# **Θεωρία Αποφάσεων**

## **Εξαμηνιαία Εργασία**

### **Χρήση Τεχνικών Επιβλεπόμενης Μάθησης για την Ανακάλυψη Εξωπλανητών**

Τσικέλης Ιωάννης

Α.Μ: 1067407

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Πανεπιστήμιο Πατρών

2021-2022

## Σύντομη περιγραφή της υλοποίησης

Στα πλαίσια της εξαμηνιαίας εργασίας του μαθήματος, επιλέχθηκε να υλοποιηθεί ένα πρόγραμμα μηχανικής μάθησης για την απόφαση ύπαρξης ή μη εξωπλανητών, με βάση δεδομένα από το διαστημικό τηλεσκόπιο [Kepler](https://en.wikipedia.org/wiki/Kepler_space_telescope). Το dataset, είναι διαθέσιμο [εδώ](https://exoplanetarchive.ipac.caltech.edu/docs/KeplerMission.html). Σχετικά με την υλοποίηση, αρχικά έγινε καθάρισμα και ανάλυση των δεδομένων (PCA και κανονικοποίηση). Έπειτα, έγινε εκπαίδευση μοντέλων Τεχνητής Νοημοσύνης (Logistic Regression και SVMs) και αξιολόγησή τους. Ο κώδικας της υλοποίησης είναι διαθέσιμος και στο version control tool του [GitHub](https://github.com/itsikelis/DecisionTheory).

import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

*# Sklearn: Data normalisation, regression models and metrics.*

from sklearn.decomposition import PCA

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn import metrics

## Καθαρισμός και Ανάλυση

### Καθαρισμός Δεδομένων

Στο πρώτο βήμα της υλοποίησης, αφού τα δεδομένα φορτώθηκαν σε ένα pandas dataframe, με βάση την παρακάτω λίστα, που εξηγεί τις ιδιότητες της κάθε μέτρησης, αποφασίστηκαν οι εξαρτημένη και οι ανεξάρτητες μεταβλητές:

|  |
| --- |
| # COLUMN kepid: KepID |
| # COLUMN kepoi\_name: KOI Name |
| # COLUMN kepler\_name: Kepler Name |
| # COLUMN koi\_disposition: Exoplanet Archive Disposition |
| # COLUMN koi\_pdisposition: Disposition Using Kepler Data |
| # COLUMN koi\_score: Disposition Score |
| # COLUMN koi\_fpflag\_nt: Not Transit-Like False Positive Flag |
| # COLUMN koi\_fpflag\_ss: Stellar Eclipse False Positive Flag |
| # COLUMN koi\_fpflag\_co: Centroid Offset False Positive Flag |
| # COLUMN koi\_fpflag\_ec: Ephemeris Match Indicates Contamination False Positive Flag |
| # COLUMN koi\_period: Orbital Period [days] |
| # COLUMN koi\_period\_err1: Orbital Period Upper Unc. [days] |
| # COLUMN koi\_period\_err2: Orbital Period Lower Unc. [days] |
| # COLUMN koi\_time0bk: Transit Epoch [BKJD] |
| # COLUMN koi\_time0bk\_err1: Transit Epoch Upper Unc. [BKJD] |
| # COLUMN koi\_time0bk\_err2: Transit Epoch Lower Unc. [BKJD] |
| # COLUMN koi\_impact: Impact Parameter |
| # COLUMN koi\_impact\_err1: Impact Parameter Upper Unc. |
| # COLUMN koi\_impact\_err2: Impact Parameter Lower Unc. |
| # COLUMN koi\_duration: Transit Duration [hrs] |
| # COLUMN koi\_duration\_err1: Transit Duration Upper Unc. [hrs] |
| # COLUMN koi\_duration\_err2: Transit Duration Lower Unc. [hrs] |
| # COLUMN koi\_depth: Transit Depth [ppm] |
| # COLUMN koi\_depth\_err1: Transit Depth Upper Unc. [ppm] |
| # COLUMN koi\_depth\_err2: Transit Depth Lower Unc. [ppm] |
| # COLUMN koi\_prad: Planetary Radius [Earth radii] |
| # COLUMN koi\_prad\_err1: Planetary Radius Upper Unc. [Earth radii] |
| # COLUMN koi\_prad\_err2: Planetary Radius Lower Unc. [Earth radii] |
| # COLUMN koi\_teq: Equilibrium Temperature [K] |
| # COLUMN koi\_teq\_err1: Equilibrium Temperature Upper Unc. [K] |
| # COLUMN koi\_teq\_err2: Equilibrium Temperature Lower Unc. [K] |
| # COLUMN koi\_insol: Insolation Flux [Earth flux] |
| # COLUMN koi\_insol\_err1: Insolation Flux Upper Unc. [Earth flux] |
| # COLUMN koi\_insol\_err2: Insolation Flux Lower Unc. [Earth flux] |
| # COLUMN koi\_model\_snr: Transit Signal-to-Noise |
| # COLUMN koi\_tce\_plnt\_num: TCE Planet Number |
| # COLUMN koi\_tce\_delivname: TCE Delivery |
| # COLUMN koi\_steff: Stellar Effective Temperature [K] |
| # COLUMN koi\_steff\_err1: Stellar Effective Temperature Upper Unc. [K] |
| # COLUMN koi\_steff\_err2: Stellar Effective Temperature Lower Unc. [K] |
| # COLUMN koi\_slogg: Stellar Surface Gravity [log10(cm/s\*\*2)] |
| # COLUMN koi\_slogg\_err1: Stellar Surface Gravity Upper Unc. [log10(cm/s\*\*2)] |
| # COLUMN koi\_slogg\_err2: Stellar Surface Gravity Lower Unc. [log10(cm/s\*\*2)] |
| # COLUMN koi\_srad: Stellar Radius [Solar radii] |
| # COLUMN koi\_srad\_err1: Stellar Radius Upper Unc. [Solar radii] |
| # COLUMN koi\_srad\_err2: Stellar Radius Lower Unc. [Solar radii] |
| # COLUMN ra: RA [decimal degrees] |
| # COLUMN dec: Dec [decimal degrees] |
| # COLUMN koi\_kepmag: Kepler-band [mag] |

