



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πρόγραμμα Μεταπτυχιακών Σπουδών**

**«ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ, ΑΣΦΑΛΕΙΑ ΚΑΙ ΑΝΑΔΥΟΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ  
ΠΛΗΡΟΦΟΡΙΑΣ»**

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Ανάπτυξη Διαδικτυακής Εφαρμογής Περιβάλλουσας Ανάλυσης Δεδομένων (ΠΑΔ)</b>  <b>Web Application for Data Envelopment Analysis (DEA)</b>
Όνοματεπώνυμο Φοιτητή	<b>ΗΛΙΑΣ – ΣΠΥΡΙΔΩΝ ΒΛΑΧΟΣ</b>
Πατρώνυμο	<b>ΒΑΣΙΛΕΙΟΣ</b>
Αριθμός Μητρώου	<b>ΜΠΚΣΑ18007</b>
Επιβλέπων	<b>Κορωνάκος Γρηγόριος</b>

Ημερομηνία Παράδοσης

11 2022

**Τριμελής Εξεταστική Επιτροπή**

Γρηγόριος Κορωνάκος

Δημήτριος Αποστόλου

Διονύσιος Σωτηρόπουλος

Καθηγητής

Επίκουρος Καθηγητής

## Περίληψη

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής αναπτύχθηκε διαδικτυακή εφαρμογή η οποία στοχεύει στη μέτρηση της αποδοτικότητας των παραγωγικών μονάδων μέσω των τεχνικών της Περιβάλλουσας Ανάλυσης Δεδομένων (ΠΑΔ). Η ΠΑΔ είναι μια μη παραμετρική τεχνική για τη μέτρηση της αποδοτικότητας ενός συνόλου οντοτήτων που ονομάζονται παραγωγικές μονάδες ή decision-making units (DMUs) και οι οποίες μετατρέπουν πολλαπλές εισροές, σε πολλαπλές εκροές. Η προσέγγιση της ΠΑΔ είναι αρκετά διαδεδομένη στη μέτρηση της αποδοτικότητας λόγω του ότι μπορεί να εφαρμοστεί σε πολλά είδη DMUs, όπως σχολεία, τράπεζες, νοσοκομεία κ.α. Η ανάπτυξη εύκολα προσβάσιμων εργαλείων όπως οι διαδικτυακές εφαρμογές, για τη μέτρηση της αποδοτικότητας κρίνεται αρκετά χρήσιμη, καθώς μεγάλος αριθμός αναλυτών αποφεύγουν να εφαρμόσουν την τεχνική λόγω της σχετικής δυσκολίας της στον ορισμό των γραμμικών προβλημάτων. Η εφαρμογή που αναπτύχθηκε λαμβάνει δεδομένα μέσω αρχείου excel, κατασκευάζει τα γραμμικά προβλήματα και τα επιλύει μέσω της βιβλιοθήκης PuLP, ο solver της η οποία χρησιμοποιεί τη μέθοδο simplex.

## Abstract

In the context of this Master's thesis, a web application which aims to assist in the efficiency measurement of production units with Data Envelopment Analysis (DEA) techniques was developed. DEA is a non-parametric technique used for measuring the efficiency of a set of decision making units (DMUs) which convert multiple inputs into multiple outputs. DEA can be applied in many different types of DMUs for efficiency measurement, like schools, banks, hospitals etc. Easily accessible tools like web applications for efficiency measurement are quite useful, as analysts are prevented from applying the technique because of the learning curve in defining the linear problems. This web application reads data from an excel file, models the linear problems and solves them through PuLP using its solver which implements the simplex method.

## Περιεχόμενα

<b>Περίληψη</b>	<b>2</b>
<b>Abstract</b>	<b>2</b>
<b>Περιεχόμενα</b>	<b>3</b>
<b>Εισαγωγή</b>	<b>5</b>
<b>1. Περιβάλλουσα Ανάλυση Δεδομένων (Data envelopment analysis)</b>	<b>5</b>
Μεθοδολογία	7
Μοντέλο CRS	7
Μοντέλο VRS	10
<b>2. Τεχνολογίες</b>	<b>11</b>
Γλώσσα Προγραμματισμού	11
Framework	12
Backend	12
Frontend	12
Βάση δεδομένων	12
Βιβλιοθήκη PuLP	13
Περιβάλλουσα Ανάλυση Δεδομένων με PuLP	13
Αποτελέσματα	17
<b>3. Υλοποίηση</b>	<b>18</b>
Διαγράμματα	18
Class diagram	18
Διάγραμμα ροής	19
Βάση δεδομένων	20
Αρχιτεκτονική Django	22
Περιγραφή	22
Development Server	22
Django Apps	22
Models	23
Django admin site	23
Πρόσβαση στο django admin	23
Docker	23
Docker Compose	24
Πακέτα - Βιβλιοθήκες	24
Βιβλιοθήκη PuLP	24
<b>4. Χρήση εφαρμογής</b>	<b>24</b>

<b>Μελέτη Περίπτωσης</b>	<b>33</b>
CRS Μοντέλο	36
VRS Μοντέλο	41
<b>Συμπεράσματα</b>	<b>44</b>
<b>Βιβλιογραφία</b>	<b>46</b>
Ηλεκτρονικές πηγές	46
<b>Παράρτημα κώδικα</b>	<b>47</b>
Github	47
Μοντέλα	47
Views	48

## Εισαγωγή

Για να βελτιωθεί η αποδοτικότητα ενός οργανισμού απαιτείται πολύ καλή εκτίμηση της εσωτερικής δομής και των λειτουργιών του, η πολυπλοκότητα των οποίων όμως, δυσκολεύει τη μέτρηση της αποδοτικότητας. Ένας τρόπος για να μετρηθεί, είναι η παραμετρική μέθοδος σύμφωνα με την οποία θα πρέπει να δημιουργηθεί μια συνάρτηση παραγωγής η οποία αποτυπώνει τη σχέση μεταξύ των εισροών και των εκροών οι οποίες συμμετέχουν στη διαδικασία παραγωγής. Αυτή η συνάρτηση όμως είναι συχνά πολύ δύσκολο έως αδύνατον να κατασκευαστεί ή εντελώς άγνωστη. Αντίθετα, η μη παραμετρική μέθοδος δεν απαιτεί τον εκ των προτέρων προσδιορισμό κάποιας μορφής η οποία συνδέει τις εισροές με τις εκροές.

Ο Farrell το 1957 διατύπωσε μια διαφορετική πρόταση για την μέτρηση της αποδοτικότητας με εμπειρικά δεδομένα, εκφράζοντάς την, ως το ηγίλιο των εκροών προς τις εισροές.

$$Efficiency = \frac{total\ outputs}{total\ inputs}$$

Το 1978 οι Charnes, Cooper και Rhodes, βασιζόμενοι στην αρχική πρόταση του Farrell, εισήγαγαν μια νέα τεχνική για την αποτίμηση της αποδοτικότητας, την Περιβάλλουσα Ανάλυση Δεδομένων. Αρχικά χρησιμοποιήθηκε για την εκτίμηση της αποδοτικότητας μη κερδοσκοπικών οργανισμών, αλλά στη συνέχεια χρησιμοποιήθηκε ως μεθοδολογία σε πολλούς τομείς και έχει καταφέρει να ανακαλύψει βελτιώσεις οι οποίες περνούσαν απαρατήρητες με άλλες τεχνικές.

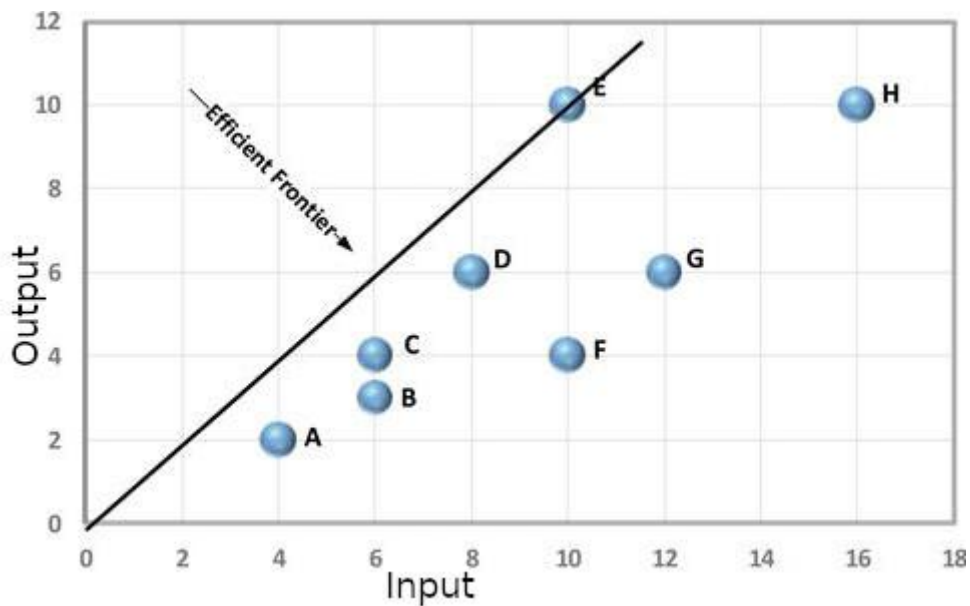
Οι οντότητες των οποίων η αποδοτικότητα μετριέται στην ΠΑΔ ονομάζονται παραγωγικές μονάδες ή αλλιώς μονάδες απόφασης, όρος ο οποίος μπορεί να χρησιμοποιηθεί για να περιγράψει πολλών ειδών οντοτήτων, αφού η ΠΑΔ βρίσκει εφαρμογή από νοσοκομεία και σχολεία, μέχρι ολόκληρες χώρες. Οι μονάδες απόφασης που αξιολογούνται πρέπει να ανήκουν σε ένα σύνολο μονάδων ίδιου είδους, οι οποίες χρησιμοποιούν ίδιου τύπου και αριθμού εκροές, για να παράξουν ίδιου τύπου και αριθμού εκροές. Το αποτέλεσμα της αξιολόγησης κάθε μονάδας ή αλλιώς βαθμός απόδοσής της θα είναι ένας αριθμός μεταξύ 0 και 1 (Cooper, Seiford, Tone 2006).

$$0 \leq E \leq 1$$

### 1. Περιβάλλουσα Ανάλυση Δεδομένων (Data envelopment analysis)

Η Περιβάλλουσα Ανάλυση Δεδομένων (ΠΑΔ) - (Data envelopment analysis - DEA) είναι μια μη παραμετρική τεχνική που χρησιμοποιείται ευρέως για τη μέτρηση της αποδοτικότητας ενός συνόλου συγκρίσιμων οντοτήτων, που ονομάζονται παραγωγικές μονάδες (Decision Making Units - DMUs) και καταναλώνουν πολλαπλές εισροές για να παράξουν πολλαπλές εκροές. Η ΠΑΔ εισήχθη από τους Charnes et al (1978) και Banker et al (1984), χρησιμοποιεί τον γραμμικό προγραμματισμό για να προσδιορίσει το σύνολο αποδοτικότητας ώστε να υπολογίσει το βαθμό απόδοσης κάθε παραγωγικής μονάδας. Επειδή κάθε μονάδα αξιολογείται σε σχέση με τις μονάδες βέλτιστης πρακτικής, οι αποδόσεις που παρέχει η ΠΑΔ είναι σχετικές και όχι απόλυτες (Κορονακός, 2019).

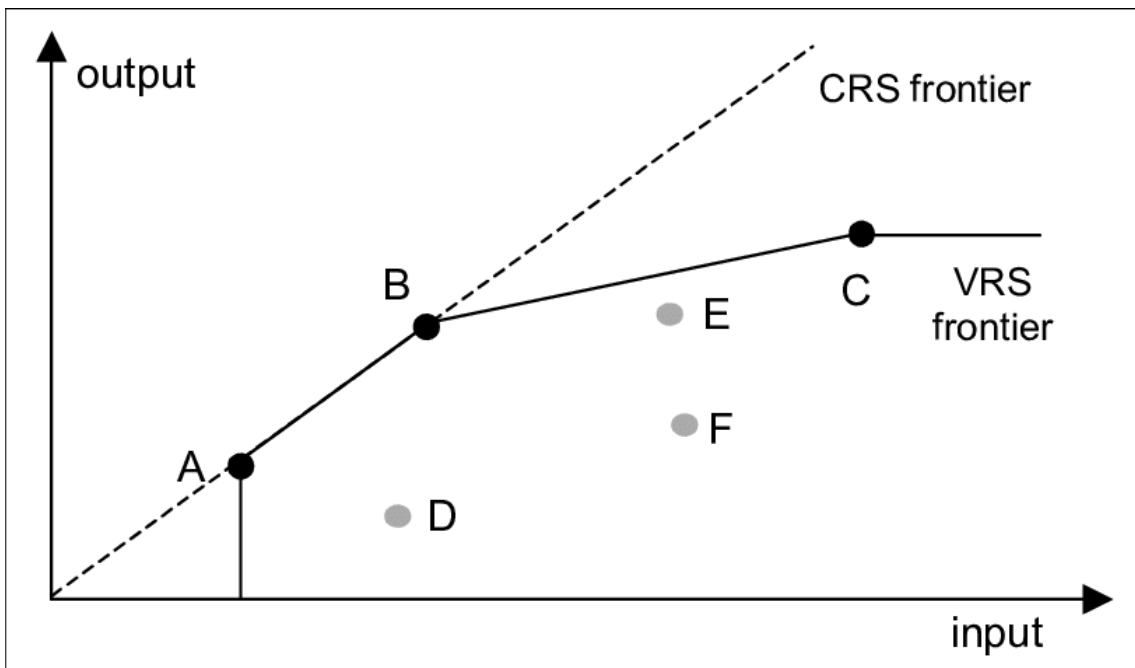
Οι αποδοτικές μονάδες απόφασης ορίζουν το σύνολο αποδοτικότητας και βρίσκονται πάνω σε αυτό, ενώ οι μη αποδοτικές περιβάλλονται από αυτό.



**Εικόνα 1.1: Σύνολο παραγωγικών δυνατοτήτων**

Στην παραπάνω εικόνα απεικονίζεται ένα σύνολο παραγωγικών μονάδων οι οποίες καταναλώνουν μια εισροή και παράγουν μια εκροή. Ο οριζόντιος άξονας αναπαριστά την εισροή και ο κάθετος στην εκροή. Η ευθεία που διέρχεται από την αρχή των αξόνων και από κάθε μονάδα, αναπαριστά το βαθμό αποδοτικότητάς της, δηλαδή το πηλίκο  $\frac{output}{input}$ , επομένως μεγαλύτερη κλίση δηλώνει καλύτερη αποδοτικότητα και η ευθεία με τη μεγαλύτερη κλίση ορίζεται ως το σύνολο αποδοτικότητας. Όλες οι μονάδες εκτός της E, περιβάλλονται από την ευθεία που διέρχεται από την αρχή των αξόνων και την E, άρα η E εκτιμάται ως η μονάδα με τη μεγαλύτερη σχετική αποδοτικότητα, ενώ οι υπόλοιπες μονάδες κρίνονται ως μη αποδοτικές. Η σχετική αποδοτικότητα της αποδοτικής μονάδας E είναι  $Eff_E = 1$ , ενώ η σχετική αποδοτικότητα των υπολοίπων μονάδων θα είναι  $0 \leq Eff \leq 1$ .

