az ML Studio példaadatival fogunk dolgozni – egy gépjárművek adatait és árait tartalmazó adatforrás alapján alkalmazásunk meg fogja tudni becsülni tetszőleges, általunk megálmodott paraméterekkel rendelkező autómodelleket várható árát.

A labor másik felében pedig megismerkedünk a Cognitive Services nyújtotta lehetőségekkel. Kipróbáljuk az API használatát egy nem és életkor felismerő program készítése segítségével.

## Előfeltételek

* Azure előfizetés
* Visual Studio 2015

## Amit érdemes átnézned

* Kapcsolódó előadásanyag

## Oktatói útmutatás

A mesterséges intelligencia alapú megoldások nélkülözhetetlen része a hatalmas tanuló minta és az ezek feldolgozására képes infrastruktúra. Bár a konkrét labor feladatok egy-egy megoldás végigviteléről szólnak, érdemes a hallgatók figyelmét felhívni a releváns részeknél az alternatív felhasználási lehetőségekre, illetve a mögöttes architektúra komplexitására.

# Machine Learning

## Machine Learning Studio

A Machine Learning Studio-ban baloldalon vannak a rendelkezésre álló modulok, középen van a vászon, ahová drag-and-droppal tudunk behúzni modulokat, jobb oldalon pedig a vásznon aktuálisan kijelölt modult tudjuk konfigurálni.

Baloldalon a modulok között felül tudunk keresni.

A legtöbb modulnak vannak kimeneti és bemeneti portjai, ezeket a vászonra behúzott modulon kis karikák reprezentálnak felül (input) és alul (output). Ezeket össze lehet kötni egymással (így áll elő a megoldás). A kimeneti portokra kattintva érhető el a Visualize opció, ahol a modulból kijövő adatokat tudjuk megjeleníteni.

A modellt futtatni a lenti fekete sávon lévő Run gombbal lehet.

## A Modell betanítása

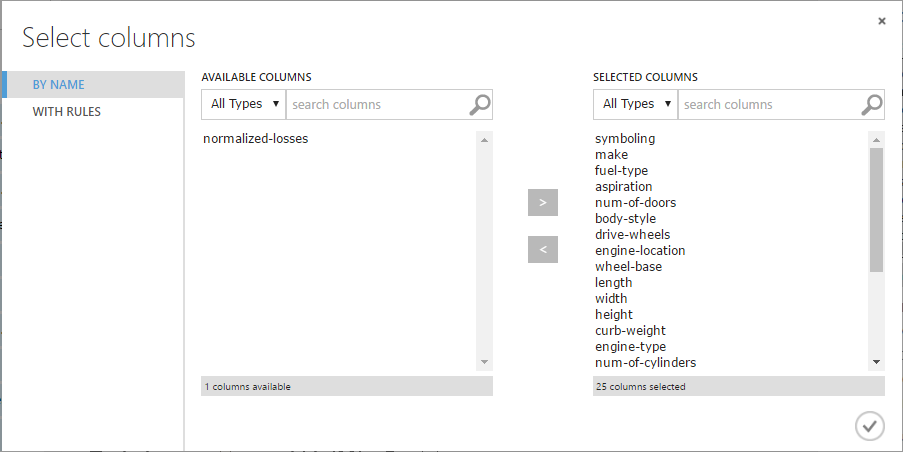
Lépjünk be Microsoft fiókunkkal az <https://studio.azureml.net/> címen!

Első körben be kell tanítanunk a modellt. Válasszuk a lenti New gombra kattintva az Experiment lehetőséget, majd a Blank experiment opciót!

Írjuk át az Experiment nevét valami beszédesebbre („Experiment created on….” az eredeti név”).

Válasszuk ki az adatforrást! A bal oldali keresőben keressük meg az Automodel price data (Raw) modult és húzzuk be a jobb oldali vászonra! A felugró ablakban statisztikákat is láthatunk, ha az egyes oszlopokra kattintunk, illetve meg tudjuk vizsgálni a teljes adathalmazt.

