**Projekt Alapító Dokumentum (PAT)**

**Projekt Célja**

A projekt célja egy Python alapú interaktív alkalmazás létrehozása, amely népességi adatok vizualizálását és előrejelzését teszi lehetővé különböző statisztikai módszerek és grafikonok segítségével. Az alkalmazás a népesség adatait vizsgálja, és interaktív grafikonokat jelenít meg a múltbeli adatok és a jövőbeli trendek alapján.

**Projekt csapat**

* **Projektmenedzser**: Tóth Zsombor
* **Fejlesztők**: Tóth Zsombor
* **Tesztelők**: Tóth Zsombor

|  |  |  |  |
| --- | --- | --- | --- |
| **Feladat** | **Fejlesztő** | **Munkaidő (óra)** | **Leírás** |
| Projekt tervezése és struktúra meghatározása | Tóth Zsombor | 3 | A projekt céljainak meghatározása, szükséges eszközök és keretrendszerek kiválasztása. |
| Adatbetöltés és előkészítés | Tóth Zsombor | 2 | A .csv fájl beolvasása, tisztítása és megfelelő formátumba konvertálása. |
| Regresszió modell és előrejelzések készítése | Tóth Zsombor | 6 | A lineáris regresszió implementálása, jövőbeli predikciók generálása. |
| Grafikonok implementálása és vizualizációk | Tóth Zsombor | 12 | A különböző ábrák létrehozása: vonalgrafikon, kördiagramok és előrejelzések. |
| Felhasználói felület kialakítása | Tóth Zsombor | 10 | A **Tkinter** GUI tervezése, gombok és interaktív elemek hozzáadása. |
| Tesztelés és hibajavítás | Tóth Zsombor | Folyamatos | A program különböző részeinek tesztelése, hibák keresése és javítása. |
| Dokumentáció és végső simítások | Tóth Zsombor | 5 | A felhasználói dokumentáció és végső programoptimalizálás elkészítése. |

**Rendszerkövetelmények**

* Python 3.8 vagy újabb
* Kötelező csomagok:
  + tkinter
  + matplotlib
  + pandas
  + numpy
  + scikit-learn

**Felhasználói felület**

Az alkalmazás egy egyszerű, grafikus felhasználói felületet (GUI) biztosít a **Tkinter** keretrendszer segítségével. Az alábbi elemeket tartalmazza:

1. **Interaktív ábrák**: A felhasználó több diagramot és grafikonot tekinthet meg, amelyek a népességi adatok különböző aspektusait mutatják.
2. **Navigációs gombok**:
   * **Előző**: Visszalépés az előző ábrához.
   * **Következő**: Továbblépés a következő ábrára.

**Funkcionalitás**

**1. Adatok betöltése és előkészítése**

A program a népességi adatokat egy .csv fájlból olvassa be, amelyet a **Pandas** könyvtár kezel. Az adatok előkészítése magában foglalja:

* Az oszlopok és sorok megfelelő formázását.
* Hiányzó értékek (NaN) kezelését.
* Számformátum konverziót.

**Példa az adatfeldolgozásra:**

data = pd.read\_csv('nepesseg.csv', encoding='ISO-8859-2', delimiter=';')

data\_cleaned = data.iloc[1:, [0, 1, 2, 3]]

data\_cleaned.columns = ['Év', 'Férfi', 'Nő', 'Összesen']

# Numerikus konverzió

data\_cleaned['Év'] = pd.to\_numeric(data\_cleaned['Év'], errors='coerce')

for col in ['Férfi', 'Nő', 'Összesen']:

data\_cleaned[col] = data\_cleaned[col].str.replace(' ', '').apply(pd.to\_numeric, errors='coerce')

**2. Népesség trendek és regressziók**

A népességi adatok trendjeit lineáris regresszió segítségével vizsgáljuk. A jövőbeli értékeket például 2050-re és 2100-ra prediktáljuk.

**Példa a lineáris regresszióra:**

x = data\_cleaned['Év'].values.reshape(-1, 1)

y\_férfi = data\_cleaned['Férfi'].values

model = LinearRegression()

model.fit(x, y\_férfi)

# Predikció a 2050-es évre

year\_2050 = np.array([[2050]])

prediction\_2050 = model.predict(year\_2050)

**3. Interaktív diagramok megjelenítése**

A program több ábrát generál:

* **Vonalgrafikon**: A népesség alakulását mutatja az évek során nemek szerint.
* **Kördiagram**: Egy adott év férfi és női népességének arányát ábrázolja.
* **Előrejelzések**: A népesség várható alakulása a jövőben.

**Példa egy vonalgrafikon létrehozására:**

fig, ax = plt.subplots(figsize=(10, 6))

ax.scatter(x, y\_férfi, color='blue', label='Férfiak')

ax.plot(x, model.predict(x), color='blue', linestyle='-', label='Férfi regresszió')

ax.set\_title('Népesség trendje', fontsize=20)

ax.set\_xlabel('Év', fontsize=15)

ax.set\_ylabel('Népesség (millió fő)', fontsize=15)

ax.legend()

ax.grid(True)

**4. Felhasználói interakció**

A felhasználók a navigációs gombok segítségével válthatnak az ábrák között. Az aktuális diagram automatikusan frissül.

**Példa a gomb események kezelésére:**

self.prev\_button = tk.Button(self.button\_frame, text="Előző", command=self.prev\_plot)

self.next\_button = tk.Button(self.button\_frame, text="Következő", command=self.next\_plot)

def prev\_plot(self):

self.current\_page = (self.current\_page - 1) % len(self.figures)

self.show\_plot(self.figures[self.current\_page])

**Output**

A program többféle grafikus ábrát generál és megjelenít:

1. **Népesség száma**
   * Cél: A népesség változásának bemutatása.
2. **Kördiagram az aktuális év népességéről**
   * Cél: Az aktuális népesség férfi-nő arányának vizualizálása.
3. **Átlagéletkor előrejelzés**
   * Cél: A férfiak, nők és az összesített átlagéletkor alakulása.
4. **Aktuális év átlagéletkor**
   * Cél: Aktuális év átlag életkora oszlopdiagramon nemenként