Το σύνολο παραγωγικών δυνατοτήτων στην παραπάνω εικόνα βασίζεται σε μια συγκεκριμένη υπόθεση, κατά την οποία θεωρούμε ότι υπάρχει αναλογική σχέση μεταξύ εισροών και εκροών. Σε αυτή την περίπτωση, όπου μια αύξηση των εισροών θα οδηγούσε σε ανάλογη αύξηση των εκροών, θεωρούμε ότι οι μονάδες απόφασης λειτουργούν με την κλίμακα σταθερών αποδόσεων (constant returns to scale - CRS). Ως αποτέλεσμα, το σύνολο αποδοτικότητας είναι μια ευθεία. Η DEA κατασκευάζει το σύνολο των παραγωγικών μονάδων, για να βρει τις μονάδες καλύτερης πρακτικής, σύμφωνα με την κλίμακα αποδόσεων. Η κλίμακα αποδόσεων ορίζει τον τρόπο με τον οποίο μια αλλαγή στις εισροές, επηρεάζει τις εκροές. Συχνά υπάρχουν παράγοντες, όπως τεχνολογικοί περιορισμοί, που επηρεάζουν την κλίμακα αποδόσεων με αποτέλεσμα αλλαγές στις εισροές να μην εκφράζονται κατά τον ίδιο τρόπο στις εκροές. Τότε θεωρούμε ότι οι μονάδες απόφασης λειτουργούν με την κλίμακα μεταβλητών αποδόσεων.



Εικόνα 1.2: CRS και VRS σύνορο αποδοτικότητας (Ceyhan, M 2008)

Στην περίπτωση της κλίμακας μεταβλητών αποδόσεων (variable returns to scale - VRS), το σύνορο αποδοτικότητας ορίζεται από μια κυρτή τεθλασμένη γραμμή, η οποία περνά από τις αποδοτικές μονάδες. Στην εικόνα 1.2 βλέπουμε ότι μόνο οι μονάδες A και B κρίνονται αποδοτικές βάσει της κλίμακας σταθερών αποδόσεων, ενώ βάσει της κλίμακας μεταβλητών αποδόσεων και η μονάδα C χαρακτηρίζεται ως αποδοτική. (Κορωνάκος, 2017)

#### Μεθοδολογία

#### Μοντέλο **CRS**

Οι Carnes, Cooper και Rhodes (1978) πρότειναν ένα μοντέλο για την αποτίμηση της αποδοτικότητας το οποίο γενικεύει την έννοια των εκροών και εισροών και μπορούσε να αποδοθεί σε πολλαπλές εκροές και εισροές, μετατρέποντάς τες σε σταθμισμένο άθροισμα εκροών και εισροών. Οι συντελεστές κάθε εκροής ή εισροής, ονομάζονται βάρη και δεν δίνονται εκ των προτέρων, αλλά αποφασίζονται από την ίδια την αποτιμώμενη μονάδα, με σκοπό να μεγιστοποιήσει την αποδοτικότητά της (Δεσπότης). Στο μοντέλο τους η αποδοτικότητα ορίζεται από την παρακάτω σχέση:

$$Ej = \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}}$$

Όπου:

j είναι η αποτιμώμενη μονάδα,



$y_{rj}$  είναι το διάνυσμα με τις εκροές της μονάδας  $j$

$x_{ij}$  είναι το διάνυσμα με τις εισροές της μονάδας  $j$

$n$  είναι το σύνολο των DMU

$s$  το σύνολο των εκρών

$m$  το σύνολο των εισροών

τα  $u$  και  $v$  είναι τα βάρη των εκρών και εισροών αντίστοιχα

Το CCR μοντέλο υπολογίζει την αποδοτικότητα κάθε μονάδας βασιζόμενο στην κλίμακα σταθερών αποδόσεων και εκφράζεται ως μαθηματικό πρόβλημα με την παρακάτω μορφή:

$\max E_j = \frac{uy_j}{vx_j}$ <p>Subject to</p> $\frac{uy_j}{vx_j} \leq 1, j = 1, \dots, n$ $u, v \geq 0$	(1.1)
--	-------

Για να μετατρέψουμε το παραπάνω πρόβλημα σε γραμμικό, θα υποθέσουμε ότι  $vx_j = 1$ . Επομένως η γραμμική μορφή του προβλήματος είναι η παρακάτω:

$\max E_j = uy_j$ <p>s.t.</p> $vx_j = 1$ $uY_j - vX_j \leq 0$ $u \geq 0, v \geq 0$	(1.2)
--	-------

Στο παραπάνω πρόβλημα έχει ληφθεί υπόψη ο προσανατολισμός με βάση τις εισροές (input orientation), σύμφωνα με τον οποίο μια μονάδα είναι μη αποδοτική όταν είναι εφικτό να μειωθεί μια εισροή και χωρίς να αυξηθεί κάποια άλλη, δε θα μειωθεί καμία εκροή. Αντίθετα, για να είναι μια μονάδα αποδοτική, θα πρέπει  $E_j = 1$  και να υπάρχει τουλάχιστον μία βέλτιστη λύση η οποία να καλύπτει τους περιορισμούς για  $u, v > 0$  ώστε να μην υπάρχουν μηδενικά βάρη με αποτέλεσμα να

παραλείπεται κάποια εισροή ή εκροή. Η βελτιστοποίηση θα πρέπει να πραγματοποιηθεί για κάθε μονάδα απόφασης.

Η παραπάνω μορφή του γραμμικού προβλήματος αναφέρεται ως Multiplier form, ορίζει δηλαδή βάρη με τα οποία πολλαπλασιάζονται οι εισροές και οι εκροές. Η δυϊκή μορφή του Multiplier προβλήματος αναφέρεται ως Envelopment μορφή, η οποία προβάλλει τις αποτιμώμενες μονάδες επάνω στο σύνоро αποδοτικότητας (Koropakos, 2017). Η προβολή αυτή ορίζεται από ένα γραμμικό συνδυασμό των μονάδων απόφασης, ο οποίος στην input oriented περίπτωση παράγει εκροές τουλάχιστον τόσες όσες η αποτιμώμενη μονάδα, καταναλώνοντας όμως όσο το δυνατόν λιγότερες εισροές. Η envelopment μορφή του παραπάνω γραμμικού προβλήματος δίνεται παρακάτω:

$\begin{aligned} & \min \theta \\ & \text{s.t.} \\ & X\lambda - \theta x_j \leq 0 \\ & Y\lambda - y_j \geq 0 \\ & \lambda \geq 0 \end{aligned}$	(1.3)
---	-------

Στη βέλτιστη λύση το  $\theta$  θα είναι  $0 < \theta \leq 1$

Στην περίπτωση όπου για τους σκοπούς της ανάλυσης μας ενδιαφέρει περισσότερο η επίτευξη υψηλότερου ποσού εκροών, θα επιλέξουμε το μοντέλο με προσανατολισμό προς τις εκροές (output orientation). Επομένως, το γραμμικό πρόβλημα στο CCR μοντέλο, στην Multiplier μορφή του, με προσανατολισμό προς τις εκροές μπορεί να εκφραστεί ως:

$\begin{aligned} & \min \frac{1}{E_j} = vX_j \\ & \text{s.t.} \\ & uy_j = 1 \\ & uY_j - vX_j \leq 0 \\ & v \geq 0 \text{ και } u \geq 0 \end{aligned}$	(1.4)
--	-------

Ενώ στην Envelopment μορφή του ως εξής:

$\begin{aligned} & \max \theta \\ & \text{s.t.} \\ & X\lambda - x_j \leq 0 \\ & Y\lambda - \theta y_j \geq 0 \\ & \lambda \geq 0 \end{aligned}$	(1.5)
---	-------

Στο CCR μοντέλο με προσανατολισμό προς τις εκροές, μια μονάδα θα είναι μη αποδοτική όταν είναι εφικτό να αυξηθεί κάποια εκροή χωρίς να μειωθεί κάποια άλλη και χωρίς να αυξηθεί καμία εισροή.

### Μοντέλο **VRS**

Οι Banker, Charnes και Cooper το 1984 πρότειναν μια παραλλαγή του βασικού μοντέλου της ΠΑΔ, το οποίο ενσωματώνει την κλίμακα μεταβλητών αποδόσεων. Στο VRS μοντέλο εισέρχεται μια νέα μεταβλητή  $u_0$  στην περίπτωση της Multiplier μορφής ή ένας νέος περιορισμός  $e\lambda = 1$  στην Envelopment. Όπως και στο CCR μοντέλο έτσι και εδώ έχουμε τον προσανατολισμό στις εισροές ή τις εκροές.

Στην περίπτωση του input orientation το μαθηματικό μοντέλο για τον υπολογισμό της αποδοτικότητας δίνεται παρακάτω με την Multiplier μορφή:

$\begin{aligned} & \max \theta = uy_j - u_0 \\ & \text{s.t.} \\ & vx_j = 1 \\ & uY - u_0 + vX_j \leq 0 \\ & v \geq 0 \text{ και } u \geq 0 \end{aligned}$	(1.6)
---	-------

Ενώ στην envelopment μορφή του δίνεται ως εξής:

$\begin{aligned} & \min \theta \\ & \text{s.t.} \\ & X\lambda - \theta x_j \leq 0 \\ & Y\lambda - y_j \geq 0 \\ & e\lambda = 1 \\ & \lambda \geq 0 \end{aligned}$	(1.7)
---	-------

Το BCC μοντέλο με output orientation στη multiplier μορφή εκφράζεται ως εξής:

$\min \frac{1}{E_j} = vX_j - u_0$ <p>s.t.</p> $uy_j = 1$ $uY_j - vX_j + u_0 \leq 0$ $v \geq 0 \text{ και } u \geq 0$	(1.8)
--	-------

Ενώ στην Envelopment μορφή του ως εξής:

$\max \theta$ <p>s.t.</p> $X\lambda - x_j \leq 0$ $Y\lambda - \theta y_j \geq 0$ $e\lambda = 1$ $\lambda \geq 0$	(1.9)
--	-------

## 2. Τεχνολογίες

### Γλώσσα Προγραμματισμού

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και συγκεκριμένα έκδοσης 3.9. Η Python είναι γνωστή ως μια από τις δημοφιλέστερες και ταχέως αναπτυσσόμενες γλώσσες προγραμματισμού. Ανήκει στις γλώσσες γενικού σκοπού και ορίζεται ως γλώσσα υψηλού επιπέδου, μεταφραζόμενη και αντικειμενοστραφής.

Η Python χρησιμοποιείται ευρέως για την κατασκευή ιστοσελίδων, αυτοματισμών και ανάλυσης δεδομένων. Επειδή ακριβώς είναι γενικού σκοπού χρησιμοποιείται σε πληθώρα προβλημάτων και όχι σε συγκεκριμένο είδος εφαρμογών. Επιπλέον χαρακτηρίζεται ως εύκολη στην εκμάθηση από αρχάριους προγραμματιστές, γεγονός που την καθιστά μια από τις πιο χρησιμοποιούμενες γλώσσες προγραμματισμού παγκοσμίως [1].

Σε Python κατασκευάζονται εφαρμογές που ανήκουν σε πολλούς διαφορετικούς τομείς, όπως μηχανικής μάθησης, αλγόριθμοι συστάσεων, tests λογισμικού, αλλά κυρίως διαδικτυακές εφαρμογές. Διαθέτει ιδιαίτερα πλούσιο οικοσύστημα βιβλιοθηκών όπως είναι το tensorflow, η numpy και scipy για οπτικοποίηση και ανάλυση δεδομένων, μηχανική μάθηση, ακόμα και επεξεργασία φυσικής γλώσσας (natural language processing).

## Framework

### Backend

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το Django web framework το οποίο αποτελεί ένα από τα δημοφιλέστερα frameworks ανάπτυξης διαδικτυακών εφαρμογών. Τα κύρια χαρακτηριστικά του Django είναι ότι προτρέπει τον προγραμματιστή να διατηρεί ένα καθαρό σχεδιαστικά κώδικα, είναι δωρεάν και ανοικτού κώδικα και έχει σχεδιαστεί για σύνδεση με βάση δεδομένων.

Επιπλέον το Django περιλαμβάνει όλα εκείνα τα στοιχεία και μηχανισμούς που χρειάζεται η πλειονότητα των διαδικτυακών εφαρμογών, ώστε ο προγραμματιστής να επικεντρωθεί στις κύριες λειτουργίες της εφαρμογής του. Αποτέλεσμα αυτού είναι ότι μηχανισμοί όπως η σύνδεση του χρήστη (login) στην εφαρμογή μας να έχουν δοκιμαστεί σε τέτοιο βαθμό ώστε να προσφέρουν εξαιρετική ασφάλεια και απροβλημάτιστη λειτουργία. Σαν ένα ολοκληρωμένο web framework, το Django ακολουθεί τις τελευταίες τάσεις στην ασφάλεια, αφού η ομάδα που το αναπτύσσει προσφέρει ενημερώσεις ασφαλείας και στις προηγούμενες εκδόσεις μέσω των LTS (long term support) εκδόσεων.

### Frontend

Για τη διεπαφή χρήστη, δηλαδή την εμφάνιση της εφαρμογής χρησιμοποιήθηκε το frontend framework Bootstrap έκδοσης 5. Η έκδοση 5 παρουσιάστηκε το 2020 και αποτελεί την εξέλιξη της έκδοσης 4. Το Bootstrap αποτελεί ένα ευρέως διαδεδομένο framework ανάπτυξης διεπαφών χρήστη και βασίζεται σε συλλογές component τα οποία χρησιμοποιούμε για να κατασκευάσουμε κάθε σελίδα. Η βασική διαφορά της έκδοσης 5 από την 4, είναι ότι δεν απαιτεί τη χρήση jQuery γεγονός το οποίο καθιστά το bootstrap ένα ιδιαίτερος ελαφρύ από άποψη πόρων framework.

Το Bootstrap εκτός από την ευκολία στη χρήση του και τη συμβατότητα όλων των components του με την συντριπτική πλειοψηφία των browsers, συνεργάζεται άψογα με το Django. Επιπλέον με τη χρήση ειδικού πακέτου μετατρέπουμε τις φόρμες εισαγωγής δεδομένων σε bootstrap 5 εμφάνιση χωρίς να χρειάζεται να γράψουμε τον HTML κώδικα για κάθε πεδίο ξεχωριστά.