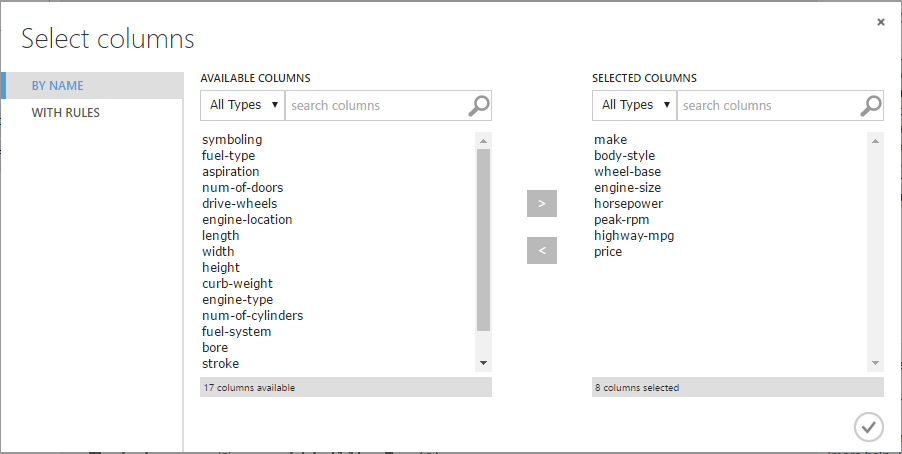
Húzzunk be egy Select Columns in Dataset modult is a vászonra, majd az adatforrást kössük össze ezzel a modullal! Jelöljük ki a modult, majd a modul beállításai között válasszuk a Launch column selector opciót! Első körben kitisztítjuk az adatforrást, hogy a normalized-losses mező ne legyen benne, erre nincs szükségünk (minden oszlopot vigyünk át a jobb oldalra, és a normalized-lossest vigyük vissza a bal oldalra.):



Következő lépés, hogy a hiányzó adatokat megfelelően kezelni tudjuk. Ehhez húzzunk a vászonra egy Clean Missing Data modult, kössük össze a Select Columns in Dataset kimenetét és ennek a bemenetét. A modul beállításai között, a „Cleaning Mode” legördülő listában válasszuk a Remove enitre row opciót! Ezzel ha egy sor bármelyik adata hiányzik, a teljes adatsor kivesszük.

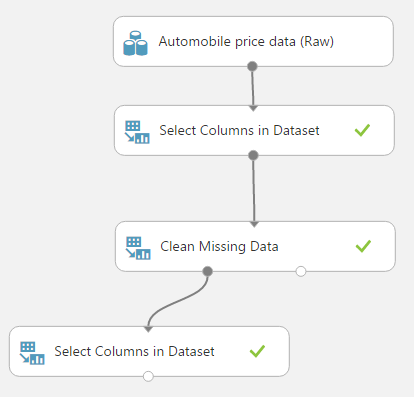
Futtassuk a megoldást! Ezután a modul kimenetét vizualizálva láthatjuk, hogy most már kevesebb sorunk van, mint az adatforrásban volt.

Következő lépés a feautre engineering. Húzzunk egy újabb Select Columns in Dataset modult a vászonra, kössük össze a Clean Missing Data kimenetét az új Select Columns in Dataset bemenetével. A Launch column selector beállításnál válasszuk az alábbiakat:



Itt most belevesszük az autónak az ár szempontjából fontos tulajdonságait (feature-ök), illetve azt a mezőt, ami az optimalizálásunk célja (price).

Ezen a ponton a modellünknek így kell kinéznie:



Futtassuk újra a modellt (ezzel előállnak a szűrt adatok).

Következő lépésben húzzunk be egy Split Data modult a vászonra és a bemenetét kössük össze a Select Columns in Dataset kimenetével. Ezzel a modullal két részre bontjuk az adatainkat: az egyik részével a modell tanítását fogjuk végezni, a másik részével pedig pontozni fogjuk a modellt. A beállításai között a „Fraction of rows in the first output dataset” opciónál írjunk 0.75-öt – az adataink 75%-át fogjuk használni a tanításhoz, a maradékot a score-inghoz.