Σαν εξαρτημένη μεταβλητή χρησιμοποιήθηκε το γνώρισμα koi\_pdisposition στο οποίο δόθηκαν οι τιμές 0: για την εσφαλμένη μέτρηση , 1: για την πιθανή ύπαρξη εξωπλανήτη. Τέλος, αφαιρέθηκαν από τον πίνακα όσες μετρήσεις περιείχαν NaNs.

*# List of unwanted dataframe columns(categoricals, system flags, indexes and null columns).*

unwanted\_cols = ['rowid', 'kepid', 'kepoi\_name', 'kepler\_name', \_disposition', 'koi\_score', 'koi\_fpflag\_nt', 'koi\_fpflag\_ss', 'koi\_fpflag\_co', 'koi\_fpflag\_ec', 'koi\_teq\_err1', 'koi\_teq\_err2', 'koi\_tce\_delivname']

*# Declare dependent and independent variables.*

dependent\_variable = 'koi\_pdisposition'

independent\_variables = ['koi\_period', 'koi\_period\_err1', 'koi\_period\_err2', 'koi\_time0bk', 'koi\_time0bk\_err1', 'koi\_time0bk\_err2', 'koi\_impact', 'koi\_impact\_err1', 'koi\_impact\_err2', 'koi\_duration', 'koi\_duration\_err1', 'koi\_duration\_err2', 'koi\_depth', 'koi\_depth\_err1', 'koi\_depth\_err2', 'koi\_prad', 'koi\_prad\_err1', 'koi\_prad\_err2', 'koi\_teq', 'koi\_insol', 'koi\_insol\_err1', 'koi\_insol\_err2', 'koi\_model\_snr', 'koi\_tce\_plnt\_num', 'koi\_steff', 'koi\_steff\_err1', 'koi\_steff\_err2', 'koi\_slogg', 'koi\_slogg\_err1', 'koi\_slogg\_err2', 'koi\_srad', 'koi\_srad\_err1', 'koi\_srad\_err2', 'ra', 'dec', 'koi\_kepmag']

*# Turn categoricals to ints in dependent variable.*

df[dependent\_variable].replace(

    ['CANDIDATE', 'FALSE POSITIVE'], [1, 0], *inplace*=True)

*# Drop unwanted columns from dataframe.*

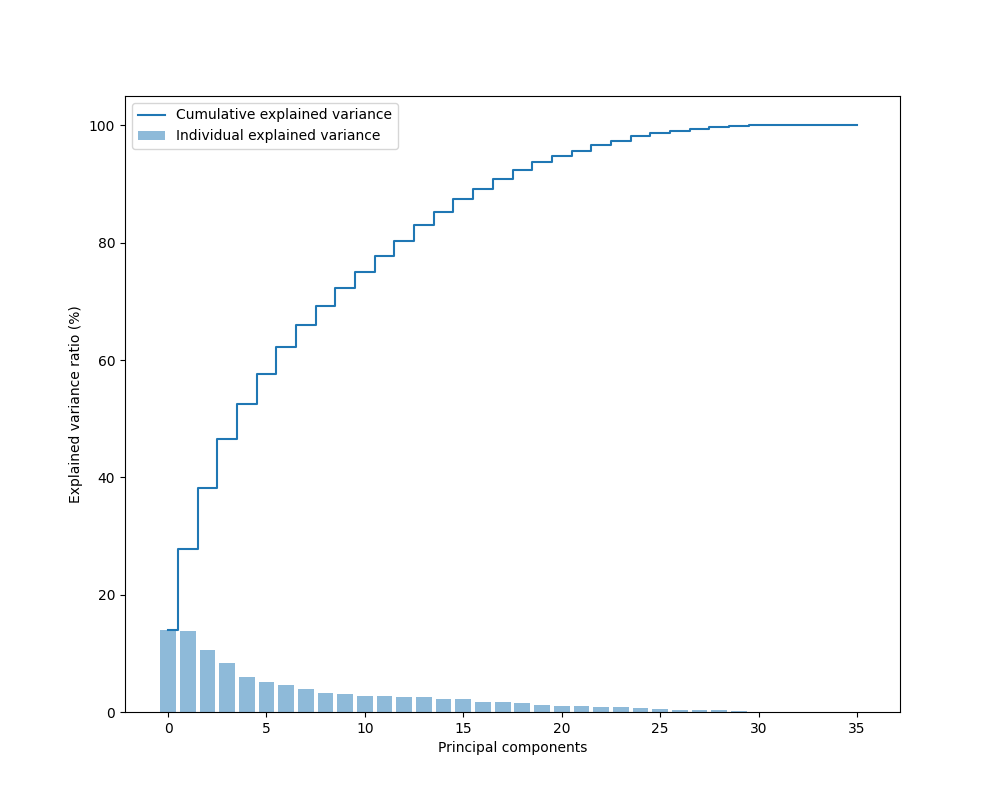
df.drop(unwanted\_cols, *axis*='columns', *inplace*=True)