### Βάση δεδομένων

Για την αποθήκευση των δεδομένων της εφαρμογής χρησιμοποιήθηκε η βάση δεδομένων PostgreSQL, η οποία αποτελεί ένα από τα πιο προηγμένα σχεσιακά συστήματα βάσεων δεδομένων. Ένας λόγος που επιλέγεται, είναι η ικανότητά της να χειριστεί πολύ μεγάλο όγκο δεδομένων σε μικρό χρονικό διάστημα, είναι δηλαδή ιδανική για εφαρμογές πραγματικού χρόνου που απαιτούν πολλές

εγγραφές και αναγνώσεις από τη βάση δεδομένων. Επιπλέον το Django συνεργάζεται εξαιρετικά με το συγκεκριμένο σύστημα βάσης δεδομένων καθώς υποστηρίζει τις περισσότερες λειτουργίες του. Ακόμα, το Django διαθέτει τύπους δεδομένων που υποστηρίζονται μόνο από την PostgreSQL.

### Βιβλιοθήκη **PuLP**

Η εφαρμογή μοντελοποιεί τα γραμμικά προβλήματα και τα επιλύει κάνοντας χρήση των λειτουργιών της βιβλιοθήκης PuLP. Η PuLP μπορεί να χαρακτηριστεί ως βιβλιοθήκη μοντελοποίησης γραμμικού προγραμματισμού και είναι κατασκευασμένη σε Python. Αφού μοντελοποιήσει το πρόβλημα, καλεί solvers για την επίλυση του γραμμικού προβλήματος. Η PuLP ως βιβλιοθήκη διαθέτει τον CBC solver, αλλά μπορεί να καλέσει και εξωτερικούς solvers όπως τον Gurobi και τον CPLEX. Ο CBC (Coin-or Branch and Cut) solver, όπως και η PuLP υποστηρίζονται από το ίδρυμα COIN-OR (4) και παρέχονται ως λογισμικό ανοικτού κώδικα.

Για τη μοντελοποίηση του προβλήματος χρησιμοποιούμε τις μεθόδους της PuLP οι κυριότερες από τις οποίες είναι:

Για την αρχικοποίηση του προβλήματος: `LpProblem("LP Problem name", LpMinimize)`

Η πρώτη παράμετρος δηλώνει το όνομα του προβλήματος, ενώ η δεύτερη δηλώνει τον τύπο, `LpMinimize` αν πρόκειται για πρόβλημα ελαχιστοποίησης ή `LpMaximize` αν πρόκειται για πρόβλημα μεγιστοποίησης.

Για τη δημιουργία των μεταβλητών: `LpVariable("variable_name", lowBound = 0)`

Η πρώτη παράμετρος δηλώνει το όνομα της μεταβλητής, η παράμετρος `lowBound` ή `upBound` τα κάτω και άνω όρια. Αν δεν εισάγουμε όρια τότε τα default όρια θεωρούνται τα  $-\infty$  και  $+\infty$ . Η μέθοδος προαιρετικά δέχεται την παράμετρο `cat` με τιμές `LpInteger` ή `LpContinuous` ή `LpBinary` ανάλογα με το είδος των τιμών που μπορεί να πάρει η μεταβλητή. Π.χ. `cat=LpInteger`

Για τη δημιουργία των περιορισμών: `LpConstraint(constraint_expression, sense=LpConstraintEQ, rhs=1, name="constraint name")`

Η πρώτη παράμετρος παρέχει τη μαθηματική έκφραση του περιορισμού, η παράμετρος `sense` δηλώνει το είδος του περιορισμού ( $\leq$ ,  $\geq$ ,  $=$ ), στην παράμετρο `rhs` παρέχεται το δεξί μέρος του περιορισμού και με την παράμετρο `name` μπορούμε δώσουμε όνομα στον περιορισμό.

Για την επίλυση του γραμμικού προβλήματος, η PuLP διαθέτει τη μέθοδο `solve()` στην οποία αν δεν εισάγουμε τον solver ως παράμετρο, θα χρησιμοποιηθεί ο default solver της PuLP. Επομένως αν θέλαμε να χρησιμοποιηθεί διαφορετικός solver, με προϋπόθεση ότι είναι εγκατεστημένος και διαθέσιμος στο σύστημα, θα έπρεπε να τον εισάγουμε στη `solve`, π.χ. `solve(CPLEX_CMD)`.

Με την `LpStatus` μπορούμε να λάβουμε πληροφορία για την κατάσταση του προβλήματος. Χαρακτηριστικά η τιμή 1 δηλώνει ότι βρέθηκε βέλτιστη λύση ενώ η τιμή -1 ότι το πρόβλημα είναι infeasible.

### Περιβάλλουσα Ανάλυση Δεδομένων με **PuLP**

Στο παράδειγμα που ακολουθεί θα περιγραφεί η διαδικασία που ακολουθήθηκε για την κατασκευή των γραμμικών προβλημάτων ώστε να εκτιμηθεί η αποδοτικότητα των παραγωγικών μονάδων. Για τους σκοπούς του παραδείγματος θα υποθέσουμε ότι στο εργοστάσιο FCTR λειτουργούν 4 μονάδες

παραγωγής (DMUs A, B, C, D, E, F, G) οι οποίες καταναλώνουν 2 εισροές (Input 1, Input 2) και παράγουν 2 εκροές (Output 1, Output 2). Τα δεδομένα που έχει στη διάθεσή του ο υπεύθυνος του εργοστασίου σχετικά με τις εισροές και εκροές των παραγωγικών μονάδων, παρουσιάζονται στον παρακάτω πίνακα.

DMUs	Input 1	Input 2	Output 1	Output 2
A	20	151	100	90
B	19	130	150	50
C	26	167	160	55
D	28	168	180	72
E	23	156	94	66
F	55	255	230	90
G	35	235	220	88

Πίνακας 1.1

Στο παράδειγμα θα χρησιμοποιηθεί το CCR μοντέλο (κλίμακα σταθερών αποδόσεων) με προσανατολισμό προς τις εκροές. Επομένως το μαθηματικό μοντέλο για την εκτίμηση της αποδοτικότητας της μονάδας A, στη Multiplier μορφή δίνεται από τη σχέση:

$$\begin{aligned}
 \min \frac{1}{E_A} &= \frac{20v_1 + 151v_2}{100u_1 + 90u_2} \\
 \text{s.t.} \\
 \frac{100u_1 + 90u_2}{20v_1 + 151v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα A,} \\
 \frac{150u_1 + 50u_2}{19v_1 + 130v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα B,} \\
 \frac{160u_1 + 55u_2}{26v_1 + 167v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα C,} \\
 \frac{180u_1 + 72u_2}{28v_1 + 168v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα D,} \\
 \frac{94u_1 + 66u_2}{23v_1 + 156v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα E,} \\
 \frac{230u_1 + 90u_2}{55v_1 + 255v_2} &\leq 1 \text{ Περιορισμός για τη μονάδα F,}
 \end{aligned}
 \tag{2.1}$$

$$\frac{220u_1 + 88u_2}{35v_1 + 235v_2} \leq 1 \text{ Περιορισμός για τη μονάδα G,}$$

$$u_1, u_2, v_1, v_2 \geq 0$$

Το παραπάνω πρόβλημα θα μετατραπεί σε γραμμικό θέτοντας τον περιορισμό:

$$\sum_{r=1}^s u_r y_{rj_0} = 1 \quad (2.2)$$

δηλαδή για τη μονάδα A:

$$100u_1 + 90u_2 = 1 \quad (2.3)$$

Επομένως, το γραμμικό πρόβλημα που θα πρέπει να λυθεί για την εκτίμηση της αποδοτικότητας της μονάδας A σε Output Oriented Multiplier μορφή του είναι:

$$\begin{aligned} & \min 20v_1 + 151v_2 \\ & \text{s.t} \\ & 100u_1 + 90u_2 = 1 \\ & 100u_1 + 90u_2 - 20v_1 - 151v_2 \leq 0 \\ & 150u_1 + 50u_2 - 19v_1 - 130v_2 \leq 0 \\ & 160u_1 + 55u_2 - 26v_1 - 167v_2 \leq 0 \\ & 180u_1 + 72u_2 - 28v_1 - 168v_2 \leq 0 \\ & 94u_1 + 66u_2 - 23v_1 - 156v_2 \leq 0 \\ & 230u_1 + 90u_2 - 55v_1 - 255v_2 \leq 0 \\ & 220u_1 + 88u_2 - 35v_1 - 235v_2 \leq 0 \\ & u_1, u_2, v_1, v_2 \geq 0 \end{aligned} \quad (2.4)$$



Έχοντας κατασκευάσει το γραμμικό πρόβλημα για τη μονάδα A θα το μοντελοποιήσουμε στην PuLP. Αρχικοποιούμε το πρόβλημα αναθέτοντάς το στη μεταβλητή `problem`, η οποία θα διαθέτει στη συνέχεια όλα τα δεδομένα του προβλήματος:

```
problem = LpProblem('dea_A', orientation)
```

Στη συνέχεια θα αρχικοποιήσουμε τις μεταβλητές του προβλήματος. Στην περίπτωση μας το πρόβλημα διαθέτει 4 μεταβλητές.

```
inp1 = LpVariable(name="input 1", lowBound=0)
inp2 = LpVariable(name="input 2", lowBound=0)
out1 = LpVariable(name="output 1", lowBound=0)
out2 = LpVariable(name="output 2", lowBound=0)
```

Εφόσον οι μεταβλητές έχουν αρχικοποιηθεί, μπορούμε να δημιουργήσουμε την αντικειμενική συνάρτηση και να την εισάγουμε στη μεταβλητή `problem` που όπως αναφέραμε διατηρεί όλα τα δεδομένα του προβλήματος.

```
obj = 20*inp1 + 151*inp2, 'objective function'
problem += obj
```

Αξίζει να σημειωθεί ότι στην PuLP μια linear expression χωρίς operator (`<=`, `==`, `>=`) όπως η  $20*inp1 + 151*inp2$ , μεταφράζεται σε `LpAffineExpression`, ενώ με operator σε `LpConstraint`

Στο επόμενο βήμα θα δημιουργήσουμε του περιορισμούς του προβλήματος. Ο περιορισμός ισότητας εισάγεται στη μεταβλητή `problem`.

```
problem += (100*out1 + 90*out2 == 1, 'equality constraint')
```

Δημιουργούμε τους περιορισμούς ανισότητας όπως φαίνεται παρακάτω

```
cA = 20*inp1 + 151*inp2 - 100*out1 - 90*out2 >= 0
cB = 19*inp1 + 130*inp2 - 150*out1 - 50*out2 >= 0
cC = 26*inp1 + 167*inp2 - 160*out1 - 55*out2 >= 0
```

```
cD = 28*inp1 + 168*inp2 - 180*out1 - 72*out2 >= 0
cE = 23*inp1 + 156*inp2 - 94*out1 - 66*out2 >= 0
cF = 55*inp1 + 255*inp2 - 230*out1 - 90*out2 >= 0
cG = 35*inp1 + 235*inp2 - 220*out1 - 88*out2 >= 0
```

Και τους εισάγουμε στη μεταβλητή problem

```
problem += cA
problem += cB
problem += cC
problem += cD
problem += cE
problem += cF
problem += cG
```

Θα μπορούσαμε να εισάγουμε κατευθείαν στο πρόβλημα τους περιορισμούς ανισότητας όπως συνέβη με την ισότητα, χωρίς να τους αναθέσουμε σε μεταβλητές. Στο παράδειγμα παρουσιάζεται η ανάθεση σε μεταβλητή διότι απαιτείται ώστε στη λύση του προβλήματος να μπορούμε να χρησιμοποιήσουμε τα λ του κάθε περιορισμού.

Τέλος επιλύεται το πρόβλημα με την

```
problem.solve()
```

#### Αποτελέσματα

Με τη λύση του προβλήματος, η problem πλέον διαθέτει όλα τα αποτελέσματα. Συγκεκριμένα, αν βρέθηκε βέλτιστη λύση, το problem.status θα επιστρέψει 1, ενώ η τιμή της αντικειμενικής συνάρτησης βρίσκεται στην problem.objective.value() όπου για την A είναι 1. Οι τιμές των μεταβλητών παρέχονται με την problem.variables() την οποία μπορούμε να διατρέξουμε με μια for loop όπως φαίνεται παρακάτω:

```
for var in problem.variables():
    print(str(var.name) + "=" + str(var.value()))
```

Οι περιορισμοί του προβλήματος είναι διαθέσιμοι στην `problem.constraints` την οποία μπορούμε να διατρέξουμε με μια `for loop` της μορφής:

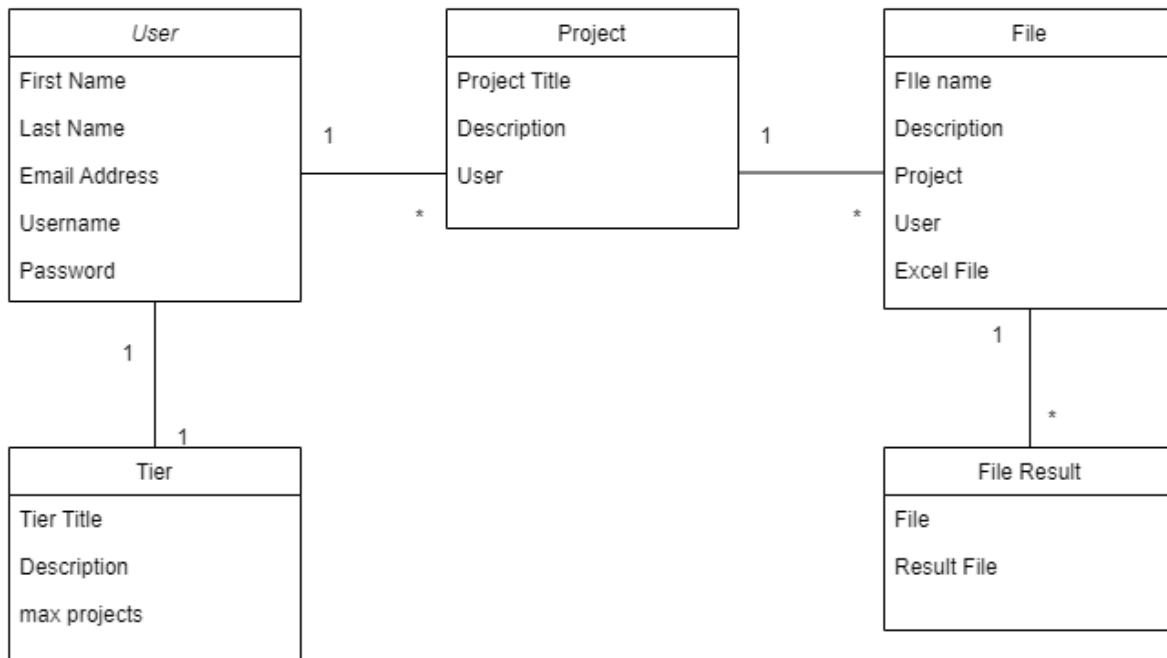
```
for name, constraint in problem.constraints.items():
    print(name, constraint)
```