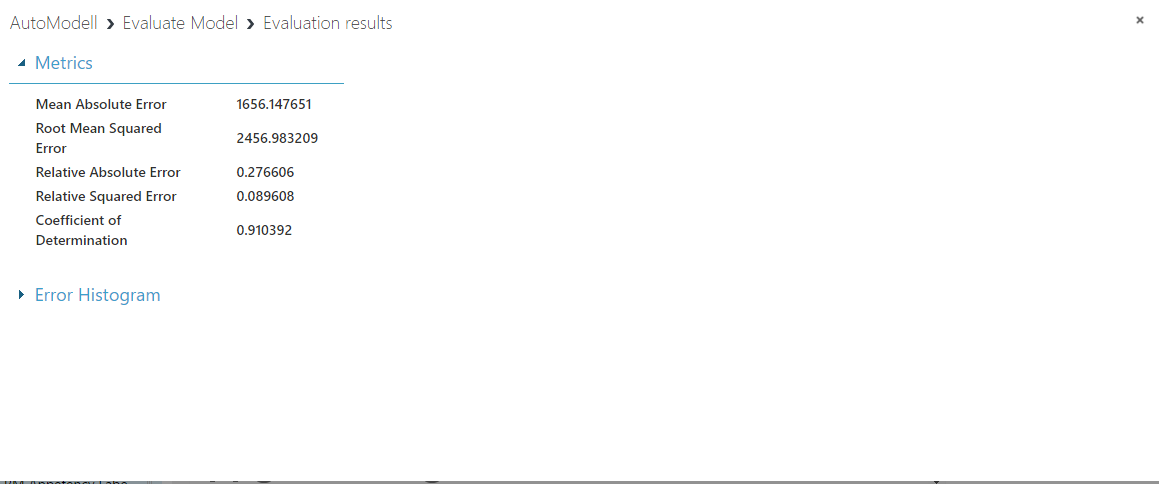
Futassuk a megoldást az adatok szétválasztásához!

Húzzunk be egy Linear Regression modult a vászonra! Ez a modul lesz a megoldásunk lelke, ez az egyik a rendelkezésünkre álló ML algoritmusok közül (egyszerű lineáris regressziót valósít meg).

Vegyünk fel a vászonra egy Train Model modult! Ez fogja a modell tanítását elvégezni. A linear regression modul kimenetét kössük be ennek a baloldali, a Split baloldali kimenetét ennek a jobboldali inputjába! Futassuk a megoldást, ezzel megtörténik a modell betanítása. A Train Model modulnak a Price oszlopot adjuk meg!

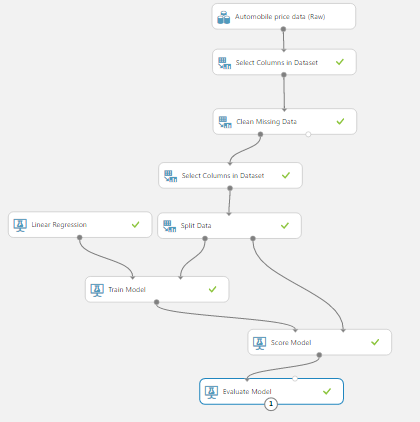
Húzzunk be egy Score Model modult is a vászonra, majd a Train model kimenetét kössük ennek baloldali bemenetére, a Split modul jobboldali kimenetét pedig ennek a jobboldali bemenetére! A betanított modellünk által felismert szabályszerűségek alapján ez a modul a Split által kiválaszott 25%-nyi adatnak fogja majd az árát megbecsülni. Futassuk a modellt, majd vizualizáljuk a Score model kimenetét! Az utolsó oszlopban látszik a becsült ár, előtte pedig a valódi ár!

Ahhoz, hogy a modellt egyszerűbben tudjuk értékelni, húzzunk be egy Evaluate Model modult, a Score Model kimenetét kössük be ennek baloldali inputjába. Ezután futtassuk a megoldást, majd vizualizáljuk a kimenetét az Evaluate Modelnek:



Ezek mind statisztikai mutatók, a legfontosabb az utolsó, ami azt mutatja meg, mennyire jól illeszkedik a modellünk az adatokra (0 és 1 közötti érték, minél nagyobb, annál jobb).

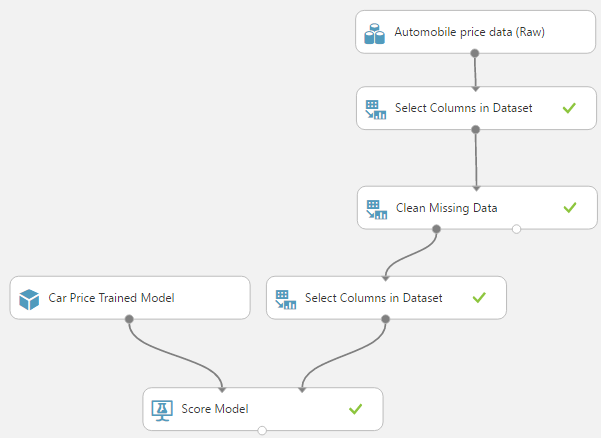
A teljes, elkészült megoldás:



## Publikálás webszolgáltatásként

Az elkészült megoldásunkat tudjuk webszolgáltatásként egy REST API-n publikálni. Elsőként kattintsunk jobb gombbal a Train Model modulon, majd válasszuk a Save as Trained Model opciót! Ezzel el tudjuk menteni a betanított modellünket egy adott néven (mondjuk Car price trained model).

