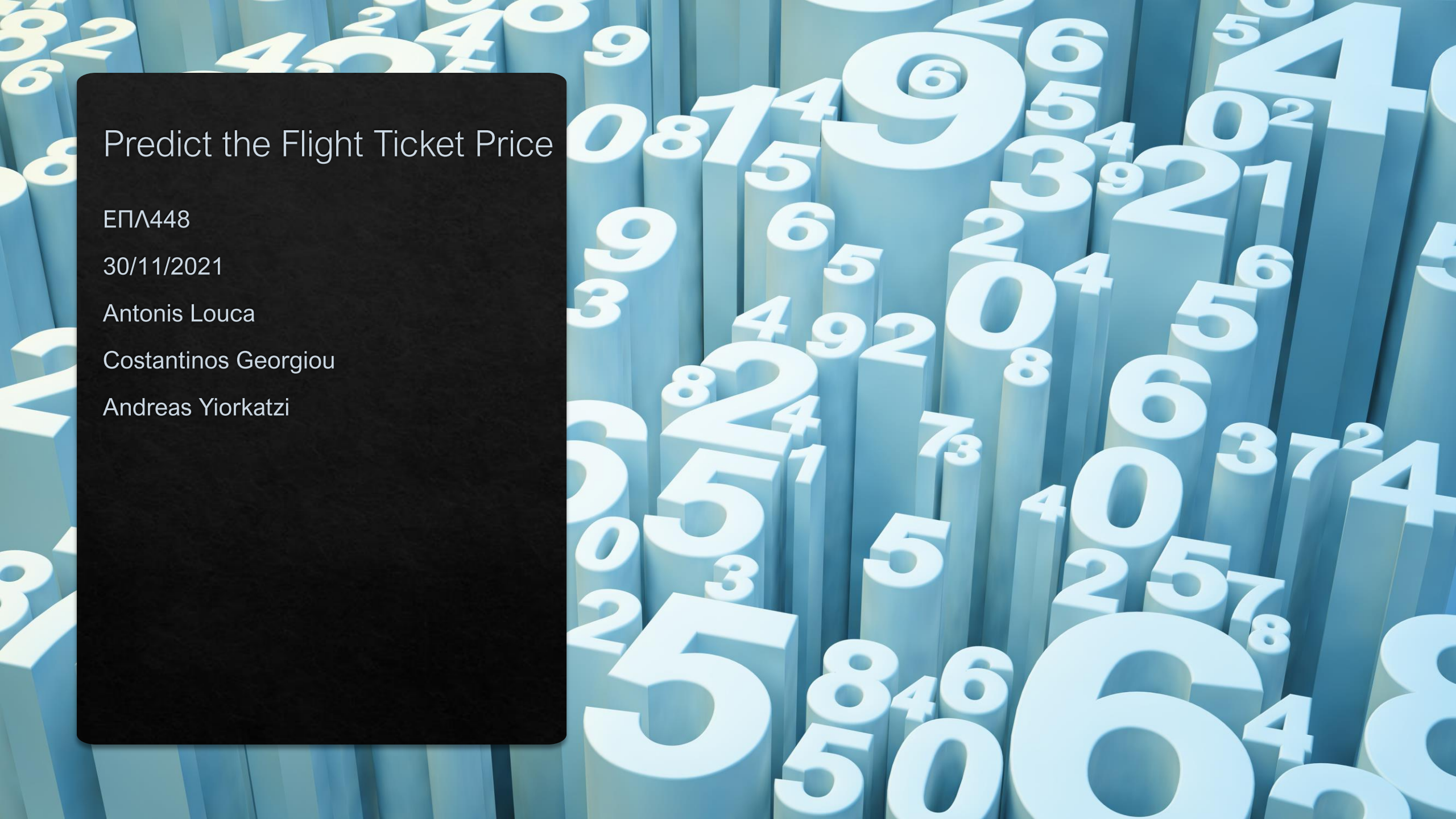


Predict the Flight Ticket Price

ΕΠΛ448

30/11/2021

Antonis Louca

Costantinos Georgiou

Andreas Yiorkatzi

Project Milestones

1η φάση

Ορισμός προβλήματος και Εξερεύνηση Δεδομένων

2η φάση

Εκπαίδευση/ Μάθηση αλγόριθμων (learning/training) και πρόβλεψη (prediction)

3η φάση

Αξιολόγηση αλγορίθμου (testing)

1η Φάση Ορισμός προβλήματος και Εξερεύνηση Δεδομένων

- Πρόβλημα πρόβλεψης (Regression)
- Ανάλυση δεδομένων (10683x11)
- Data Selection
- Data Preprocessing
- Data Transformation