### 3. Υλοποίηση

Διαγράμματα

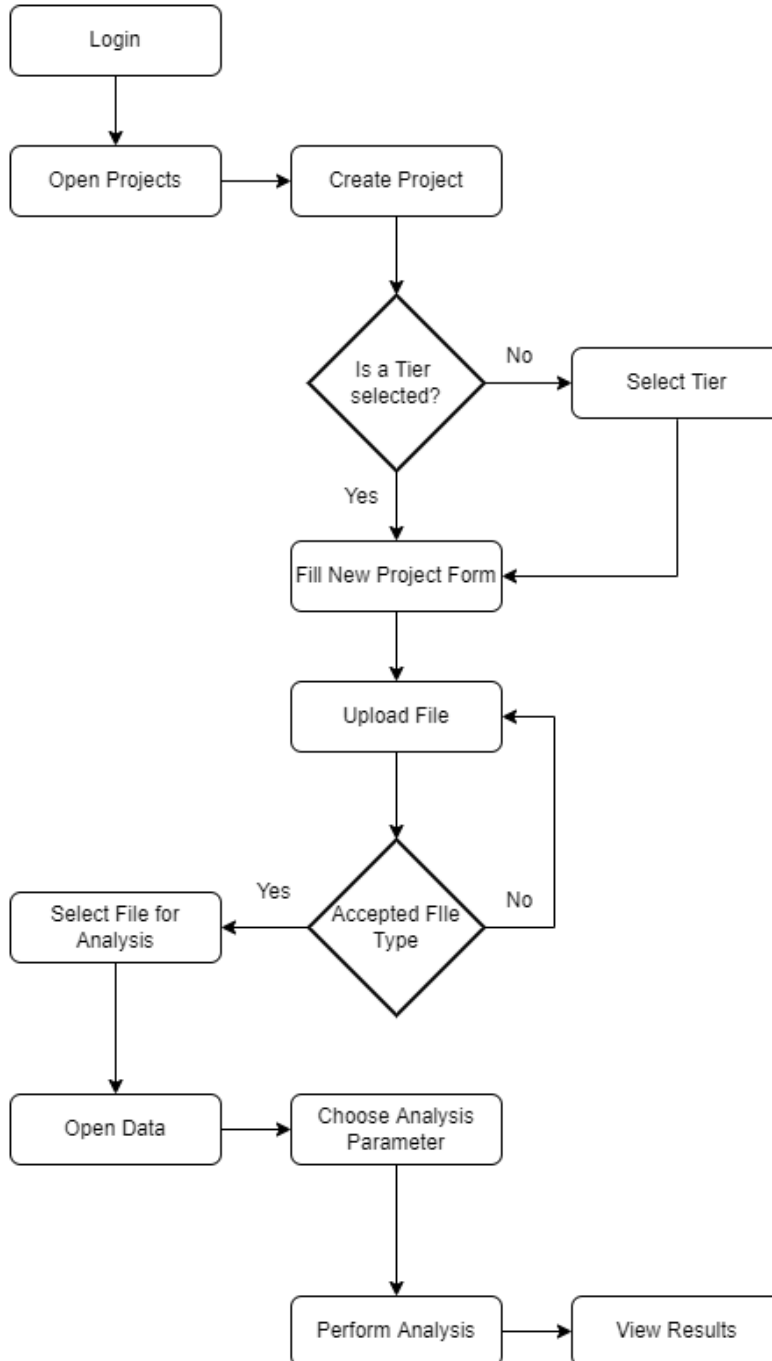
#### Class diagram

Στο class diagram παρακάτω παρουσιάζονται οι βασικές οντότητες, ενώ ακολουθεί σύντομη περιγραφή της εφαρμογής, ώστε να γίνει πιο κατανοητό. Η οντότητα `User` ανήκει σε κάποια βαθμίδα (`Tier`) η οποία καθορίζει το μέγιστο αριθμό `Projects` που μπορεί να δημιουργήσει. Ο χρήστης δημιουργεί `project` επομένως συνδέεται με την οντότητα `Project`. Στη συνέχεια προσθέτει το αρχείο που περιέχει τα δεδομένα που θα αναλυθούν το οποίο αναπαριστάται με την οντότητα `File` η οποία συνδέεται με την `Project` και την `User`. Τέλος η οντότητα `File Result` αναλαμβάνει την αποθήκευση των παραγόμενων αποτελεσμάτων και συνδέεται με τη `File` ώστε να γνωρίζουμε ποια αποτελέσματα ανήκουν σε κάθε αρχείο.



**Εικόνα 3.1 Class Διάγραμμα**

Διάγραμμα ροής



**Εικόνα 3.2 Διάγραμμα Ροής**

Στο παραπάνω διάγραμμα περιγράφονται οι βασικές ενέργειες ενός χρήστη από την πρώτη σύνδεση στο λογαριασμό του, μέχρι να λάβει τα αποτελέσματα μιας ανάλυσης. Δεν αναπαριστά κάθε οθόνη που θα συναντήσει, ούτε κάθε λειτουργία. Συγκεκριμένα έχουν παραληφθεί, η διαδικασία εγγραφής, οι οποιοσδήποτε στατικές σελίδες συμπεριλαμβανομένου βοηθητικών στοιχείων, καθώς και οι λειτουργίες ενημέρωσης και διαγραφής αντικειμένων Project και File.

**Βάση δεδομένων**

Στο παρακάτω διάγραμμα παρουσιάζονται όλες οι πίνακες της βάσης δεδομένων της εφαρμογής καθώς και οι σχέσεις μεταξύ τους. Το συγκεκριμένο διάγραμμα δημιουργήθηκε από το pgAdmin 4 μέσω της επιλογής Generate ERD (δεξί κλικ στη βάση για την οποία θέλουμε να κατασκευαστεί το entity relationship διάγραμμα).



## Αρχιτεκτονική **Django**

### Περιγραφή

Ένα Django project χρησιμοποιεί συγκεκριμένη δομή φακέλων και αρχείων, η οποία στοχεύει στην ελαχιστοποίηση συγγραφής ίδιου κώδικα. Η αρχική δομή ενός Django project αποτελείται από ένα φάκελο στον οποίο περιλαμβάνεται ολόκληρο το project και μέσα σε αυτό, υποφακέλους για κάθε Django App. Με προϋπόθεση ότι έχουμε εγκατεστημένη την Python και έχουμε ενεργοποιημένο python virtual environment, εγκαθιστούμε το django με την εντολή `pip install django`. Για να δημιουργήσουμε ένα νέο django project χρησιμοποιούμε την εντολή `django-admin startproject project_name` όπου `project_name`, το όνομα που δίνουμε στο project που μόλις δημιουργούμε. Με την εντολή αυτή δημιουργείται ο φάκελος `project_name` ο οποίος περιέχει όλα τα αρχεία που απαιτούνται για τη λειτουργία ενός django website και κυρίως το αρχείο `manage.py` με το οποίο στη συνέχεια χειριζόμαστε τα εργαλεία του django.

### Development Server

Το Django framework περιλαμβάνει server ανάπτυξης, επιταχύνοντας τις διαδικασίες που απαιτούνται για να ξεκινήσουμε να αναπτύσσουμε την εφαρμογή μας. Ο συγκεκριμένος server είναι κατασκευασμένος με Python επομένως δεν απαιτείται εγκατάσταση επιπλέον πακέτου στο μηχάνημα και δε χρειάζεται καμία ρύθμιση για να λειτουργήσει. Είναι όμως πολύ σημαντικό να θυμόμαστε ότι η χρήση του πρέπει να περιορίζεται αποκλειστικά σε περιβάλλοντα ανάπτυξης και όχι παραγωγικά. Για να εκκινήσουμε τον django development server χρησιμοποιούμε την εντολή `python manage.py runserver` και η εφαρμογή θα είναι προσβάσιμη στη διεύθυνση `http://127.0.0.1:8000/`

### Django Apps

Ένα Django project αποτελείται από ένα ή περισσότερα apps. Η βασική διαφορά του project από το app είναι ότι το app αποτελεί ένα τμήμα του συνολικού project το οποίο προσφέρει κάποιες λειτουργίες. Το project αποτελεί το σύνολο των ρυθμίσεων και των apps μιας διαδικτυακής εφαρμογής. Για να δημιουργήσουμε ένα django app χρησιμοποιούμε την εντολή `python manage.py startapp polls`. Στη δυνατότητα δημιουργίας apps βασίζεται η ιδιότητα του Django με την οποία προτρέπει την επαναχρησιμοποίηση κώδικα και όχι τη συγγραφή ομοίου κώδικα για κάποιο άλλο τμήμα της εφαρμογής. Μπορούμε δηλαδή να χρησιμοποιήσουμε τμήματα ενός app όπως μεθόδους, μοντέλα, validators κ.α. σε άλλο app του ίδιου django project. Μπορούμε ακόμα να αντιγράψουμε τον φάκελο ενός app σε ένα άλλο django project και με την προϋπόθεση ότι τηρούνται οι απαιτήσεις συμβατότητας της Python, του Django αλλά και όποιου άλλου python πακέτου χρησιμοποιεί, τότε θα αποκτήσουμε όλη τη λειτουργικότητα του app στο νέο μας project. Φυσικά, μπορούμε ακολουθώντας κατάλληλη διαδικασία να μετατρέψουμε το app σε δημόσια διαθέσιμο python πακέτο ώστε να χρησιμοποιηθεί και από άλλους προγραμματιστές [2].

## Models

Η δημιουργία των πινάκων (database tables) και των συσχετίσεων (relationships) στη βάση δεδομένων γίνεται με το Django ORM (object relational mapper). Μια απλουστευμένη εξήγηση του ρόλου του Django ORM είναι να μας παρέχει ένα μηχανισμό δημιουργίας, ανάκτησης, ενημέρωσης και διαγραφής αντικειμένων ανεξάρτητα από το είδος ή το σύστημα βάσης δεδομένων. Οι πίνακες αναπαριστώνται ως κλάσεις τις οποίες δημιουργούμε στο αρχείο Models.py. Για να δημιουργηθούν οι πίνακες στη βάση δεδομένων μέσω των κλάσεων που έχουμε δημιουργήσει, εκτελούμε την εντολή `python manage.py makemigrations`. Με την παραπάνω εντολή δημιουργούνται τα migrations αρχεία τα οποία αναπαριστούν όλες τις αλλαγές που έχουμε πραγματοποιηθεί στα μοντέλα σε σχέση με την προηγούμενη κατάστασή τους. Αφού δημιουργηθούν τα migration αρχεία, θα πρέπει να εκτελεστούν οι αλλαγές στη βάση δεδομένων. Αυτό γίνεται με την εντολή `python manage.py migrate`.

## Django admin site

Το Django framework περιέχει ένα εργαλείο διαχείρισης μέσω του οποίου μπορούμε να εκτελέσουμε CRUD operations, δηλαδή εργασίες δημιουργίας, ανάγνωσης ή ενημέρωσης και διαγραφής αντικειμένων. Οι λειτουργικότητα του admin περιβάλλοντος δημιουργείται βάσει των κλάσεων που έχουν δημιουργηθεί στο αρχείο Models.py. Κάθε αντικείμενο θα είναι προσβάσιμο στο διαχειριστικό περιβάλλον αρκεί να δηλωθεί η κλάση του στο αρχείο admin.py του εκάστοτε django app. Το διαχειριστικό περιβάλλον ενεργοποιείται αυτόματα κατά τη δημιουργία του νέου Django project (βλέπε [εντολή startproject](#)) και είναι προσβάσιμο στο path /admin/ π.χ. <http://127.0.0.1:8000/admin/>.

### Πρόσβαση στο **django admin**

Για να αποκτήσουμε πρόσβαση στο django admin θα πρέπει να δημιουργήσουμε τον αρχικό χρήστη διαχειριστή με την εντολή `django-admin createsuperuser`. Παρέχοντας τα στοιχεία σύνδεσης μέσω της κονσόλας, δημιουργείται ένας super user χρήστης ο οποίος έχει πλήρη δικαιώματα πρόσβασης. Μέσω του admin περιβάλλοντος μπορούμε να δημιουργήσουμε χρήστες με την προϋπόθεση ότι χρησιμοποιούμε το authentication system του django. Κάθε λογαριασμός χρήστη μπορεί να αποκτήσει πρόσβαση στο admin αν θέσουμε το πεδίο `is_staff` του λογαριασμού του σε True.

## Docker

Εκτός των εργαλείων ανάπτυξης της εφαρμογής, χρησιμοποιήθηκαν το Docker για τη λειτουργία της σε παραγωγικό περιβάλλον, αφού όπως αναφέρθηκε προηγουμένως ο development server του django είναι αποκλειστικά για σκοπούς ανάπτυξης. Το docker αποτελείται από ένα σύνολο εργαλείων με τα οποία μπορούμε να ομαδοποιήσουμε μια εφαρμογή μαζί με το λειτουργικό σύστημα καθώς και το σύνολο του λογισμικού που χρειάζεται σε ένα πακέτο, το λεγόμενο container.

Τα containers είναι απομονωμένα το ένα από το άλλο, εικονικά μηχανήματα, αλλά επειδή μοιράζονται δομικά στοιχεία του λειτουργικού του host μηχανήματος, χρησιμοποιούν λιγότερους πόρους από τις συνήθεις εικονικές μηχανές. Για να τρέξουν αυτά τα containers χρειάζεται να είναι εγκατεστημένο στο host μηχανήμα το docker engine. Το πλεονέκτημα των container σε σχέση με το να εγκαταστήσουμε την εφαρμογή μας και όλο το λογισμικό που χρειάζεται, κατευθείαν στο host



μηχάνημα, είναι ότι μέσω των container η εφαρμογή μπορεί να τρέξει σε οποιοδήποτε μηχάνημα χωρίς να χρειάζεται κάθε φορά να εγκαταστήσουμε ή να προβούμε σε ειδικές ρυθμίσεις που τυχόν απαιτούνται. Το Docker είναι διαθέσιμο για Linux, Windows και MacOS επομένως ένα container που τρέχει στον υπολογιστή μας θα μπορούσε να τρέξει σε οποιοδήποτε άλλο μηχάνημα / server ανεξαρτήτου λειτουργικού συστήματος, με μόνη προϋπόθεση να έχει εγκατεστημένο το docker engine και πρόσβαση στο διαδίκτυο.