A lenti fekete sávon válasszuk a Save as opciót, ezzel elkészítjük a modellnek egy másolatát. Ezt a másolatot szerkesszük úgy, hogy a Split, Linear Regression, Train Model és Evaluate Model modulokat töröljük. A Score Model jobb oldalára most kössük be a Project Columns kimenetét. Húzzuk be a vászonra a korábban elmentett trained modelt, és ennek kimenetét kössük be a Score model másik bemenetére:



Ezután nyomjuk meg a „Set up a web service” menüpontot! Alapból a Web service output jó helyen van (a Score Model kimenetén), a Web Service Inputot pedig kössük a Score Model jobb oldali bemenetére! Így a pontozás most már nem a bemeneti adatokra fog futni, hanem arra, amit a felhasználó megad (a Project Columns kimenetét továbbra is összekötve kell hagyni a bemenettel, mert az adatok sémáját innen ismeri fel a szolgáltatás), a kimenetben pedig a korábban látott vizualizáció fog előállni.

Futtassuk a modellt, majd válasszuk a Deploy Web Service opciót!

Deploy után ki tudjuk próbálni a webszolgáltatás Request/Reponse üzemmódját (van egy batch üzemmód is) a Test linkre kattintva! Itt kell kitölteni az adatokat, majd a lent megjelenik az eredmény, ha készen van és látjuk az árat.

Az API help page linkre kattintva a weboldal alján találunk egy C# kódot is, amivel meghívható a szolgáltatás, ha van időnk, akkor ezt is kipróbálhatjuk (ne felejtsük el az API kulcsot az előző oldalról bemásolni a kódba).

# Microsoft Cognitive Services

A feladat során a Microsoft Cognitive Services arcfelismerő algoritmusát fogjuk egy C# alapú konzolos alkalmazásból meghívni. (A szolgáltatást amúgy közvetlenül, böngészőből is kipróbálhatjuk, ha időnk engedi)

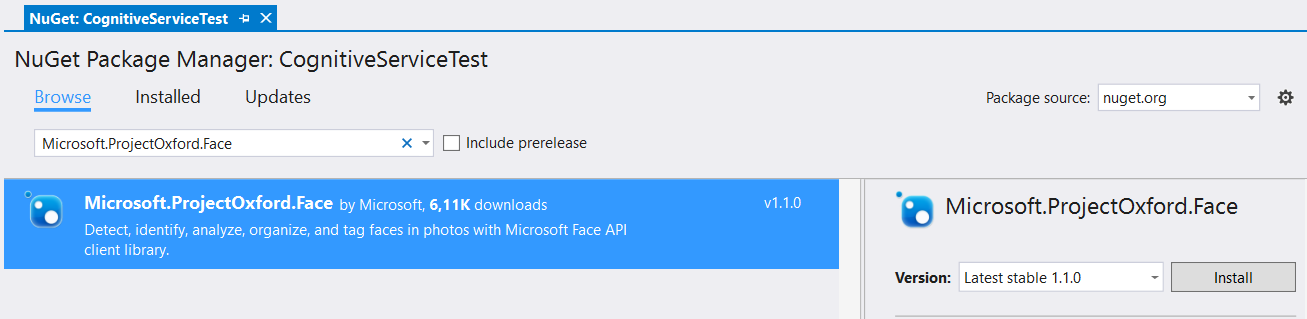
## Környezet előkészítése

Hozzunk létre egy új konzol alkalmazást a Visual Studioban!

Visual Studio Solution Explorer-ben a Solution-re jobb kattintás🡪 Add🡪New Project 🡪Visual C# 🡪Console Application. A projekt neve legyen „CoginitiveServiceTest”

Azonnal állítsuk is be, hogy a Start gomb megnyomására ez a projekt induljon el. Ehhez kattintsunk jobb egérgombbal a projekten 🡪 Set as Startup project.