*# Remove all rows with NaN values.*

df.dropna(*inplace*=True)

### Principal Component Analysis

Για την ανάλυση των δεδομένων, χρησιμοποιήθηκε η τεχνική Principal Component Analysis. Ο κώδικας για το script αυτό βρίσκεται στο αρχείο variance\_calculation.py. Τα ποσοστά συνεισφοράς της κάθε μεταβλητής στην απόφαση, φαίνονται στο παρακάτω διάγραμμα:



Παρατηρώντας το διαγράμμα, εξάγεται το συμπέρασμα ότι το 95% της απόφασης μπορεί να ληφθεί με βάση τις πρώτες 20 μεταβλητές. Γίνεται λοιπόν, μέσω της PCA μεθόδου να μειωθούν τα βασικά στοιχεία του dataset, για αποφυγή πολλών υπολογισμών.

### Normalisation & Centering

Για την αποφυγή διακυμάνσεων στα μοντέλα και περιττών υπολογισμών, επιλέχθηκε να γίνει κανονικοποίηση στα δεδομένα ειδόδου. Τα κανονικοποιημένα δεδομένα ελέγχθηκαν ξεχωριστά από τα δεδομένα που προέκυψαν από την PCA. Η κανονικοποίηση έγινε ξεχωριστά για τα δεδομένα εκπαίδευσης και ελέγχου:

*# Normalise train data.*

sc\_train = StandardScaler()                 *# Instantiate scaler object.*

X\_train = sc\_train.fit\_transform(X\_train)   *# Normalise data in X\_train.*

*# Normalise test data.*

sc\_test = StandardScaler()                  *# Instantiate scaler object.*

X\_test = sc\_test.fit\_transform(X\_test)      *# Normalise data in X\_test.*

## Models

### Logistic Regression

Για το μοντέλο Logistic Regression χρησιμοποιήθηκαν δύο είδη των δεδομένων. Τα δεδομένα που προέκυψαν μετά από την ανάλυση PCA και τα δεδομένα μετά την ξεχωριστή κανονικοποίηση των train και test υποσυνόλων.

Τα αποτελέσματα έχουν ως εξής:

Normalisation Logistic Regression Results:

Model Accuracy (%): 76.68681587975821

Confusion Matrix :

[[2258 828]

[ 599 2436]]

Classification Report (1: Candidate Exoplanet, 0: False Positive Measurement ) :

precision recall f1-score support

0 0.79 0.73 0.76 3086

1 0.75 0.80 0.77 3035

accuracy 0.77 6121

macro avg 0.77 0.77 0.77 6121

weighted avg 0.77 0.77 0.77 6121

Maximum mean values of train/test sets before normalisation:

Train set: 22935.690354555853

Test set: 23823.096046397644

Maximum mean values of train/test sets before normalisation:

Train set: 3.0935562494778285e-15

Test set: 1.9246525990691818e-15

Normalisation Logistic Regression Results:

Model Accuracy (%): 81.52262702172848

Confusion Matrix :

[[2398 688]

[ 443 2592]]

Classification Report (1: Candidate Exoplanet, 0: False Positive Measurement ) :

precision recall f1-score support

0 0.84 0.78 0.81 3086

1 0.79 0.85 0.82 3035

accuracy 0.82 6121

macro avg 0.82 0.82 0.82 6121

weighted avg 0.82 0.82 0.81 6121

### SVMs

Για τον περαιτέρω έλεγχο υλοποιήθηκε και ένα μοντέλο SVM. Τα αποτελέσματά του έχουν ως εξής:

Model Accuracy (%): 82.42117301094592

Confusion Matrix :

[[2424 662]

[ 414 2621]]

Classification Report (1: Candidate Exoplanet, 0: False Positive Measurement ) :

precision recall f1-score support

0 0.85 0.79 0.82 3086

1 0.80 0.86 0.83 3035

accuracy 0.82 6121

macro avg 0.83 0.82 0.82 6121

weighted avg 0.83 0.82 0.82 6121

Τέλος, καταγράφηκαν και οι γραφικές παραστάσεις ROC για όλα τα μοντέλα.

