MINISTRY OF EDUCATION OF REPUBLIC OF MOLDOVA

Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Department of Software Engineering and Automatics

# Report

on Artificial Intelligence Fundamentals
Laboratory Work nr. I

Performed by: **Boaghe Mariana, Ejova Ecaterina**, FAF -171, FAF-172

Verified by: **Mihail Gavrilița**, asist. univ.

Chișinău, 2021

# Contents

# 1 Introduction

A dream come true: Mariana and Katja got a position of developers in the "HeinleinAI" company. Our testing period began with the following assignment: *".. to create an expert system. Research the types of tourists that visit Luna-City. This system would allow the user to answer some questions about a possible tourist."*

The code was also received. This code was intended for an old version of python, about the distant 2006 (think only a few centuries ago!). Realizing that we will not be able to rewrite the code at least for version 3.9, we decided to write the code ourselves. We hope that our mentor will appreciate it and we will be able to become full-fledged employees of the company.

We invite you to take a comfortable seat and plunge into the ocean of history with us.

# 2 The Task

Searching for tourists, expertly Finally, your dream came true and you've landed a job at "HeinleinAI" – the biggest AI company in the whole Luna-City! You're on a testing period so you will want to make sure that you do your best. While you are daydreaming about free lunch, the mentor comes and hands you your first task – you'll need to build an expert system. They say that while going through the central library database they stumbled upon this ancient approach that looked like it could solve a particular lunar problem – detecting tourists. The tourists on Luna-City are a big source of income and while almost every salesman and hotel administrator can easily tell one from a Loonie, our mentor researches a more systematic way of detecting them. While this would prove a trivial task to the city's main computer, resources are scarce these days and so you will be researching alternative, more special approaches. For starters, research the types of tourists that visit Luna-City and collect a database of at least 5 tourist types and the criteria by which they can be distinguished from Loonies and between themselves (i.e. clothes, accent, gait, height and opinion on politics). After your database is done, create a system that would allow the user to answer some questions about a possible tourist. If the set of given answers matches a type of tourist from the database, this should be the system's answer. If on the other hand, the system determines that the person in question is a Loonie, the answer should be returned accordingly. Make sure to consider the case when the set of answers does not find a match in the database (highly improbable if you've done your research well). Another thing to consider is how the system will be deployed. The machines that are at your disposal are only capable of running Python code or Docker images. So if you choose to write your system in something exotic, like Prolog, make sure to package it all accordingly. To aid you in your task, the mentor gave you some ancient "lab" document and some Python code that he found in the library. Who knows how that might help..

# 3 Solution Description

The following libraries were used:

**pandas** is a data analysis library providing fast, flexible, and expressive data structures designed to work with relational or table-like data (SQL table or Excel spreadsheet). It is a fundamental high-level building block for doing practical, real world data analysis in Python.

**warnings**, because the following error came out when compared

```python
if newdf['Tourist'].values == 0
```

"FutureWarning: elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison".

So we choose to ignore:

```python
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Read input file:

```python
df = pd.read_csv('touristData.csv')
```

Next, we created a list of questions:

```python
questions = ['1. What calendar are you using?\n\n', '2.What is your mother tongue?\n\n', '3.What is precious to you?\n', '4.How do you define your gait?\n', '5.Your clothes?\n', '6.Your height?\n']
```

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects.

So, grab DataFrame rows where column = a specific value, our input value in terminal:

```python
newdf = df.loc[df[df.columns[z]] == answ]
```

Only the looni have zero in the "Tourist" column, all the others/tourists have 1. The system checks if the entered answer is in the line with 0, it means that this is a resident of the Looni City and no points are awarded to him/her. If 1, then the tourist gets points.

We used the syntax for item in sequence with sequence as range(len(questions)) to iterate over the numbers from 1 up to the length of the iterable questions:

```python
for q in range(len(questions))
```

Database **touristData.csv**:

| Tourist | Calendar | Language | Precious | Gait | Clothe |
|---|---|---|---|---|---|
| 0 | Lunar | English Portuguese Korean Yoruba | Air | simplistic | spaces |
| 1 | Gregorian | English and Italian languages | Family | confident | Black |
| 1 | Harmony with nature | Xenolinguists | Eywa | pieceful | Loincl |
| 1 | Georgian | English and French language | Tiara | confident | Dresse |
| 1 | Roman | Latin and Greek | Civic crown | Important | Tunic |
| 1 | Magic | Khuzdul | Snow white | normal | Overal |

## Conclusions

The program sums up the points and displays a message on what percentage the user is a tourist. Only the tourist gets 16 points for each answer, and the resident/looni gets 0.

# References

[1] Boaghe Mariana, Ejova Ecaterina. Source code for the laboratory work. Accessed February 11, 2021. X.

[2] The Python Standard Library https://docs.python.org/3/library/stdtypes.html.

[3] The Panda Libray https://malouche.github.io/teachingpython/panda_library.html

[4] https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html

[5] https://www.webpages.uidaho.edu/~stevel/cheatsheets/Pandas%20DataFrame%20Notes_12pages.pdf

# Appendix

Dear looni, below is our code. Enjoy!

Listing 1: Our Code

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import warnings

warnings.simplefilter(action='ignore', category=FutureWarning)

print('\nWelcome to Luna City! Please answer the following questions so the our system can identify you.\n')
df = pd.read_csv('touristData.csv')

print (df.columns)

questions = ['1. What calendar are you using?\n\n', '2.What is your mother tongue?\n\n', '3.What is precious to you?\n', '4.How do you define your gait?\n', '5.Your clothes?\n', '6.Your height?\n']
def check(n, answ, df):
    n_col = len(df.columns)
    z = 1
    looni = 0
    while z < n_col:
        if n+1 == z:
            newdf = df.loc[df[df.columns[z]] == answ]
            print (newdf) #to see if input value that is in DB refers to tourist or looni
            if newdf['Tourist'].values == 0:
                looni += 0
                print('looni—', looni)
            else:
                looni += 16
                print('tourist—', looni)
        z += 1
    return looni

list_all_answers=[]
for q in range(len(questions)):
    answ = input(questions[q])
    list_all_answers.append(check(q, answ, df))
    #check(q,answ, df)

print('\nYou are ', sum(list_all_answers),'% tourist. Welcome to the Luna City!')
```