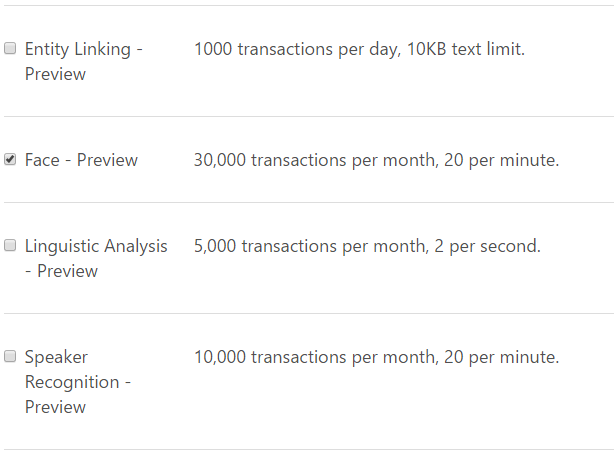
Ezután adjunk hozzá egy új NuGet csomagot a projekthez. Ehhez a Solution Explorer-be kattintsunk Jobb gombbal a létrehozott projektre, majd a felugró menübe válasszuk ki a Manage NuGet Packages opciót, és keressük meg a megfelelő csomagot a képnek megfelelően.



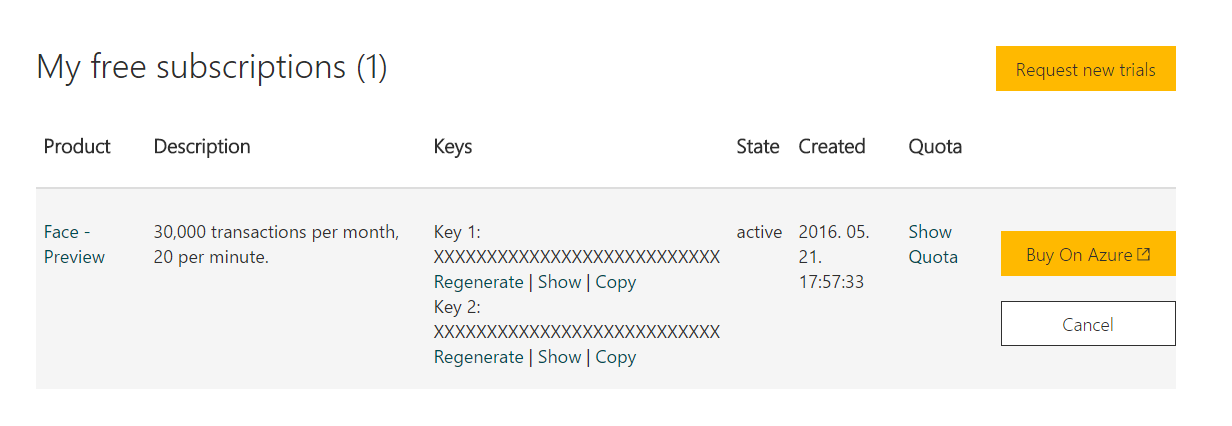
A Microsoft.ProjectOxford.Face NuGet csomag tartalmazza az API-hoz tartozó SDK-t.

Természetesen az API egyszerű HTTP kérésekkel is elérhető lenne, azonban a munkánk során az egyszerűség kedvéért az SDK-t fogjuk használni.

Ezután egy API kulcsot kell igényelnünk a Cognitive Services API használhatához. Ehhez látogassunk el a <https://www.microsoft.com/cognitive-services> weboldalra, majd a jobb felső sarokba található My Account gombra kattintva jelentkezzünk be. A bejelentkezés után megjelenő oldalon kiválaszthatjuk melyik API-kat szeretnénk használni. A gyakorlat során a Face API-t fogjuk kipróbálni így ehhez kérjünk is kulcsot.



Miután az oldal alján elfogadtuk a feltételeket kattintsunk a Subscribe gombra. A megjelenő oldalon láthatjuk az előfizetéseinket, és elérhetjük az API kulcsokat is.



Kattintsunk a Key 1 alatt található show gombra hogy megjelenjen az egyik API kulcsunk, mert később szükségünk lesz rá. Bármikor visszatérhetünk erre az oldalra a My Account gombra kattintva.

## Kliens elkészítése

Miután előkészítettük a környezetet, térjünk vissza a Visual Studio projekthez. A konzolalkalmazásunk úgy fog működni, hogy indítás után bekér egy URL címet, ami egy olyan képre mutat, amin egy vagy több arc található, majd kiírja a képen szereplő személyek életkorát és nemét az arcfelismerő API segítségével.