## Docker Compose

Το docker compose είναι ένα εργαλείο με το οποίο μπορούμε να δομήσουμε και να τρέξουμε εφαρμογές που χρειάζονται περισσότερα από ένα containers, αλλά μέσα στον ίδιο host. Επίσης δίνει τη δυνατότητα να δηλώσουμε volumes των οποίων τα δεδομένα παραμένουν ακόμα και μετά την «καταστροφή» του container, όπως μιας βάσης δεδομένων ή ενός file storage.

Στην εφαρμογή που αναπτύχθηκε έχουν ενσωματωθεί τα εργαλεία Docker και μέσω του docker-compose δημιουργούνται 3 containers της εφαρμογής, της βάσης δεδομένων και του nginx. Ο nginx χρησιμοποιείται ως web server αντί του django development server αφού όπως αναφέρθηκε χρησιμοποιείται μόνο σε περιβάλλοντα ανάπτυξης. Επιπλέον δημιουργούνται volumes για τη βάση δεδομένων και τα αρχεία της εφαρμογής.

## Πακέτα - Βιβλιοθήκες

Σε αυτή την ενότητα θα αναφερθούν τα σημαντικότερα πακέτα (python packages) και εκδόσεις που χρησιμοποιήθηκαν για τις ανάγκες της εφαρμογής. Η εφαρμογή αναπτύχθηκε σε Python έκδοσης 3.9 και η εγκατάσταση των πακέτων έγινε με τον pip package manager έκδοσης 21.3.1, ενώ χρησιμοποιήθηκε το Django 4. Για την επικοινωνία με τη βάση δεδομένων PostgreSQL εγκαταστήσαμε το psycopg2-binary (pip install psycopg2-binary) έκδοσης 2.9.3. Για τη δημιουργία αρχείων PDF με τα αποτελέσματα της ανάλυσης χρησιμοποιήθηκε το εργαλείο Report Lab (pip install reportlab) έκδοσης 3.6.6.

## Βιβλιοθήκη PuLP

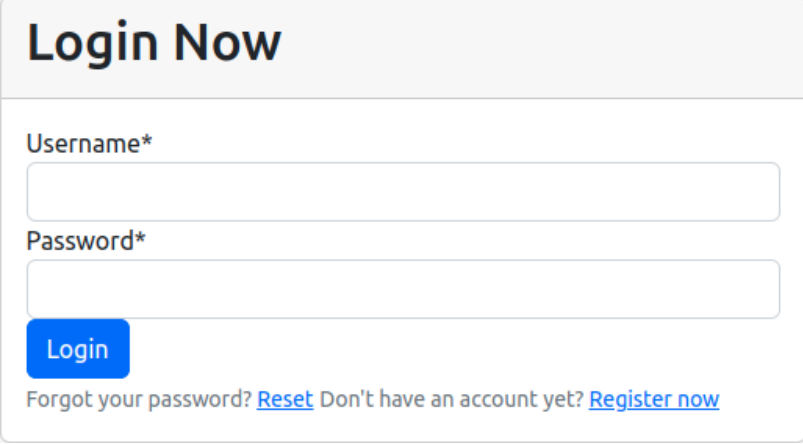
Η ανάλυση των δεδομένων που παρέχει η εφαρμογή βασίζεται στη βιβλιοθήκη PuLP, την οποία εγκαταστήσαμε με το pip install pulp. Η έκδοση της PuLP που εγκαταστάθηκε είναι η 2.7.0. Επιπλέον η PuLP απαιτεί Python έκδοσης 2.7 ή πιο πρόσφατη από την 3.4, ενώ στο μηχάνημα που αναπτύχθηκε η εφαρμογή χρησιμοποιήθηκε Python 3.9

Εφόσον έχουμε εγκαταστήσει τη βιβλιοθήκη στο python περιβάλλον, για να τη χρησιμοποιήσουμε στην εφαρμογή μας αρκεί να τη δηλώσουμε στα imports με την εντολή: `from pulp import *`.

## 4. Χρήση εφαρμογής

Σε αυτό το κεφάλαιο θα περιγραφεί ο τρόπος λειτουργίας της εφαρμογής που αναπτύχθηκε. Προκειμένου ο χρήστης να αποκτήσει πρόσβαση στις λειτουργίες της, θα πρέπει να συνδεθεί στο

λογαριασμό του. Οι επισκέπτες έχουν πρόσβαση μόνο στις δημόσιες σελίδες της εφαρμογής. Στην περίπτωση που είναι νέος χρήστης, παρέχεται δυνατότητα εγγραφής, μέσω της επιλογής Register.



**Login Now**

Username\*

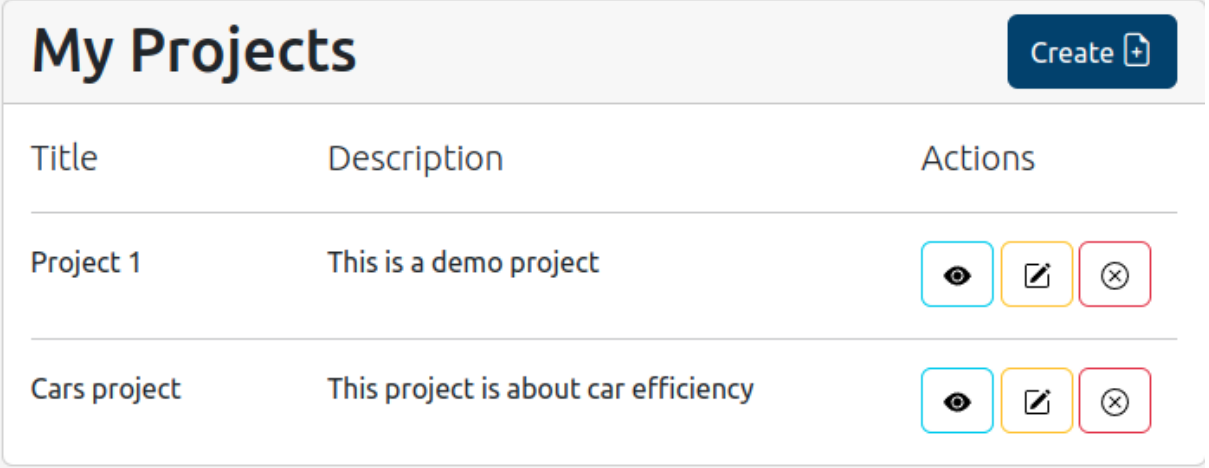
Password\*







Login

Forgot your password? [Reset](#) Don't have an account yet? [Register now](#)

Εικόνα 4.1 Φόρμα login

Εφόσον πλέον έχουμε συνδεθεί, μεταβαίνουμε στα project που έχουμε δημιουργήσει. Μέσω των project μπορούμε να ξεχωρίσουμε διαφορετικής θεματολογίας εργασίες. Αποτελούν το «χώρο» μέσα στον οποίο θα ανεβάσουμε αρχεία με δεδομένα για να αναλυθούν.



Title	Description	Actions
Project 1	This is a demo project	  
Cars project	This project is about car efficiency	  

Εικόνα 4.2 Λίστα project

Εδώ εμφανίζονται τα project μας προκειμένου να επιλέξουμε αυτό που μας ενδιαφέρει, ενώ όπως βλέπουμε υπάρχει η δυνατότητα δημιουργίας νέου project μέσω της επιλογής Create. Για κάθε project παρέχονται οι επιλογές, άνοιγμα, επεξεργασία και διαγραφή.

## New Project

Title\*

Project DEA

Description\*

Data envelopment analysis on bank branches

Create

### Εικόνα 4.3 Δημιουργία Project

Η δημιουργία νέου project ολοκληρώνεται συμπληρώνοντας τίτλο και περιγραφή για αυτό. Αφού δημιουργήθηκε, το επιλέγουμε για να εισέλθουμε σε αυτό.

## Project DEA

Return ↩

Description

Data envelopment analysis on bank branches

There are not any files associated with this project

Add File

### Εικόνα 4.4 Εμφάνιση νέου project

Επειδή πρόκειται για νέο project του οποίου ο «χώρος» είναι κενός, μας ενημερώνει για το ότι δεν έχουν υπάρχουν αρχεία, επομένως δεν μπορούμε να επιλέξουμε κάποιο για ανάλυση. Προκειμένου να αναλυθούν τα δεδομένα, η εφαρμογή τα δέχεται με κατάλληλα διαμορφωμένο αρχείο excel, το οποίο θα μεταφορτώσουμε μέσω της επιλογής Add File.

### Add new file to Project

Name\*

Description\*

File\*

Choose File banks\_data\_1.xlsx

Add file

#### Εικόνα 4.5 Προσθήκη αρχείου

Καλούμαστε να δώσουμε τίτλο και περιγραφή για το αρχείο, ώστε να το αναγνωρίζουμε εντός της εφαρμογής και φυσικά το προς μεταφόρτωση αρχείο.

<div data-bbox="193 994 794 1458"> <h3>Project DEA</h3> <p>Return</p> <p>Description</p> <p>Data envelopment analysis on bank branches</p> <p><b>Banks_1</b></p> <p>Excel file with the bank branches data.</p> <p>Add File</p> </div>	<div data-bbox="794 994 1394 1458"> <h3>File Banks_1</h3> <p>Return</p> <p>Click to download file: <a href="#">Banks_1</a></p> <p>Description: Excel file with the bank branches data.</p> <p>Open Data</p> </div>
<p><b>Εικόνα 4.6.1 Μεταφόρτωση αρχείου</b></p>	<p><b>Εικόνα 4.6.2 Άνοιγμα αρχείου</b></p>

Όπως και για τα project, έτσι και για τα αρχεία παρέχονται οι επιλογές, άνοιγμα, επεξεργασία και διαγραφή. Στη συνέχεια επιλέγουμε το αρχείο, τα δεδομένα του οποίου θα αναλυθούν. Στην εικόνα 4.6.2 βλέπουμε την επιλογή για λήψη του αρχείου και την επιλογή Open Data, μέσω της οποία μεταφερόμαστε στην οθόνη εισαγωγής των παραμέτρων που αφορούν τη δημιουργία και επίλυση των γραμμικών προβλημάτων.

Το προς ανάλυση αρχείο πρέπει να είναι τύπου excel (.xlsx) και να έχει στην πρώτη στήλη τα ονόματα των DMUs. Επίσης στην πρώτη γραμμή να περιέχονται τα ονόματα των εισροών και εκροών.

Δεν πρέπει να υπάρχουν κενές γραμμές ή κενές στήλες. Παρακάτω παρουσιάζεται η τυπική μορφή που πρέπει να έχει το αρχείο.

DMUs	Input 1	Input 2	Output 1	Output 2
DMU_1	$x_{11}$	$x_{12}$	$y_{11}$	$y_{12}$
DMU_2	$x_{21}$	$x_{22}$	$y_{21}$	$y_{22}$
DMU_3	$x_{31}$	$x_{32}$	$y_{31}$	$y_{32}$
DMU_4	$x_{41}$	$x_{42}$	$y_{41}$	$y_{42}$

Πίνακας 4.1 Δομή αποδεκτού αρχείου

Μέσω της παρακάτω οθόνης (εικόνα 4.7) καλούμαστε να επιλέξουμε ποιες στήλες αποτελούν input και ποιες output, το είδος της ανάλυσης, δηλαδή αν θα είναι στραμμένη στις εισροές ή τις εκροές, και την κλίμακα αποδόσεων.

## File : Banks\_1

### Optimization options

**Column staff is**  
 Input  Output

**Column accounts is**  
 Input  Output

**Column Personal transactions is**  
 Input  Output

**Column business transactions is**  
 Input  Output

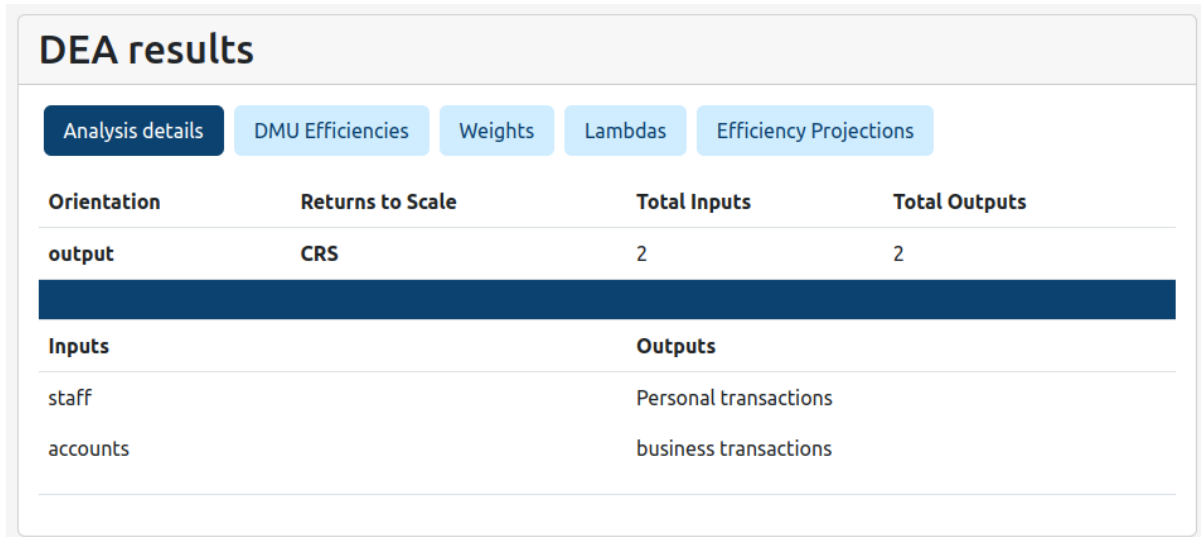
Select orientation

Select returns to scale

In case one or more of your DMUs are not recognised, please ensure that your excel file is structured according to this [example file](#).