Data Selection

- Διαγραφή features τα οποία δεν είχαν κάτι να προσφέρουν
- Αφαίρεση γραμμών που έλειπαν σημαντικά data
- Ακολουθήσαμε 2 παραλλαγές
  - 1 χωρίς την στήλη Route
  - 1 κρατώντας την στήλη Route

Data Preprocessing

AIRLINE

DEP TIME

DATE OF JOURNEY

SOURCE KAI DESTINATION

DURATION

TOTAL STOPS

ADDITIONAL INFO

ROUTE

One Hot encodings

Airline	Source	Destination
• 12 airlines • Dummy Encoding (One Hot)	• 5 Source Cities • Dummy Encoding (One Hot)	• 6 Destination Cities • Dummy Encoding (One Hot)

Cyclical Feature encoding

Date of Journey

- Month and Day
- Cyclical Month
- Cyclical Day (Of the week: Monday=0, Sunday=6)

Departure Time

- Changed from HH:MM to HH.MM
- Cyclical min: 00.00 max 23.59

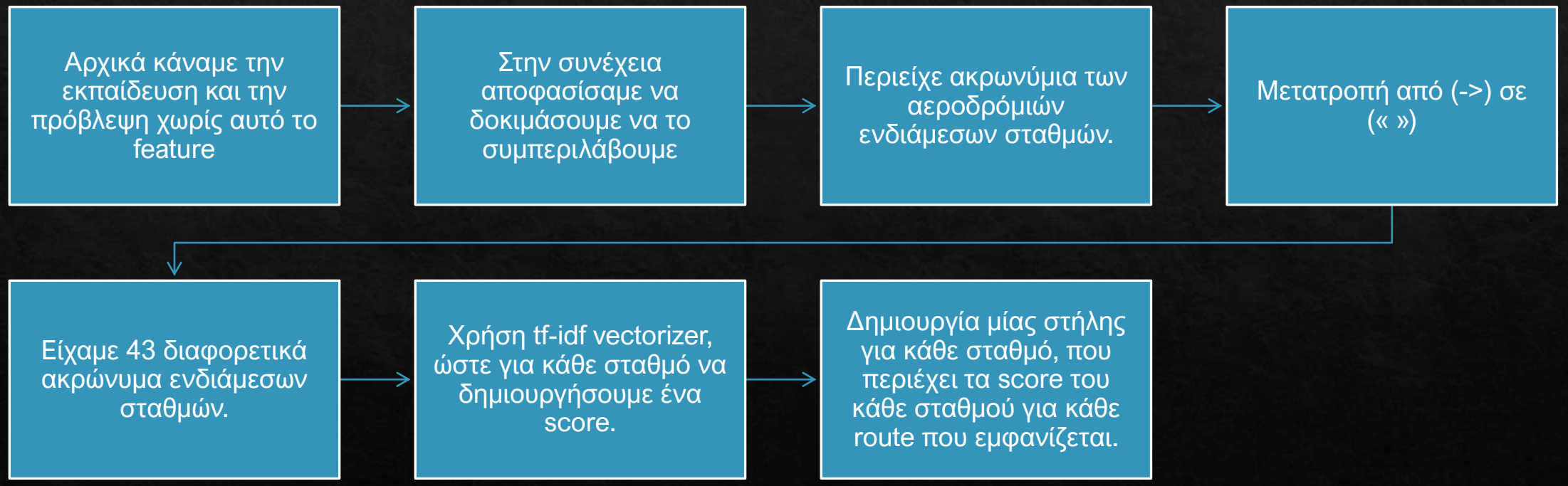
Ordinal Encoding – Additional Info

8 διαφορετικές επιπρόσθετες πληροφορίες για κάθε πτήση

Αύξουσα σειρά σε σχέση με το πόσο προσδίδει στο κόστος του εισιτηρίου

- 0 → No food
- 1 → in-flight meal not included
- 2 → no check in baggage included
- 3 → red-eye flight
- 4 → 1 long layover
- 5 → 1 short layover
- 6 → 2 long layovers
- 7 → change airports
- 8 → business class

Route -TF-IDF


Αρχικά κάναμε την εκπαίδευση και την πρόβλεψη χωρίς αυτό το feature
Στην συνέχεια αποφασίσαμε να δοκιμάσουμε να το συμπεριλάβουμε
Περιείχε ακρωνύμια των αεροδρόμιών ενδιάμεσων σταθμών.
Μετατροπή από (->) σε (« »)
Είχαμε 43 διαφορετικά ακρώνυμα ενδιάμεσων σταθμών.
Χρήση tf-idf vectorizer, ώστε για κάθε σταθμό να δημιουργήσουμε ένα score.
Δημιουργία μίας στήλης για κάθε σταθμό, που περιέχει τα score του κάθε σταθμού για κάθε route που εμφανίζεται.