Részletes API leírás a hivatalos dokumentációban: <https://dev.projectoxford.ai/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395236>

Első lépésként készítsük el a metódust ami elvégzi a hívást és feldolgozza a kapott választ.

static async Task MakeRequest(string imageUrl)

{

    var client = new FaceServiceClient("{api kulcs ide}");

    var attributes = new List<FaceAttributeType> {FaceAttributeType.Age, FaceAttribute

Type.Gender};

    var res = await client.DetectAsync(imageUrl, false, true, attributes);

    foreach (var face in res)

    {

        Console.WriteLine($"Kor: {face.FaceAttributes.Age} Nem: {face.FaceAttributes.Gender}");

    }

}

Hívjuk fel a figyelmet hogy az egyszerűség kedvéért a kód kizárólag a legfontosabb részeket tartalmazza!

A metódusban létrehozunk egy kliens-t ami konstruktor paraméterbe várja az API kulcsunkat. Majd létrehozunk egy listát, amelyben megadjuk, hogy milyen adatokat szeretnénk visszakapni a képen szereplő arcokról. Ezután meghívjuk az API-t, majd miután megkaptuk a választ kiírjuk minden archoz az életkort és a nemet.

**Használat előtt ne felejtsük el kitölteni az api kulcsot.**

Figyeljünk rá, hogy ingyenesen percenként csak 20 kérést engedélyez, ezt nagy valószínűséggel nem lépjük túl a munka során de fontos észben tartani.

Utolsó lépésként hívjuk meg a metódust a megfelelő paraméterrel. Ehhez töltsük ki a Main függvényt az alábbi módon:

static void Main(string[] args)

{

    Console.WriteLine("Kérem adja meg a kép URL-jét:");

    MakeRequest(Console.ReadLine()).Wait();

    Console.WriteLine("A kilépéshez nyomjon meg egy gombot.");

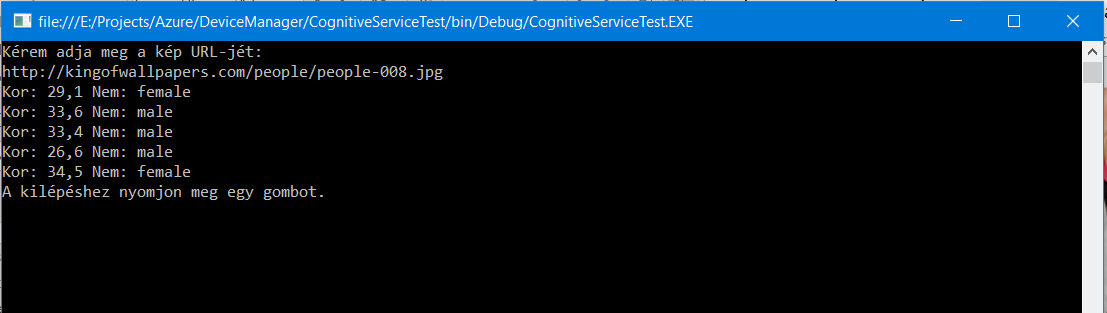
    Console.ReadKey();

}

Miután ezzel kész vagyunk, keressünk egy képet az interneten ami lehetőleg több arcot is tartalmaz, majd futtassuk az alkalmazást és adjuk meg a képre mutató linket.



Az alkalmazást lefuttatva a fent látható képre, az alábbi eredményt kapjuk:



Látható, hogy a szolgáltatás mind az öt képen szereplő személyt felismerte, és visszaadta a nemüket és az életkorukat.

Összefoglalásként elmondhatjuk, hogy a Cognitive Services API használatával viszonylag kevés munkával tudunk mesterséges intelligenciát építeni az alkalmazásunkba anélkül, hogy különleges ismeretekkel kéne rendelkeznünk a szolgáltatás mögött működő összetett rendszerekről. Az API-k biztosítanak ingyenes kipróbálási lehetőséget, részletes dokumentáció illetve SDK is elérhető hozzájuk, ami tovább egyszerűsíti a kipróbálást az érdeklődők számára.

# További információk

* Dokumentáció a Machine Learning-hez:

<https://azure.microsoft.com/en-us/documentation/services/machine-learning/>

* Dokumentáció és példa a Cognitive Services használatához: <https://www.microsoft.com/cognitive-services/>