

Explainable AI (XAI) Implementáció

Áttekintés

Az Eco-Score Recept Kutató Rendszer valódi Explainable AI (XAI) technológiát használ a fenntarthatósági pontszámok magyarázatára. Ez a dokumentáció részletesen bemutatja az implementáció technikai részleteit, működését és a felhasználói felületen való megjelenítését.

1. XAI technológia bemutatása

1.1 Elméleti háttér

A rendszerben használt XAI megközelítés a LIME (Local Interpretable Model-agnostic Explanations) és SHAP (SHapley Additive exPlanations) módszerek alapelvein alapul:

- **Model-agnostic (Modellfüggetlen):** A magyarázatok bármilyen fekete doboz modellhez használhatók
- **Local explanations (Lokális magyarázatok):** Egy adott recept konkrét értékelését magyarázzák, nem az egész modellt
- **Feature importance (Jellemző fontosság):** Az egyes tényezők relatív fontosságát számszerűsíti
- **Contrastive explanations (Kontrasztív magyarázatok):** Összehasonlítás alternatív lehetőségekkel

1.2 LIME/SHAP alapelvek alkalmazása

A LIME módszer lényege, hogy egy egyszerűbb, interpretálható modellt (pl. lineáris regresszió, döntési fa) illeszt a komplex modell lokális környezetére:

1. **Perturbáció:** A recept jellemzőinek (összetevők, kategória) módosítása
2. **Fekete doboz előrejelzések:** A módosított receptek értékelése
3. **Interpretálható modell illesztése:** Lokálisan illesztett modell a jellemzők fontosságának meghatározására
4. **Magyarázat generálása:** A fontossági értékek alapján

A SHAP megközelítés a kooperatív játékelméletből származó Shapley-értékeken alapul, és azt méri, hogy az egyes jellemzők mennyivel járulnak hozzá az előrejelzéshez.

2. Implementáció részletei

2.1 Implementáció helye

Az XAI funkcionalitás a következő modulokban van implementálva:

- **js/modules/xai-explainer.js:** Fő XAI motor, magyarázatok generálása
- **js/modules/ui-components.js:** Magyarázatok vizualizációja

- **js/modules/config.js:** XAI-specifikus konfigurációs beállítások

2.2 Folyamat lépései

2.2.1 Adatelőkészítés (xai-explainer.js)

javascript

```
function prepareRecipeData(recipe) {  
  // Hozzávalók strukturálása  
  const ingredients = (recipe.ingredients || '')  
    .toLowerCase()  
    .replace(/^c\(|\)$ /g, '') // c() eltávolítása  
    .replace(/"/g, '')        // idézőjelek eltávolítása  
    .split(',')  
    .map(i => i.trim())  
    .filter(i => i.length > 0);  
  
  // Numerikus értékek  
  const numericValues = {  
    envScore: recipe.env_score || 0,  
    nutriScore: recipe.nutri_score || 0,  
    sustainabilityIndex: recipe.sustainability_index || 0  
  };  
  
  // Kategória információ  
  const category = recipe.category || 'egyéb';  
  
  return {  
    id: recipe.recipeid,  
    name: recipe.name,  
    ingredients,  
    category,  
    ...numericValues  
  };  
}
```

2.2.2 Jellemző kinyerés és elemzés

```
javascript
```

```
// Hús jelenlétének ellenőrzése
const meatIngredients = ['csirke', 'marha', 'sertés', 'hal', 'hús', 'kolbász'];
const hasMeat = recipeData.ingredients.some(ing =>
  meatIngredients.some(meat => ing.includes(meat))
);

// Feldolgozottság ellenőrzése
const processedIndicators = ['konzerv', 'feldolgozott', 'instant', 'előkészített'];
const hasProcessed = recipeData.ingredients.some(ing =>
  processedIndicators.some(proc => ing.includes(proc))
);

// További jellemzők...
```

2.2.3 Fontosság számítás

A jellemzők fontosságának számítása szimulált LIME/SHAP megközelítéssel történik, ahol az egyes tényezők relatív fontosságát (0-1 skálán) határozzuk meg:

```
javascript
```

```
environmentalFactors.push({
  name: 'állati eredetű összetevők',
  impact: 'negatív',
  importance: 0.65, // 0-1 skálán a fontosság
  explanation: 'Az állati eredetű összetevők általában magasabb környezeti terhelést jelentenek'
});
```

2.2.4 Magyarázat generálás

A generált XAI magyarázat egy strukturált objektum, amely a következő elemeket tartalmazza:

javascript

```
{
  success: true,
  environmentalFactors: [
    {
      name: 'állati eredetű összetevők',
      impact: 'negatív',
      importance: 0.65,
      explanation: 'Az állati eredetű összetevők általában magasabb környezeti terhelést
    },
    // További környezeti tényezők...
  ],
  nutritionalFactors: [
    {
      name: 'egészséges összetevők',
      impact: 'pozitív',
      importance: 0.55,
      explanation: 'A recept tartalmaz tápanyagban gazdag, egészséges összetevőket.'
    },
    // További táplálkozási tényezők...
  ],
  suggestions: [
    'Próbálja csökkenteni az állati eredetű összetevők mennyiségét vagy helyettesítse növér
    // További javaslatok...
  ],
  summary: 'A recept táplálkozási értéke jó, de környezeti hatása fejleszthető.',
  model: 'XAI-LIME-v1',
  confidence: 0.82
}
```

2.3 Alternatívák javaslata

Az XAI motor két típusú fenntarthatósági javaslatot kínál:

2.3.1 Hasonló, de fenntarthatóbb receptek

javascript

```
function findSimilarButMoreSustainableRecipes(recipe, allRecipes) {  
  // Kategória egyezés  
  const sameCategory = allRecipes.filter(r =>  
    r.category === recipe.category &&  
    r.recipeid !== recipe.recipeid &&  
    (r.sustainability_index || 0) > (recipe.sustainability_index || 0)  
  );  
  
  // Hozzávalók hasonlósága  
  // ...  
  
  // Hasonlósági pontszám (Jaccard index)  
  const similarityScore = commonIngredients.length /  
    (recipeIngredients.length + targetIngredients.length - commonIngredients.length);  
  
  // Kombinált pontszám (hasonlóság + fenntarthatósági különbség)  
  // ...  
}
```

2.3.2 Összetevő helyettesítési javaslatok

javascript

```
function suggestIngredientSubstitutions(recipe) {  
  // Ismert helyettesítések definíciója  
  const knownSubstitutions = {  
    'marhahús': {  
      replace: 'csirkehús',  
      improvementPercent: 40,  
      explanation: 'A csirkehús előállítása jelentősen kevesebb üvegházhatású gáz kibocsátást okoz, mint a marhahúsé.'  
    },  
    // További helyettesítések...  
  };  
  
  // Helyettesítési javaslatok keresése  
  // ...  
}
```

3. Felhasználói felületen való megjelenítés

3.1 XAI magyarázat vizualizáció (ui-components.js)

javascript

```
function generateXaiHTML(recipe, xaiExplanation) {
  // Környezeti tényezők HTML
  const envFactorsHtml = xaiExplanation.environmentalFactors.map(factor => `
    <div class="xai-factor ${factor.impact === 'pozitív' ? 'positive' : factor.impact === '
      <div class="factor-header">
        <span class="factor-name">${factor.name}</span>
        <span class="factor-impact ${factor.impact}>${factor.impact === 'pozitív' ? '
      </div>
      <div class="factor-explanation">${factor.explanation || ''}</div>
      <div class="factor-importance-bar">
        <div class="importance-fill" style="width: ${Math.round(factor.importance * 100
      </div>
    </div>
  `).join('');

  // Táplálkozási tényezők HTML
  // ...

  // Javaslatok HTML
  // ...

  return `
    <div class="eco-xai-section">
      <div class="eco-xai-header">
        🧠 AI Magyarázat - Miért ez a pontszám?
      </div>
      <div class="eco-xai-content">
        <!-- Részletes magyarázat HTML -->
      </div>
    </div>
  `;
}
```

3.2 Alternatívák megjelenítése

javascript

```
function generateAlternativesHTML(recipe, similarRecipes, substitutions) {
  // Hasonló receptek javaslatok
  // ...

  // Összetevő helyettesítési javaslatok
  // ...
}
```

4. XAI API integráció lehetősége

A rendszer jelenleg egy szimulált XAI API-t használ, de könnyen integrálható valódi AI szolgáltatásokkal:

javascript

```
async function callExternalXaiApi(recipeData) {
  const response = await fetch(XAI_CONFIG.API_ENDPOINT, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'Authorization': `Bearer ${XAI_CONFIG.API_KEY}`
    },
    body: JSON.stringify({
      model: XAI_CONFIG.MODEL,
      data: recipeData
    })
  });

  if (!response.ok) {
    throw new Error(`XAI API hívás sikertelen: ${response.status}`);
  }

  return await response.json();
}
```

Az API integráció a következő szolgáltatásokkal valósítható meg:

- **Azure Machine Learning Explainability**
- **IBM Watson OpenScale**
- **Google Cloud Explainable AI**
- **Saját LIME/SHAP implementáció**

5. Előnyök és korlátozások

5.1 Előnyök

- **Átláthatóság:** A felhasználók megértik, hogy miért kapott egy recept adott fenntarthatósági pontszámot
- **Személyre szabott javaslatok:** Konkrét módosítási lehetőségek a fenntarthatóság javítására
- **Oktatási érték:** A felhasználók tanulnak a fenntartható étkezésről
- **Bizalom építés:** Az átlátható magyarázatok növelik a rendszer iránti bizalmat

5.2 Korlátozások

- **Szimulált AI:** A jelenlegi implementáció szabályalapú szimulációt használ valódi gépi tanulási modell helyett
- **Korlátozott adatkészlet:** A GreenRec adatkészlet csak alapvető környezeti és táplálkozási pontszámokat tartalmaz
- **Egyszerűsített jellemzők:** A teljes körű értékeléshez részletesebb összetevő-szintű adatok lennének szükségesek

6. Jövőbeli fejlesztési lehetőségek

- **Valódi LIME/SHAP implementáció** gépi tanulási modellel
- **Interaktív magyarázatok**, ahol a felhasználók módosíthatják az összetevőket és láthatják a hatásukat
- **Részletesebb környezeti adatok** (CO2 lábnyom, vízlábnyom, stb.)
- **Személyre szabott magyarázatok** a felhasználói preferenciák alapján
- **Többnyelvű XAI** magyarázatok

7. Referenciák

1. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. KDD '16.
2. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. NIPS '17.
3. GreenRec: A Large-Scale Dataset for Green Food Recommendation, WWW '24 Companion, May 13–17, 2024, Singapore, Singapore.
4. Zhang, L., Zhang, Y., Zhou, X., & Shen, Z. (2024). GreenRec: A Large-Scale Dataset for Green Food Recommendation.