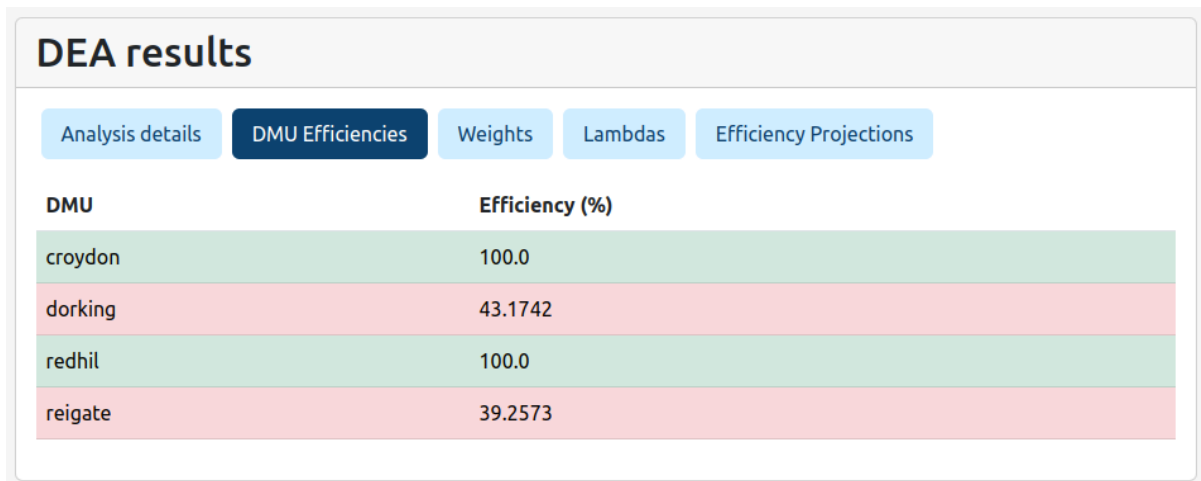
### Εικόνα 4.7 Επιλογή εισρών-εκρών και μεθόδου ανάλυσης

Εφόσον έχουμε κάνει τις κατάλληλες ρυθμίσεις σύμφωνα με τους στόχους της ανάλυσης, επιλέγουμε solve, ώστε να κατασκευαστεί και να λυθεί το γραμμικό πρόβλημα για κάθε DMU, δηλαδή για κάθε row του excel που περιέχει τα δεδομένα. Η ανάλυση ολοκληρώνεται με την εμφάνιση της οθόνης αποτελεσμάτων.



### Εικόνα 4.8.1 Εμφάνιση αποτελεσμάτων

Στην επιλογή "DMU Efficiencies" παρουσιάζονται οι αποδοτικότητες των μονάδων σε ποσοστό % και με χρωματικό χαρακτηρισμό



### Εικόνα 4.8.2 Εμφάνιση αποδοτικών μονάδων

Στην επιλογή “Weights” εμφανίζονται οι τιμές των μεταβλητών κάθε μονάδας

Analysis details				DMU Efficiencies				Weights				Lambdas				Efficiency Projections			
<b>DMU</b>				<b>Efficiency</b>															
croydon				100.0															
<b>Personal_transactions</b>				<b>accounts</b>				<b>business_transactions</b>				<b>staff</b>							
0.00800000				0.00105263				0.00000000				0.00000000							
<b>DMU</b>				<b>Efficiency</b>															
dorking				43.1742															
<b>Personal_transactions</b>				<b>accounts</b>				<b>business_transactions</b>				<b>staff</b>							
0.00704935				0.00000000				0.03449144				0.14476334							
<b>DMU</b>				<b>Efficiency</b>															
redhil				100.0															
<b>Personal_transactions</b>				<b>accounts</b>				<b>business_transactions</b>				<b>staff</b>							
0.00000000				0.00138889				0.01818182				0.00000000							
<b>DMU</b>				<b>Efficiency</b>															
reigate				39.2573															
<b>Personal_transactions</b>				<b>accounts</b>				<b>business_transactions</b>				<b>staff</b>							
0.02999539				0.00530688				0.02584218				0.00000000							

Εικόνα 4.8.4 Συντελεστές βαρύτητας (weights)

Επιπλέον στην επιλογή Lambdas εμφανίζονται οι τιμές του  $\theta$  και των  $\lambda$  για κάθε μονάδα

### DEA results

Analysis details DMU Efficiencies Weights **Lambdas** Efficiency Projections

DMU	$\theta$	croydon	dorking	redhil	reigate
croydon	1.0	1.0	0.0	0.0	0.0
dorking	2.3162	0.6606	0.0	0.2417	0.0
redhil	1.0	0.0	0.0	1.0	0.0
reigate	2.5473	0.2702	0.0	0.3101	0.0

Εικόνα 4.8.5 Τιμές  $\lambda$  και  $\theta$  κάθε μονάδας

Τέλος στην επιλογή “Efficiency Projections” υπολογίζονται οι τιμές των μεταβλητών ώστε μια μη αποδοτική μονάδα να προβληθεί στο σύνορο αποδοτικότητας.

### DEA results

Analysis details DMU Efficiencies Weights Lambdas **Efficiency Projections**

only inefficient units appear

DMU	staff	accounts	Personal transactions	business transactions
dorking	16.0	801.6113	101.9134	46.3243
reigate	10.136	480.0	58.5879	30.5676

Εικόνα 4.8.6 Προβολές αποδοτικότητας

Τα δεδομένα κάθε ανάλυσης έχουν εγγραφεί σε αρχείο PDF το οποίο είναι διαθέσιμο οποιαδήποτε στιγμή για προβολή ή λήψη χωρίς να χρειάζεται να τρέξουμε εκ νέου ανάλυση. Στο περιεχόμενου του PDF εμφανίζονται τα αποτελέσματα κάθε μονάδας.



## Results for Banks\_1

20/11/2022

Orientation: Output

Returns to Scale: Constant

### Inputs

staff  
accounts

### Outputs

Personal transactions  
business transactions

## Analysis results

### Objective function values and efficiency ratios %

For DMU croydon: Obj. Func. = 1.0 | Efficiency = 100.0 %  
For DMU dorking: Obj. Func. = 2.3162 | Efficiency = 43.1742 %  
For DMU redhil: Obj. Func. = 1.0 | Efficiency = 100.0 %  
For DMU reigate: Obj. Func. = 2.5473 | Efficiency = 39.2573 %

### Efficiency Projections

#### *For DMU dorking*

staff = 15.999999939999999  
accounts = 801.6112763  
Personal transactions = 101.91339345  
business transactions = 46.3242697

#### *For DMU reigate*

staff = 10.13604068  
accounts = 480.00000329999995  
Personal transactions = 58.58790995  
business transactions = 30.567605199999996

### Εικόνα 4.9 Προβολή PDF με τα αποτελέσματα της ανάλυσης

Τα αρχεία PDF αποθηκεύονται εντός της εφαρμογής για μελλοντική χρήση και έχουμε τη δυνατότητα να τα κατεβάσουμε οποιαδήποτε στιγμή.

**File Banks\_1** Return ↶



Click to download file: [Banks\\_1](#)

Description: Excel file with the bank branches data.

**Results**



---

Nov. 12, 2022,  
10:42 p.m.

---

Nov. 12, 2022,  
10:43 p.m.

Open Data

**Εικόνα 4.10** Λίστα αποτελεσμάτων ανάλυσης

### Μελέτη Περίπτωσης

Σε αυτή την ενότητα θα μελετηθεί η περίπτωση μιας τράπεζας (Chu & Zhu, 2021) η οποία επιθυμεί να μετρήσει την αποδοτικότητα των υποκαταστημάτων της. Πρόκειται για την Bank of Changsha στην Κίνα όπου η διεύθυνσή της διαθέτει δεδομένα για τα 31 υποκαταστήματά της σχετικά με τον αριθμό των υπαλλήλων, τα περιουσιακά στοιχεία, το κόστος λειτουργίας τους, τα έσοδα από τόκους, τα γενικά έσοδα και τις επισφαλείς οφειλές.

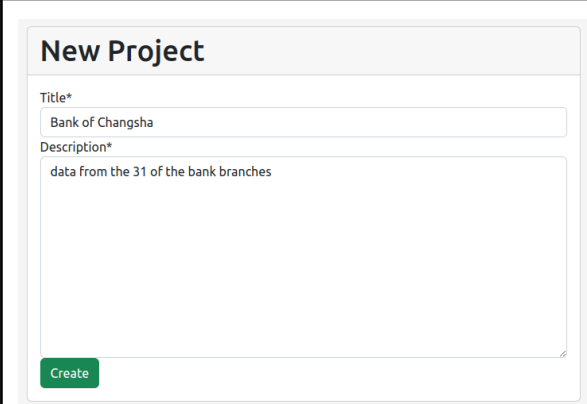
Ως εισροές κάθε υποκαταστήματος χαρακτηρίζονται οι υπάλληλοι, τα περιουσιακά στοιχεία και το κόστος λειτουργίας του. Ως εκροές θεωρούνται τα έσοδα από τόκους, τα γενικά έσοδα και οι επισφαλείς οφειλές. Οι υπάλληλοι αναγράφονται σε αριθμό προσωπικού, ενώ τα ποσά σε Yuan Κίνας. Στρέφουμε το ενδιαφέρον στα εισοδήματα των υποκαταστημάτων δηλαδή τη μεγιστοποίηση των εκροών τους διατηρώντας το επίπεδο των εισροών τους, επιλέγουμε προσανατολισμό στις εκροές (output orientation). Το σύνολο των δεδομένων παρουσιάζονται στον πίνακα 4.1.

DMUs	Employees (input_x1)	Assets (input_x2)	Operations (input_x3)	Interest income (output_y1)	Non-Interes t income (output_y2)	Bad debts (output_y3)
DMU_1	254	20741668	3532	42416	18867	16583
DMU_2	266	1238002	4524	35921	16123	32704
DMU_3	144	713075	906	9720	4537	7104
DMU_4	220	586641	4246	40832	23636	31463
DMU_5	189	777671	1824	19839	9355	7616
DMU_6	134	532863	2852	11103	4866	4553
DMU_7	178	541977	1613	12883	4791	5591
DMU_8	169	485534	1512	15878	5175	6152
DMU_9	153	566095	1424	16539	4116	10635
DMU_10	84	403103	2407	10145	3132	3763
DMU_11	81	269142	4530	16806	3524	11270
DMU_12	78	391531	984	6409	2088	2890
DMU_13	134	1602474	2940	6335	3550	5943
DMU_14	85	2019583	3058	18605	3908	7194
DMU_15	137	1065375	3153	11426	3356	6251
DMU_16	188	2218811	2995	16946	3516	6742
DMU_17	143	1362668	2136	13178	5757	6340
DMU_18	175	2474926	5217	20642	6175	9304
DMU_19	171	1170429	1596	13369	3337	3341
DMU_20	226	1409712	3416	29732	16869	13838
DMU_21	182	1559882	2256	20741	6313	11236
DMU_22	177	1332425	4397	16337	5932	10327
DMU_23	165	1461203	3083	22817	10485	7150

DMU_24	221	1394571	4625	30052	9061	13703
DMU_25	223	1453229	3397	29611	16771	13986
DMU_26	76	665045	1078	5935	3546	1499
DMU_27	136	838799	929	9911	4493	6901
DMU_28	138	791714	3106	11139	3397	6334
DMU_29	216	2344464	1921	16985	5951	9011
DMU_30	192	2008119	1842	19625	9349	7786
DMU_31	193	322101	5284	28706	13233	15744

**Πίνακας 4.2: Δεδομένα υποκαταστημάτων**

Έχοντας τα δεδομένα σε αρχείο excel με την παραπάνω δομή, θα το μεταφορτώσουμε στην εφαρμογή. Παρατηρούμε ότι στην πρώτη στήλη περιέχονται τα ονόματα των DMU δηλαδή των υποκαταστημάτων, ενώ στην πρώτη γραμμή τα ονόματα των μεταβλητών. Αρχικά δημιουργούμε νέο project στην εφαρμογή το επιλέγουμε και μεταφορτώνουμε το αρχείο excel.



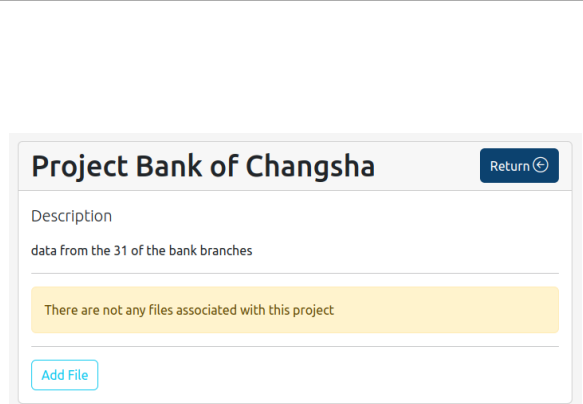
**New Project**

Title\*  
Bank of Changsha

Description\*  
data from the 31 of the bank branches

Create

**Εικόνα 4.11 Δημιουργία Project**



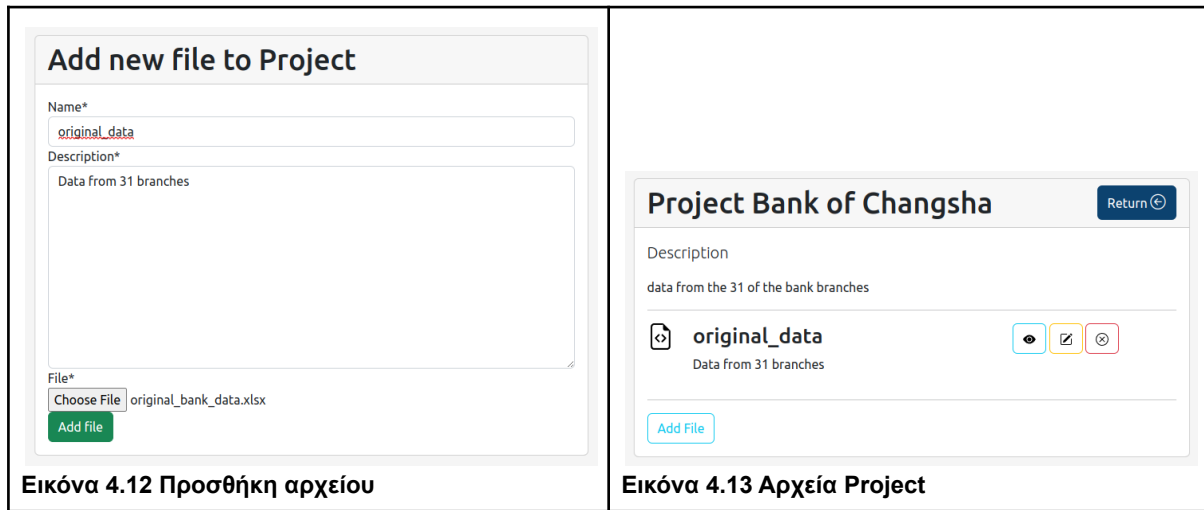
**Project Bank of Changsha** Return

Description  
data from the 31 of the bank branches

There are not any files associated with this project

Add File

**Εικόνα 4.12 Επιλογή Project**



Στη συνέχεια επιλέγουμε το αρχείο που μεταφορτώσαμε για να εισάγουμε τις ρυθμίσεις της ανάλυσης. Στην περίπτωση που μελετάμε όπως αναφέρθηκε παραπάνω, η ανάλυση έχει προσανατολισμός στις εκροές.

### **CRS** Μοντέλο

Αρχικά θα επιλέξουμε την κλίμακα σταθερών αποδόσεων (Constant Returns to Scale-CRS), επομένως εισάγουμε τις κατάλληλες επιλογές όπως φαίνεται στην εικόνα 4.14.