Αλλαγή μορφής

Duration

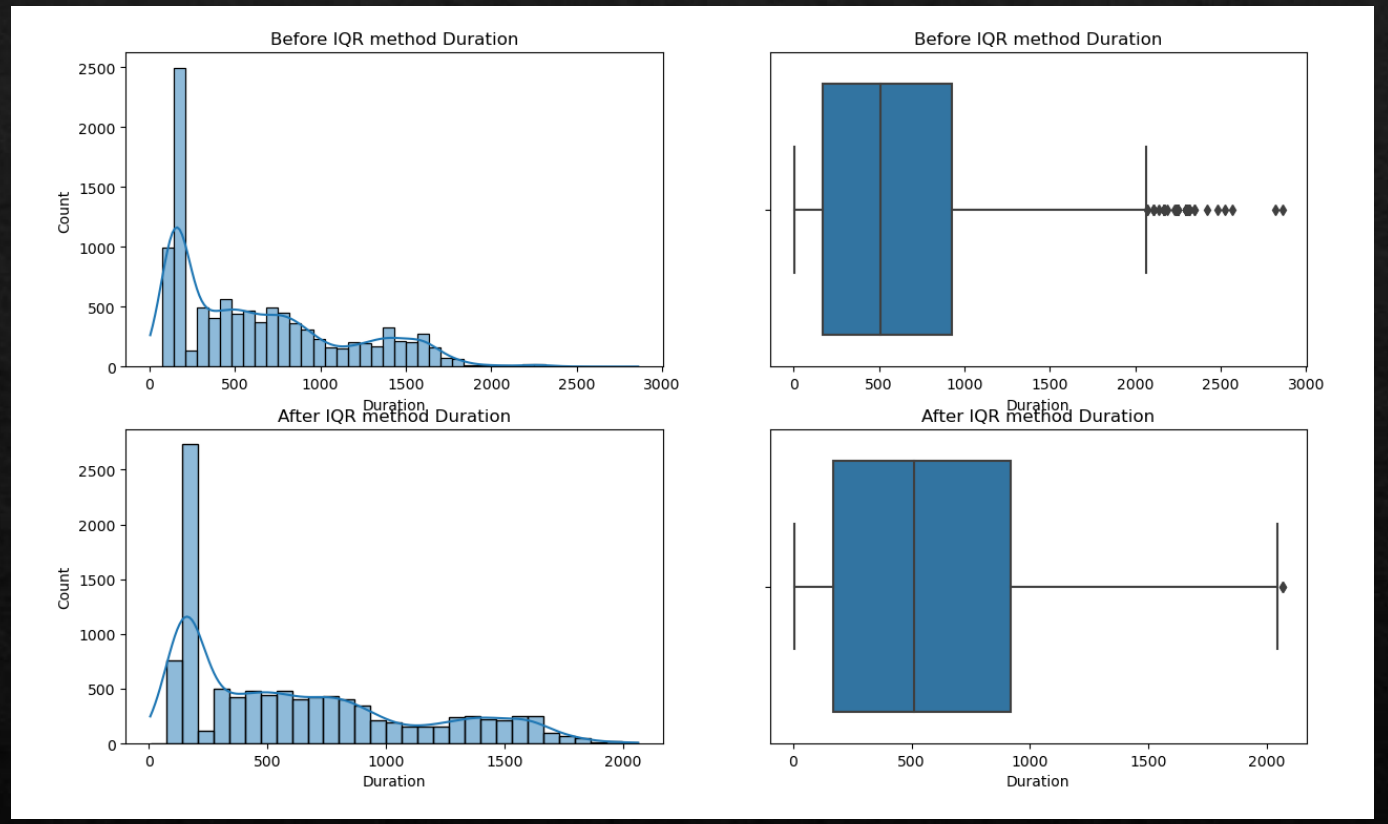
- Από μορφή 1h20m σε 80 λεπτά

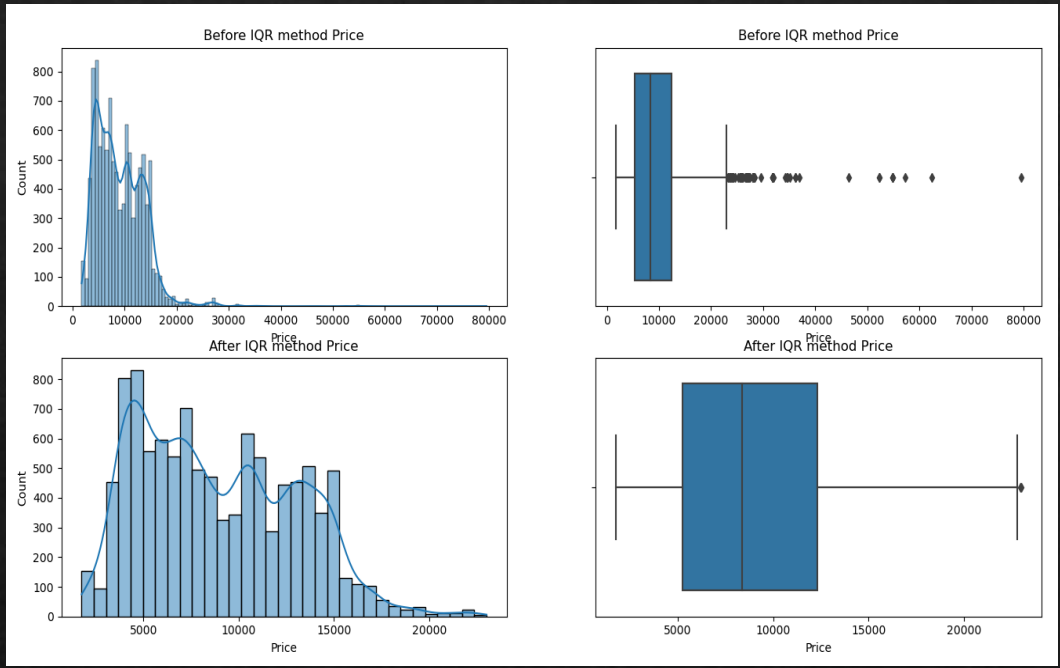
Total Stops

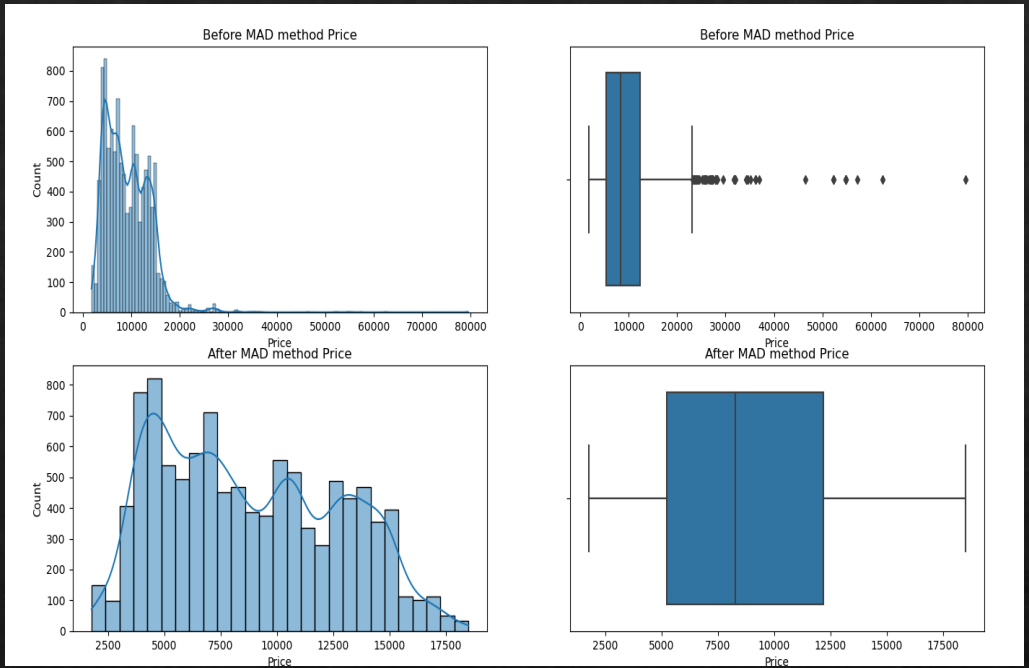
- Από 1 stop, 2 stops, κτλ. τα μετατρέψαμε σε ένα ακέραιο μόνο

Εντοπισμός των Outliers

- Εντοπίστηκαν outlier στις στήλες “Duration”, “Total_Stops” και “Price”
- Χρησιμοποιήσαμε την μέθοδο IQR διότι συγκριτικά με την MAD μπορούσε να βρει τα outliers πολύ πιο αποδοτικά

