

# Αναφορά Σχεδιασμού και Υλοποίησης Εφαρμογής

Ομάδα Metask Inc

Ιορδάνης Τσουτσούλης Π2019117  
Κωνσταντίνος Ζωγράφος inf2021058  
Βασίλειος Λάμπας inf2021118

2024

## Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>2</b>
<b>2</b>	<b>Σχεδιασμός Εφαρμογής</b>	<b>2</b>
2.1	Αρχιτεκτονική . . . . .	2
<b>3</b>	<b>Υλοποίηση</b>	<b>2</b>
3.1	Φόρτωση Δεδομένων . . . . .	2
3.2	Ανάλυση Δεδομένων . . . . .	3
<b>4</b>	<b>Αποτελέσματα</b>	<b>3</b>
4.1	2D Visualization . . . . .	3
<b>5</b>	<b>Συμπεράσματα</b>	<b>4</b>
<b>6</b>	<b>Συνεισφορά Ομάδας</b>	<b>4</b>
<b>7</b>	<b>UML Διάγραμμα</b>	<b>4</b>
<b>8</b>	<b>Κύκλος Ζωής Έκδοσης Λογισμικού</b>	<b>5</b>
8.1	AgileΜεθοδολογία . . . . .	5

## 1 Εισαγωγή

Η παρούσα αναφορά περιγράφει τη σχεδίαση, υλοποίηση, ανάλυση και τα αποτελέσματα της εφαρμογής ανάλυσης δεδομένων.

## 2 Σχεδιασμός Εφαρμογής

Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας Πύλη και τις βιβλιοθήκες Streamlit, pandas, scikit-learn και plotly.

### 2.1 Αρχιτεκτονική

Η αρχιτεκτονική της εφαρμογής περιλαμβάνει τα ακόλουθα στοιχεία:

- **Streamlit**: Χρησιμοποιείται για την ανάπτυξη της διεπαφής χρήστη.
- **pandas**: Για τη διαχείριση και ανάλυση των δεδομένων.
- **scikit-learn**: Για την εκπαίδευση μοντέλων μηχανικής μάθησης.
- **plotly**: Για τη δημιουργία διαδραστικών διαγραμμάτων.

## 3 Υλοποίηση

Η υλοποίηση της εφαρμογής περιλαμβάνει τα ακόλουθα βήματα:

### 3.1 Φόρτωση Δεδομένων

Η εφαρμογή επιτρέπει στους χρήστες να φορτώνουν αρχεία "Σ" ή Εξελ:

```
import pandas as pd
import streamlit as st

uploaded_file = st.file_uploader("Upload a CSV or Excel file", type=["csv", "xlsx"])
if uploaded_file is not None:
    data = pd.read_csv(uploaded_file)
    st.write(data)
```

## 3.2 Ανάλυση Δεδομένων

Περιλαμβάνει τον έλεγχο των προδιαγραφών των δεδομένων, την ανάλυση ΠΑ και t-SNE, καθώς και την εκπαίδευση μοντέλων μηχανικής μάθησης. Ο κώδικας για την ανάλυση των δεδομένων περιλαμβάνει:

```
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import plotly.express as px

# PCA Analysis
pca = PCA(n_components=2)
principal_components = pca.fit_transform(data)
fig_pca = px.scatter(principal_components, x=0, y=1, title='PCA Analysis')
st.plotly_chart(fig_pca, use_container_width=True)

# t-SNE Analysis
tsne = TSNE(n_components=2)
tsne_components = tsne.fit_transform(data)
fig_tsne = px.scatter(tsne_components, x=0, y=1, title='t-SNE Analysis')
st.plotly_chart(fig_tsne, use_container_width=True)
```

## 4 Αποτελέσματα

Τα αποτελέσματα των αναλύσεων παρουσιάζονται με τη βοήθεια διαγραμμάτων.

### 4.1 2D Visualization



Figure 1: PCA Analysis

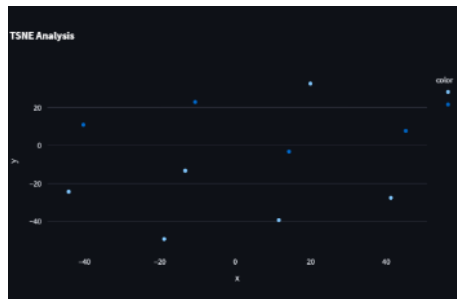


Figure 2: t-SNE Analysis

## 5 Συμπεράσματα

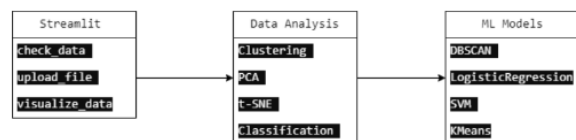
Η εφαρμογή επιτυγχάνει την ανάλυση και οπτικοποίηση δεδομένων, παρέχοντας χρήσιμα αποτελέσματα και συγκρίσεις αλγορίθμων.

## 6 Συνεισφορά Ομάδας

Η συνεισφορά κάθε μέλους της ομάδας ήταν η εξής:

- Ιορδάνης Τσουτσουλης: ασχολήθηκε με το διάβασμα των csv excel αρχείων
- Κωνσταντίνος Ζωγράφος: ανέλαβε να βρει ποιούς αλγορίθμους clustering και ομαδοποίησης θα χρησιμοποιήσουμε
- Βασίλειος Λάμπας :Ασχολήθηκε με τις τεχνικές οπτικοποίησης της εφαρμογής
- Εφόσων είχαμε έτοιμο τον κώδικα της εργασίας μας συνεχίσαμε με το host του site χρησιμοποιώντας το Docker. Επειτα φτιάξαμε και το Latex αρχείο όπου αναφέραμε τα όσα κάναμε!

## 7 UML Διάγραμμα



Σχήμα 3: UML Διάγραμμα της Εφαρμογής

## 8 Κύκλος Ζωής Έκδοσης Λογισμικού

Το μοντέλο κύκλου ζωής που ακολουθήσαμε για την ανάπτυξη της εφαρμογής είναι το Αγίλε. Το Αγίλε επιτρέπει την ευέλικτη και σταδιακή ανάπτυξη, διευκολύνοντας τη συνεχή βελτίωση και προσαρμογή στις απαιτήσεις των χρηστών.

### 8.1 AgileΜεθοδολογία

Η Αγίλε μεθοδολογία περιλαμβάνει τα ακόλουθα στάδια:

- **Σχεδιασμός:** Καθορισμός των απαιτήσεων και σχεδιασμός της αρχιτεκτονικής.
- **Ανάπτυξη:** Ανάπτυξη της εφαρμογής μέσω επαναληπτικών σπριντ.
- **Δοκιμή:** Δοκιμή και βελτίωση της εφαρμογής σε κάθε σπριντ.
- **Κυκλοφορία:** Διάθεση της εφαρμογής και υποστήριξη των χρηστών.

## 9 Github

<https://github.com/Metsk-Inc/TL2024/tree/main/TL2>