## File : original\_data

### Optimization options

---

**Column input\_x1 is**  
 Input    Output

**Column input\_x2 is**  
 Input    Output

**Column input\_x3 is**  
 Input    Output

**Column output\_y1 is**  
 Input    Output

**Column output\_y2 is**  
 Input    Output

**Column output\_y3 is**  
 Input    Output

In case one or more of your DMUs are not recognised, please ensure that your excel file is structured according to this [example file](#).

---

Select orientation

Output Orientation

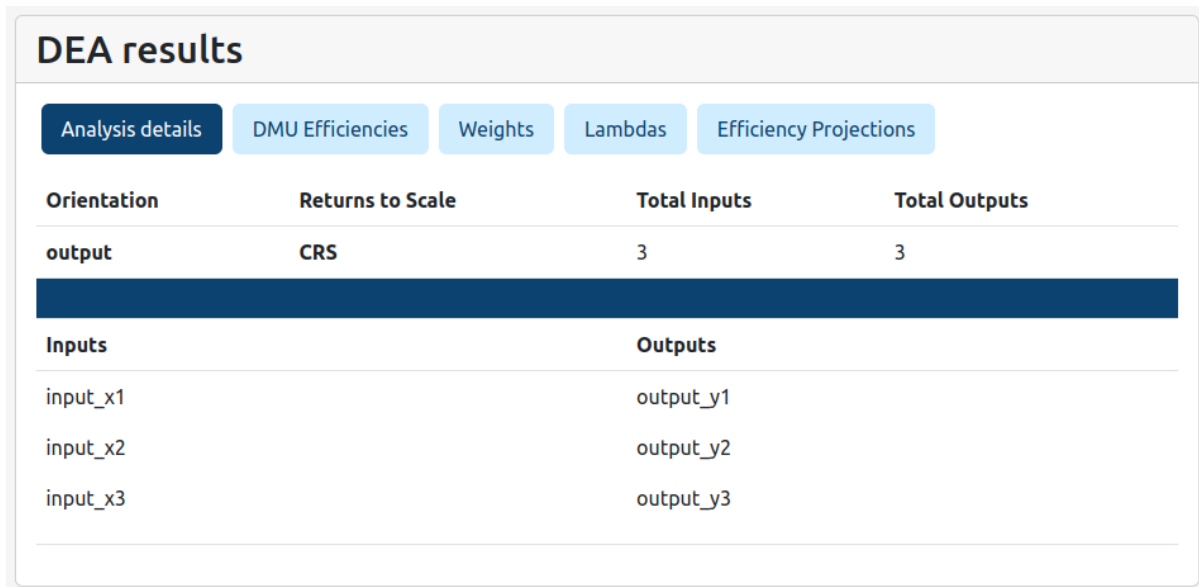
Select returns to scale

CRS

Solve

**Εικόνα 4.14:** Εισαγωγή ρυθμίσεων ανάλυσης

Έχοντας επιλέξει τις στήλες που αποτελούν εισροές και εκροές καθώς και το είδος του προβλήματος επιλέγουμε Solve.



**Εικόνα 4.15: Λεπτομέρειες ανάλυσης**

Τα γραμμικά προβλήματα λύθηκαν επιτυχώς και λαμβάνουμε τα αποτελέσματα. Παρατηρούμε ότι με το CRS output oriented μοντέλο έχουμε 8 αποδοτικές μονάδες και 23 μη αποδοτικές.

DMU	Efficiency Ratios
DMU_1	100
DMU_2	97.1912
DMU_3	100
DMU_4	100
DMU_5	100
DMU_6	44.0238
DMU_7	73.0353
DMU_8	95.2472
DMU_9	100
DMU_10	62.4493
DMU_11	100

Μεταπτυχιακή Διατριβή	Βλάχος Ηλίας - Σπυρίδων
-----------------------	-------------------------

DMU_12	61.207
DMU_13	31.0116
DMU_14	100
DMU_15	43.2096
DMU_16	55.6174
DMU_17	59.9736
DMU_18	57.6701
DMU_19	71.9735
DMU_20	89.1027
DMU_21	85.3097
DMU_22	47.515
DMU_23	75.7117
DMU_24	72.0046
DMU_25	89.1266
DMU_26	59.0912
DMU_27	98.7362
DMU_28	42.2833
DMU_29	77.4293
DMU_30	97.6944
DMU_31	100

**Πίνακας 4.3: CRS ποσοστά αποδοτικότητας**

Οι αποδοτικότητες στον πίνακα 4.3 υπολογίζονται σε ποσοστό και παρατηρούμε ότι από τις 23 μη αποδοτικές μονάδες μόνο 4 παρουσιάζουν αποδοτικότητα άνω του 90%, με μέσο όρο  $avg_{eff} \approx 77\%$

Στον πίνακα 4.4 παρουσιάζονται οι προβολές αποδοτικότητας για κάθε μη αποδοτική μονάδα, σύμφωνα με τις οποίες ο μέσος όρος ποσοστού βελτίωσης των εκροών είναι: για την  $y_1 \approx 63.7\%$ , για την  $y_2 \approx 109.7\%$  και για την  $y_3 = 140\%$



Μεταπτυχιακή Διατριβή	Βλάχος Ηλίας - Σπυρίδων
-----------------------	-------------------------

DMUs	Interest income (output_y1)	Non-Interest income (output_y2)	Bad debts (output_y3)
DMU_2	43833.3505	25018.6806	33650.1156
DMU_6	25220.7169	13666.8233	18670.7302
DMU_7	17639.0579	6559.7087	10887.8147
DMU_8	16671.0715	5562.5316	11266.2188
DMU_10	16244.9343	7396.294	11480.6435
DMU_12	10471.3301	4195.0035	7256.9432
DMU_13	24870.4	14396.4727	19163.8273
DMU_15	26442.8461	12843.2688	17811.3919
DMU_16	30469.5707	15310.6455	21426.1611
DMU_17	21973.5144	10455.8573	15417.0273
DMU_18	35794.1392	12517.3993	19454.8337
DMU_19	18575.3157	4853.1194	11648.0768
DMU_20	33367.5814	18931.7144	24525.9479
DMU_21	24313.067	9550.4532	16327.1471
DMU_22	34382.4325	15992.7761	22914.8513
DMU_23	30136.139	17041.9538	22360.3851
DMU_24	41735.4775	22417.7888	30346.833
DMU_25	33223.3378	18816.9464	24311.0697
DMU_26	10366.6736	6000.8497	7988.0155
DMU_27	10038.1203	4550.628	6989.5135
DMU_28	26343.5096	13406.7773	18557.0094
DMU_29	21936.4134	7685.8166	11637.8582
DMU_30	20088.9133	9570	8351.7636

Πίνακας 4.4 CRS Προβολές αποδοτικότητας

Για παράδειγμα η DMU 2 για να θεωρηθεί αποδοτική θα πρέπει να αυξήσει την εκροή  $y_1$  κατά 22%, την  $y_2$  κατά 55% και την  $y_3$  κατά 2.9%

### VRS Μοντέλο

Στη συνέχεια θα αναλύσουμε τα ίδια δεδομένα αλλά με την κλίμακα μεταβλητών αποδόσεων.

## File : original\_data

### Optimization options

**Column input\_x1 is**  
 Input  Output

**Column input\_x2 is**  
 Input  Output

**Column input\_x3 is**  
 Input  Output

**Column output\_y1 is**  
 Input  Output

**Column output\_y2 is**  
 Input  Output

**Column output\_y3 is**  
 Input  Output

In case one or more of your DMUs are not recognised, please ensure that your excel file is structured according to this [example file](#).

Select orientation:

Select returns to scale:

**Εικόνα 4.18** Εισαγωγή ρυθμίσεων ανάλυσης

Στην περίπτωση μεταβλητών αποδόσεων κλίμακας (VRS) παρατηρούμε πλέον 13 αποδοτικές μονάδες και 18 μη αποδοτικές, όπως βλέπουμε στον πίνακα 4.5

DMU	Efficiency Ratios
-----	-------------------

Μεταπτυχιακή Διατριβή	Βλάχος Ηλίας - Σπυρίδων
-----------------------	-------------------------

DMU_1	100
DMU_2	100
DMU_3	100
DMU_4	100
DMU_5	100
DMU_6	50.782
DMU_7	75.9763
DMU_8	100
DMU_9	100
DMU_10	88.4877
DMU_11	100
DMU_12	100
DMU_13	37.9607
DMU_14	100
DMU_15	45.9179
DMU_16	56.373
DMU_17	65.4236
DMU_18	61.7589
DMU_19	73.2493
DMU_20	89.3655
DMU_21	86.1326
DMU_22	48.4027
DMU_23	81.1622
DMU_24	73.5132
DMU_25	89.3815

Μεταπτυχιακή Διατριβή	Βλάχος Ηλίας - Σπυρίδων
-----------------------	-------------------------

DMU_26	100
DMU_27	100
DMU_28	46.2577
DMU_29	79.0077
DMU_30	97.924
DMU_31	100

Πίνακας 4.5: VRS ποσοστά αποδοτικότητας

Ο μέσος όρος της αποδοτικότητας των μονάδων είναι 82%.

Στον παρακάτω πίνακα παρουσιάζονται οι προβολές αποδοτικότητας των μη αποδοτικών μονάδων.

DMUs	Interest income (output_y1)	Non-Interest income (output_y2)	Bad debts (output_y3)
DMU_6	21864.0699	10760.8995	15388.7937
DMU_7	16956.1214	6305.7345	9004.405
DMU_10	11464.5719	3539.3829	6821.2515
DMU_13	21846.3592	10774.6667	15655.7058
DMU_15	24883.6162	11350.9718	16071.5735
DMU_16	30061.136	15098.8032	20987.0999
DMU_17	20142.2824	9506.0545	13448.4808
DMU_18	33423.0001	17060.0001	23373.3334
DMU_19	18250.9018	5310.8171	11616.9743
DMU_20	33269.0388	18875.8044	25211.8377
DMU_21	24079.961	9879.3296	16304.7985
DMU_22	33752.2889	17352.2666	23732.874
DMU_23	28112.9838	14951.8087	19760.3444

Μεταπτυχιακή Διατριβή	Βλάχος Ηλίας - Σπυρίδων
-----------------------	-------------------------

DMU_24	40878.5884	23495.7354	31025.3531
DMU_25	33129.8669	18764.0065	25028.8534
DMU_28	24079.8908	11447.1123	16343.773
DMU_29	21497.9181	7568.7362	13457.1353
DMU_30	20041.4166	9547.3734	9039.5373

**Πίνακας 4.6 VRS Προβολές αποδοτικότητας**

Με τις προβολές αποδοτικότητας στη VRS ανάλυση, ο μέσος όρος ποσοστού βελτίωσης των εκροών είναι  $y_1 \approx 60.2\%$ , για την  $y_2 \approx 109.9\%$  και για την  $y_3 = 127\%$

Για παράδειγμα η DMU 7 για να γίνει αποδοτική θα πρέπει να αυξήσει την εκροή  $y_1$  κατά 31.6%, την  $y_2$  κατά 31.6% και την  $y_3$  κατά 61%.

Τέλος, στον πίνακα 4.7 παρουσιάζονται συγκεντρωτικά οι παρατηρήσεις βάσει των αποτελεσμάτων που προέκυψαν από τα μοντέλα CRS και VRS με προσανατολισμό στις εκροές (**Output Orientation**).

	<b>CRS Output Oriented</b>	<b>VRS Output Oriented</b>
Αποδοτικές μονάδες	8	13
Μη αποδοτικές μονάδες	23	18
Μέσος όρος efficiency ratio	77%	82%
Ελάχιστο efficiency ratio	31%	38%
Μ.Ο. Ποσοστού βελτίωσης $y_1$	63.7%	60.2%
Μ.Ο. Ποσοστού βελτίωσης $y_2$	109.7%	109.9%
Μ.Ο. Ποσοστού βελτίωσης $y_3$	140%	127%

**Πίνακας 4.7 Συγκεντρωτικά αποτελέσματα από τα μοντέλα CRS και VRS με προσανατολισμό στις εκροές (Output Orientation)**

## Συμπεράσματα

Λαμβάνοντας υπόψη την ευρύτερη κατάσταση που επικρατεί στις αρχές της τρίτης δεκαετίας του 21ου αιώνα, κρίνεται αναγκαία, ίσως περισσότερο από ποτέ, η μέτρηση και βελτίωση της αποδοτικότητας των οργανισμών όχι μόνο για την ισχυροποίηση της θέσης τους σε σχέση με τον ανταγωνισμό, αλλά πολλές φορές για την επιβίωσή τους. Οι οργανισμοί θα πρέπει να εντάξουν τη μέτρηση της αποδοτικότητας στα σπάνια στοιχεία μέτρησης και η ΠΑΔ αποτελεί το κατάλληλο εργαλείο για το σκοπό αυτό. Η ικανότητα της ΠΑΔ να διαχειριστεί πολλαπλές εισροές και εκροές την καθιστά

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΠΕΡΙΒΑΛΛΟΥΣΑΣ ΑΝΑΛΥΣΗΣ ΔΕΔΟΜΕΝΩΝ (ΠΑΔ)	44
--	----

αξιόπιστη και πολύτιμη τεχνική στην προσπάθεια των αναλυτών εξαγωγής δεδομένων τα οποία θα υποστηρίξουν τους αρμόδιους για τη λήψη αποφάσεων.

Η παρούσα μεταπτυχιακή διατριβή στοχεύει στην ανάδειξη της περιβάλλουσας ανάλυσης δεδομένων ως κατάλληλη τεχνική για την εκτίμηση της αποδοτικότητας. Περιγράφει τα πλεονεκτήματα και τη μεθοδολογία της, ενώ κατασκευάστηκε διαδικτυακή εφαρμογή ως εργαλείο για τη μέτρηση της αποδοτικότητας. Η εφαρμογή αυτή αναπτύχθηκε με γνώμονα τον μη εξοικειωμένο χρήστη, θέλοντας να διευκολύνει εκείνον, ο οποίος έρχεται ίσως για πρώτη φορά σε επαφή με εργαλεία τέτοιου τύπου.

Τέλος, αξίζει να αναφερθεί η χρησιμότητα του δωρεάν και ανοικτού κώδικα λογισμικού, καθώς ιδιαίτερα στον ακαδημαϊκό χώρο όπου οι πόροι είναι περιορισμένοι, μπορούν να αξιοποιηθούν τέτοια εργαλεία με σκοπό τη διευκόλυνση της κοινότητας. Ως αποτέλεσμα αυτού, η εφαρμογή που αναπτύχθηκε παρέχεται σε ανοικτό αποθετήριο στην πλατφόρμα GitHub στη διεύθυνση [https://github.com/iliasvlachos/msc\\_dea](https://github.com/iliasvlachos/msc_dea).