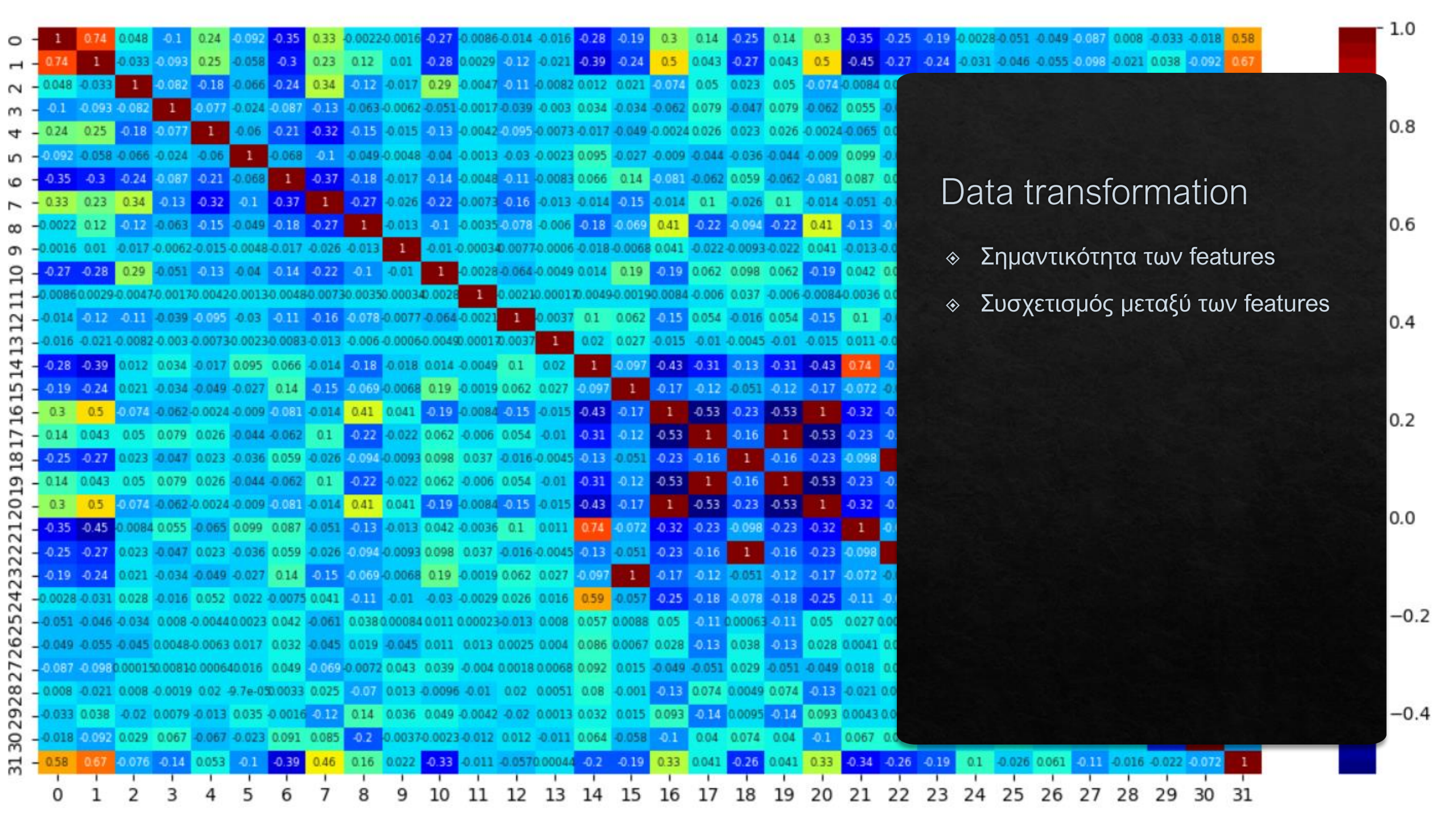



Before IQR method Price
Count
Price
Before IQR method Price
Price
After IQR method Price
Count
Price
After IQR method Price
Price


Before MAD method Price
Count
Price
Before MAD method Price
Price
After MAD method Price
Count
Price
After MAD method Price
Price