## Βιβλιογραφία

Andersen, Erling D., and Knud D. Andersen. (2000) "The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm." High performance optimization. Springer US.

Ashoor, L. (2012) Performance Analysis integrating Data Envelopment Analysis and Multiple Objective Linear Programming, Διδακτορική Διατριβή Manchester Business School

Banker, R.D., Charnes, A. and Cooper, W.W. (1984) Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. Management Science, 30, 1078-1092.

Banker, R. Cooper, W. Seiford, L. Thrall, R. Zhu, J. (2004), Returns to scale in different DEA models, European Journal of Operational Research, 2004, 154(2), 345-362.

Benneyan, J., Sunnetci, A., Ceyhan, M., (2008) [Data envelopment analysis models for identifying and benchmarking the best healthcare processes](#), International Journal of Six Sigma and Competitive Advantage

Charnes A, Cooper WW, Rhodes E. (1978) Measuring the efficiency of decision making units. European Journal of Operational Research, 2:429-444.

Chu J., Zhu J., (2021) [Production scale-based two-stage network data envelopment analysis](#), European Journal of Operational Research

Cooper, W. Seiford, L. Tone, K. (2006) Introduction to Data Envelopment Analysis and Its Uses, Springer US.

Δεσπότης (Χ.Χ.) Διαφάνειες μαθήματος Τεχνικές Βελτιστοποίησης, Πανεπιστήμιο Πειραιώς

Koronakos G. (2019) [A Taxonomy and Review of the Network Data Envelopment Analysis Literature](#)

Koronakos G. (2017) [The composition and the weak-link approaches to Network Data Envelopment Analysis](#), Διδακτορική Διατριβή, Πανεπιστήμιο Πειραιώς

Κοπελιάρη Μ. (2013) Μοντέλα Περιβάλλουσας Ανάλυσης Δεδομένων σε παραγωγικές διαδικασίες δύο σταδίων, Μεταπτυχιακή Διατριβή Πανεπιστημίου Πειραιώς

## Ηλεκτρονικές πηγές

1. Stack Overflow [Developer Survey 2022](#)
2. Django documentation [How to write reusable apps](#)
3. Django documentation [The admin site](#)
4. Computational Infrastructure for operations Research ([COIN-OR](#))
5. Bootstrap 5 [documentation](#)

## Παράρτημα κώδικα

### Github

Η εφαρμογή είναι διαθέσιμη σε github αποθετήριο στη διεύθυνση [https://github.com/iliasvlachos/msc\\_dea](https://github.com/iliasvlachos/msc_dea)

### Μοντέλα

```
class Project(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField()
    date_created = models.DateTimeField(default=timezone.now)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        return reverse('projects-list')
```

#### Μοντέλο Project

Το μοντέλο αναφέρεται στην οντότητα των project που δημιουργεί ο χρήστης και ομαδοποιεί τα μεταφορτωμένα αρχεία του.

```
class File(models.Model):
    name = models.CharField(max_length=200)
    description = models.TextField()
    file = models.FileField(default='default.txt', upload_to=user_directory)
    date_uploaded = models.DateTimeField(default=timezone.now)
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('projects-detail', kwargs={'pk': self.project.id})
```

#### Μοντέλο File

Το μοντέλο File αναφέρεται στα αρχεία που μεταφορτώνει ο χρήστης εντός του project. Τα αρχεία αυτά θα περιέχουν τα προς ανάλυση δεδομένα.



```
class FileResult(models.Model):
    file = models.ForeignKey(File, on_delete=models.CASCADE)
    date = models.DateTimeField(default=timezone.now)
    results_file = models.FileField(default='default.txt', upload_to=user_results_directory, null=True)

    def __str__(self):
        return "results for :"+self.file.name
```

### Μοντέλο FileResult

Το μοντέλο FileResult αναφέρεται στην αποθήκευση του αρχείου με τα αποτελέσματα της ανάλυσης που διεξήχθει.

### Views

```
@login_required
def lp_solve(request, pk, fi):
    if request.method == 'POST':
        project_file = File.objects.get(pk=fi)
        names = getFileColumns(project_file.file.path).keys()
        inputs = []
        outputs = []
        orientation = request.POST.get('minmax')
        # orientation = "output"
        for name in names:
            if request.POST.get(name + '_input') == '1':
                inputs.append(name)
            else:
                outputs.append(name)
        if orientation == 'input':
            minmax = 'Max'
            print("minmax = " + minmax)
        else:
            minmax = 'Min'
            print("minmax = " + minmax)
        rts = request.POST.get('rts')
        result_data = pulp_solve(project_file, inputs, outputs, minmax, rts)
        context = {
            'result_data': result_data
        }
        return render(request, 'file_result/pulp_result_data.html', context)
    else:
        project_file = File.objects.get(pk=fi)
        if project_file.user != request.user:
            raise PermissionDenied()
```

```

names = getFileColumns(project_file.file.path).keys()
context = {
    'names': names,
    'file': project_file
}
return render(request, 'files/project_file_lp_solve.html', context)

```

### Μέθοδος lp\_solve

Η μέθοδος lp\_solve αναλαμβάνει την εμφάνιση της φόρμας εισαγωγής των επιλογών του χρήστη σχετικά με την ανάλυση και τη λήψη αυτών ώστε να εισαχθούν παραμετρικά στη μέθοδο επίλυσης του προβλήματος pulp\_solve(project\_file, inputs, outputs, minmax, rts)

```

def pulp_solve(file, inputs, outputs, minmax, rts):
    ten_8 = 1 / math.pow(10, 8)
    columns = getFileColumns(file.file.path)
    rows = getFileRows(file.file.path)
    results = {}
    objective_values = {}
    eff_ratios = {}
    # create problem
    if minmax == "Max":
        # input orientation: MM problem is max
        orientation = LpMaximize
    else:
        # output orientation: MM problem is min
        orientation = LpMinimize
    problem_name = str(file.name)
    problem_name = problem_name.replace(" ", "_")

    # for each row (DMU)
    for (index, row) in enumerate(rows):
        problem = LpProblem(problem_name, orientation)
        row_vars = []
        input_obj = []
        output_obj = []
        # print("DMU " + str(index) + " is " + str(row))
        # for each column (variable)
        for col in columns:
            col_name = col.replace(" ", "_")
            # print("column " + str(col_name) + " = " + str(columns[col][index]))
            # define the variables
            col_name = LpVariable(name=col_name, lowBound=0)
            # append var in list
            row_vars.append(col_name)
            # append the variable with its value in inputs or outputs
            if col in inputs:

```

```

        input_obj.append(int(columns[col][index]) * col_name)
    elif col in outputs:
        output_obj.append(int(columns[col][index]) * col_name)
# for vrs add extra variable
if rts == "vrs":
    v_zero = LpVariable(name="var_zero")

    if minmax == 'Min':
        problem.addConstraint(LpConstraint(lpSum(output_obj), sense=LpConstraintEQ, rhs=1,
name="equality constraint"))
        # if vrs add extra variable
        if rts == "vrs":
            input_obj.append(v_zero * (-1))
        # if output oriented then build objective function with inputs (min inputs)
        obj = lpSum(input_obj)
    elif minmax == 'Max':
        problem.addConstraint(LpConstraint(lpSum(input_obj), sense=LpConstraintEQ, rhs=1,
name="equality constraint"))
        # if vrs add extra variable
        if rts == "vrs":
            output_obj.append(v_zero * (-1))
        # if input oriented then build objective function with outputs (max outputs)
        obj = lpSum(output_obj)

problem += obj

# build inequality constraints
for (row_index, row_name) in enumerate(rows):
    constraint_row_inputs = []
    constraint_row_outputs = []
    c_index = "c_" + str(row_index)
    # print(c_index)
    for (col_index, col) in enumerate(columns):
        if col in inputs:
            constraint_row_inputs.append(int(columns[col][row_index]) * row_vars[col_index])
        elif col in outputs:
            constraint_row_outputs.append(int(columns[col][row_index]) * row_vars[col_index])
    if rts == "vrs" and minmax == "Min":
        constraint = lpSum(constraint_row_inputs) - lpSum(constraint_row_outputs) - v_zero
    elif rts == "vrs" and minmax == "Max":
        constraint = lpSum(constraint_row_inputs) - lpSum(constraint_row_outputs) + v_zero
    else:
        constraint = lpSum(constraint_row_inputs) - lpSum(constraint_row_outputs)
    problem.addConstraint(LpConstraint(e=constraint, sense=LpConstraintGE, name=c_index,
rhs=0))

problem.solve(PULP_CBC_CMD(mip=False))
results[row] = problem

```

```

objective_value = round(problem.objective.value(), 4)
objective_values[row] = objective_value
# calculate efficiency ratios %
if minmax == 'Min':
    # minimization problem, so eff = 1/obj
    eff_ratio = (1 / objective_value) * 100
else:
    eff_ratio = objective_value * 100
eff_ratios[row] = round(eff_ratio, 4)

# analysis details dict
analysis = {'orientation': minmax, 'rts': rts, 'inputs': inputs, 'outputs': outputs}

# calculate efficiency projections
projections = calculate_projections(results, minmax, objective_values, rows, columns, inputs,
outputs)

# results
result_data = {
    "details": analysis,
    "results": results,
    "obj_values": objective_values,
    "eff_ratios": eff_ratios,
    "projections": projections
}

# create and save results in file
result_file = save_results_file(file, result_data)

return result_data

```

#### Μέθοδος pulp\_solve

Η μέθοδος αναλαμβάνει να κατασκευάσει το γραμμικό πρόβλημα για κάθε μονάδα ώστε να μοντελοποιηθεί κατάλληλα και να επιλυθεί με την PuLP. Για τον υπολογισμό των προβολών κατασκευάστηκε επιπλέον μέθοδος η οποία παρουσιάζεται παρακάτω.

```

def calculate_projections(results, minmax, obj_values, rows, columns, inputs, outputs):
    l_dict = {}
    l_dict_lists = {}
    projection_values = {}
    projection_sums = {}

    # for each DMU
    for row, result in results.items():
        c_dict = {}

```

```

l_list = []
# if DMU is inefficient
if obj_values[row] != 1.0:
    for name, constraint in result.constraints.items():
        # skip first constraint
        if name != 'equality_constraint':
            c_dict[name] = constraint.pi
            l_list.append(constraint.pi)

    l_dict[row] = c_dict
    l_dict_lists[row] = l_list

rows_list = list(rows.values())

for dm_u, l_items in l_dict_lists.items():
    col_lists = []
    for col in columns:
        col_lists.append([])

    for (item_index, l_item) in enumerate(l_items):
        for (col_index, col) in enumerate(columns):
            col_value = l_item * columns[col][item_index]
            col_lists[col_index].append(col_value)

    projection_values[dm_u] = col_lists

for row, result in projection_values.items():
    projection_sums_list = []
    for (col_index, col) in enumerate(columns):
        projection_sums_list.append(sum(result[col_index]))
    projection_sums[row] = projection_sums_list

columns_list = []
for col in columns:
    columns_list.append(col)
projection_sums["cols"] = columns_list
# print(projection_sums)
return projection_sums

```

**Μέθοδος calculate\_projections**

Η μέθοδος αυτή λαμβάνει ως input το αποτέλεσμα της pulp solve() υπολογίζει για κάθε μη αποδοτική μονάδα την τιμή για κάθε μεταβλητή της και επιστρέφει ένα dictionary από λίστες. Τα keys είναι τα ονόματα κάθε μονάδας και οι λίστες περιέχουν τις τιμές των μεταβλητών.

Τέλος για την παραγωγή του αρχείου αποτελεσμάτων καλείται η save\_results\_file

```
def save_results_file(file, result_data):
    # print(result_data["details"]["orientation"])
    # print(result_data)
    # build pdf data
    buffer = io.BytesIO()
    doc = SimpleDocTemplate(buffer, pagesize=letter,
                            rightMargin=72, leftMargin=72,
                            topMargin=72, bottomMargin=18,
                            author='Django DEA App')
    doc.title = "DEA results"
    Story = []
    styles = getSampleStyleSheet()
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Results for " + file.name, styles['Title']))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph(datetime.today().strftime('%d/%m/%Y'), styles["Normal"]))
    Story.append(Spacer(1, 12))
    if result_data["details"]["orientation"] == 'Max':
        orientation = "Input"
    else:
        orientation = "Output"
    Story.append(Paragraph('Orientation: ' + orientation, styles["Normal"]))
    Story.append(Spacer(1, 12))
    if result_data["details"]["rts"] == 'crs':
        rts = 'Constant'
    else:
        rts = 'Variable'
    Story.append(Paragraph('Returns to Scale: ' + rts, styles["Normal"]))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Inputs", styles["Heading2"]))
    Story.append(Spacer(1, 12))
    for input in result_data["details"]["inputs"]:
        Story.append(Paragraph(input, styles["Normal"]))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Outputs", styles["Heading2"]))
    Story.append(Spacer(1, 12))
    for output in result_data["details"]["outputs"]:
        Story.append(Paragraph(output, styles["Normal"]))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Analysis results", styles["Heading1"]))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Objective function values and efficiency ratios %", styles["Heading2"]))
    Story.append(Spacer(1, 12))
    for obj, value in result_data["obj_values"].items():
        Story.append(Paragraph("For DMU " + obj + ": Obj. Func. = " + str(value) + " | Efficiency = " +
                               str(result_data["eff_ratios"][obj]) + " %", styles["Normal"]))
    Story.append(Spacer(1, 12))
    Story.append(Paragraph("Efficiency Projections", styles["Heading2"]))
    Story.append(Spacer(1, 12))
```

```
for obj, projections in result_data["projections"].items():
    if obj != 'cols':
        Story.append(Paragraph("For DMU " +obj, styles["Heading3"]))
        for index, col in enumerate(result_data["projections"]["cols"]):
            Story.append(Paragraph(col + " = " + str(projections[index]), styles["Normal"]))
        Story.append(Spacer(1, 12))
    Story.append(Spacer(1, 12))

doc.build(Story)
buffer.seek(0)
pdf_file = File(buffer, name=file.name + "_results.pdf")

# insert in FileResult class
file_results = FileResult(file=file, results_file=pdf_file)
file_results.save()
return file_results
```

**Μέθοδος save\_results\_file**

Με αυτή τη μέθοδο παράγεται ένα PDF αρχείο το οποίο περιέχει τα αποτελέσματα της ανάλυσης, το οποίο αποθηκεύεται στο μοντέλο FileResult για μελλοντική χρήση.

```
def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            messages.success(request, f'Account created for {username}')
            return redirect('dea-index')
        else:
            form = UserRegisterForm()
            return render(request, 'users/register.html', {'form': form})
```

**Μέθοδος register**

Η μέθοδος register αναλαμβάνει την δημιουργία νέου λογαριασμού κατά την εγγραφή του χρήστη χρησιμοποιώντας το Authentication System του Django