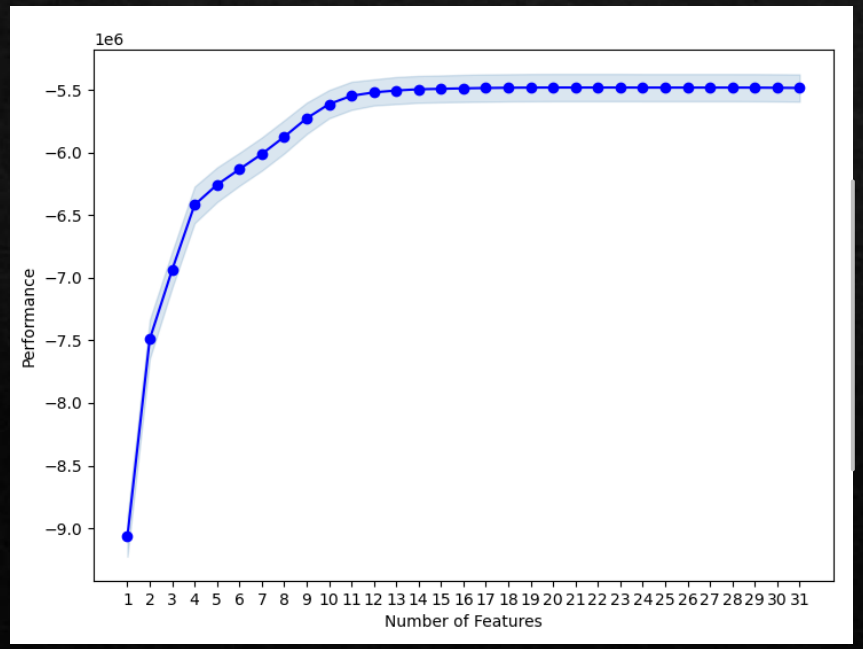
IQR vs MAD


Data
Transformation

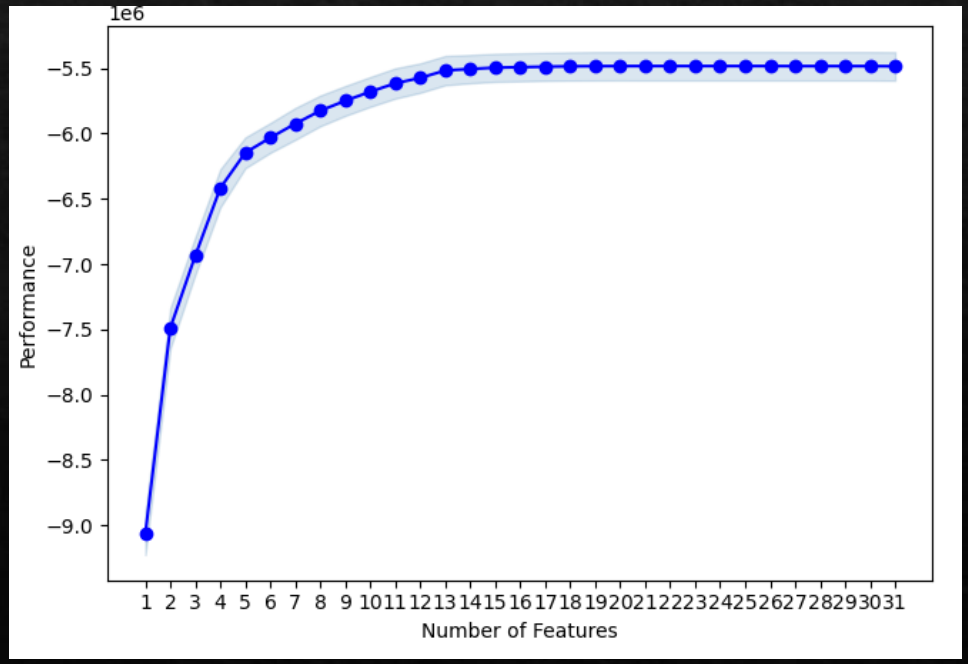

Data transformation
Σημαντικότητα των features
Συσχετισμός μεταξύ των features

2 Τεχνικές που χρησιμοποιήσαμε

Backward Elimination



Forward Selection



2η Φάση: Εκπαίδευση/Μάθηση αλγόριθμων (learning/training) και πρόβλεψη (prediction)

Αλγόριθμοι που χρησιμοποιήσαμε

Linear Regressor	Polynomial Regressor	Support Vector Regressor (SVR)	Random Forest Regressor
Lasso Regressor	Decision Tree Regressor	Gradient Boost Regressor	Light Gradient Boosting Machine (optimized library from Microsoft)

GridSearchCv

01

Δημιουργήσαμε ένα parameter grid για τον κάθε αλγόριθμο

02

Χρησιμοποιήσαμε GridSearchCv με cv=5 για να βρούμε τις καλύτερες παραμέτρους για κάθε αλγόριθμο.

03

Χρονοβόρα διαδικασία

Evaluation function

Χωρίσαμε το dataset μας σε training (70%) και testing (30%)

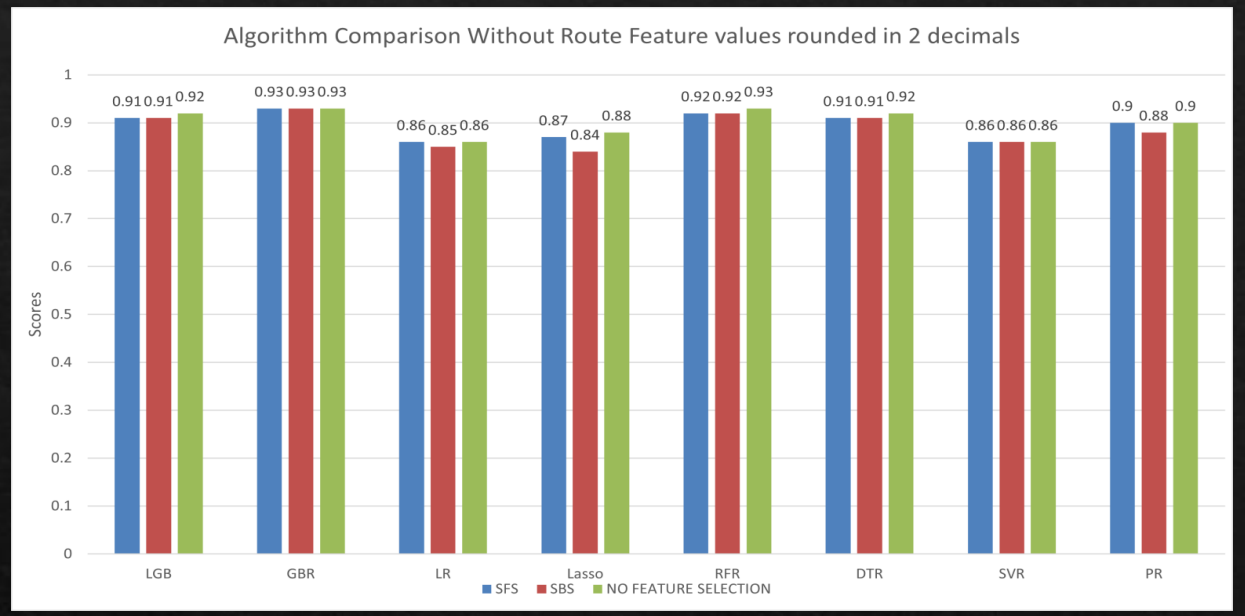
Αφού εκπαιδεύσαμε τους αλγόριθμους με τις κατάλληλες παραμέτρους (στο training set)

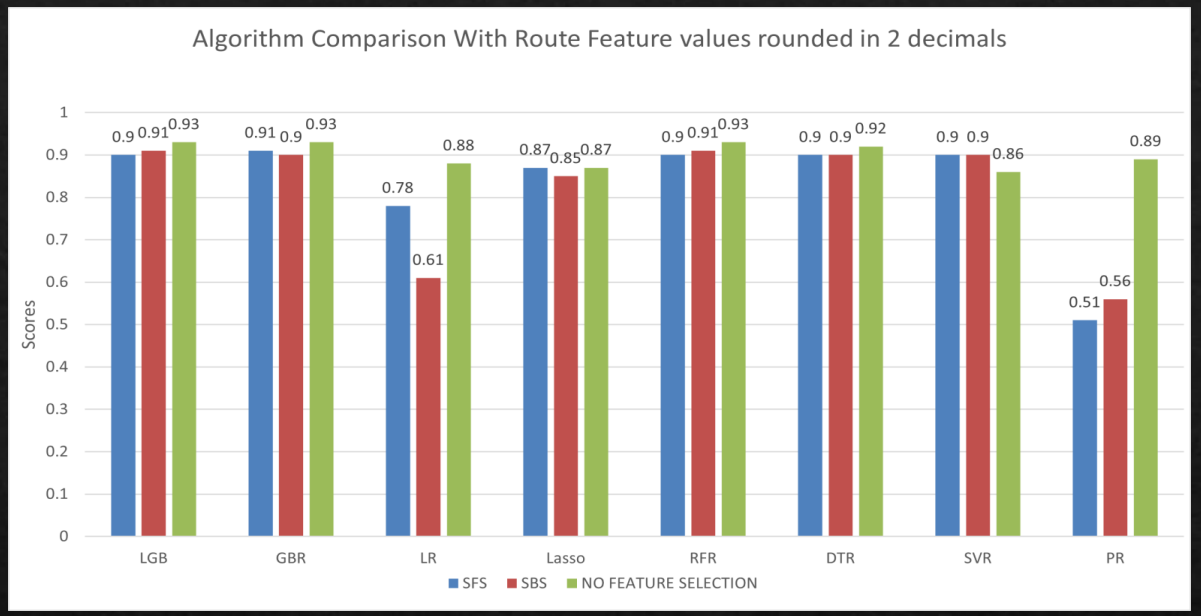
Πρόβλεψη με την χρήση κάθε αλγόριθμου

Χρησιμοποιήσαμε την evaluation function

1 - np.sqrt(np.square(np.log10(y_pred +1) - np.log10(y_true +1)).mean()

Η οποία δινόταν από την σελίδα του διαγωνισμού για να συγκρίνουμε τα αποτελέσματα μας.





3η Φάση: Αξιολόγηση αλγορίθμου (testing)

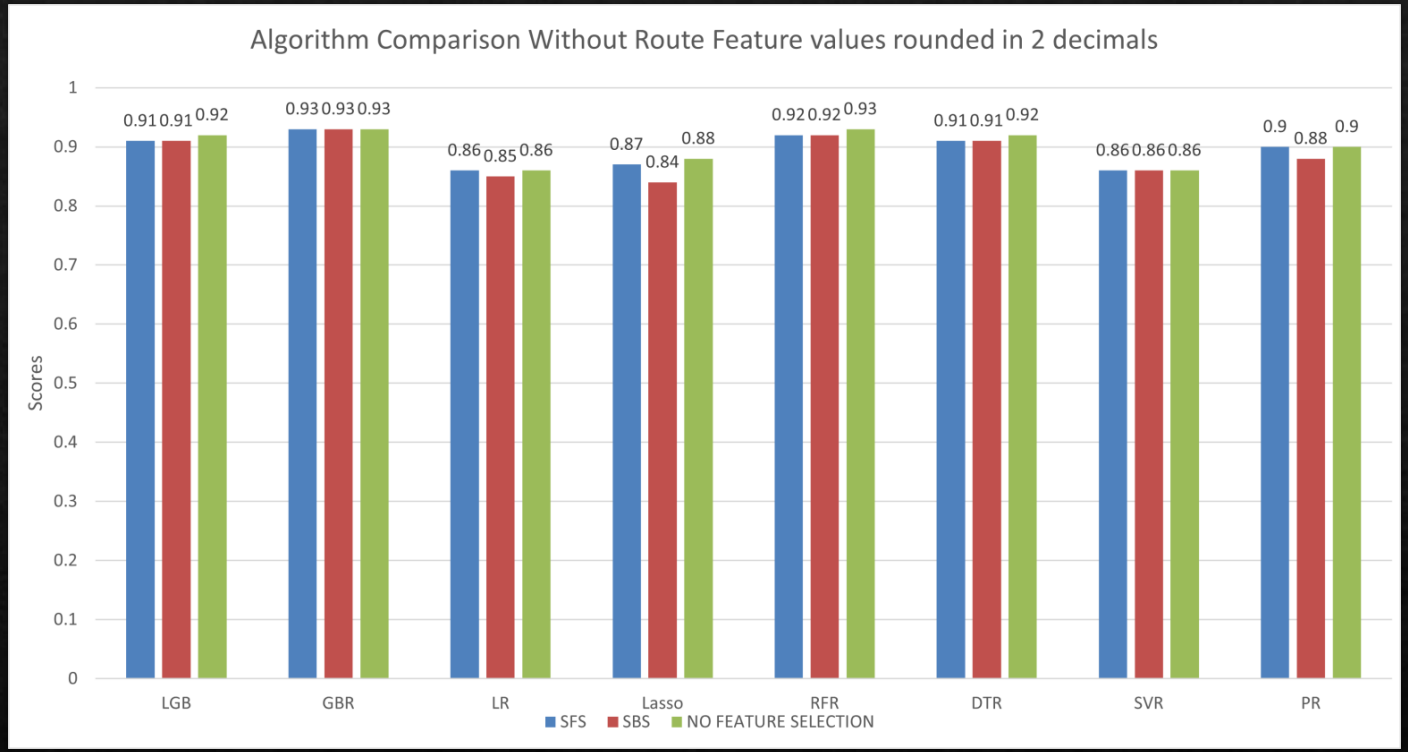
- Τα αποτελέσματα χωρίζονται σε αποτελέσματα με route και χωρίς.
- Διακρίνουμε επίσης την χρήση feature selection, σε SFS, SBS και χωρίς feature selection

Αποτελέσματα χωρίς την χρήση της στήλης Route

Καλύτερα αποτελέσματα: gradient boost regressor (0.93) & random forest χωρίς feature selection.

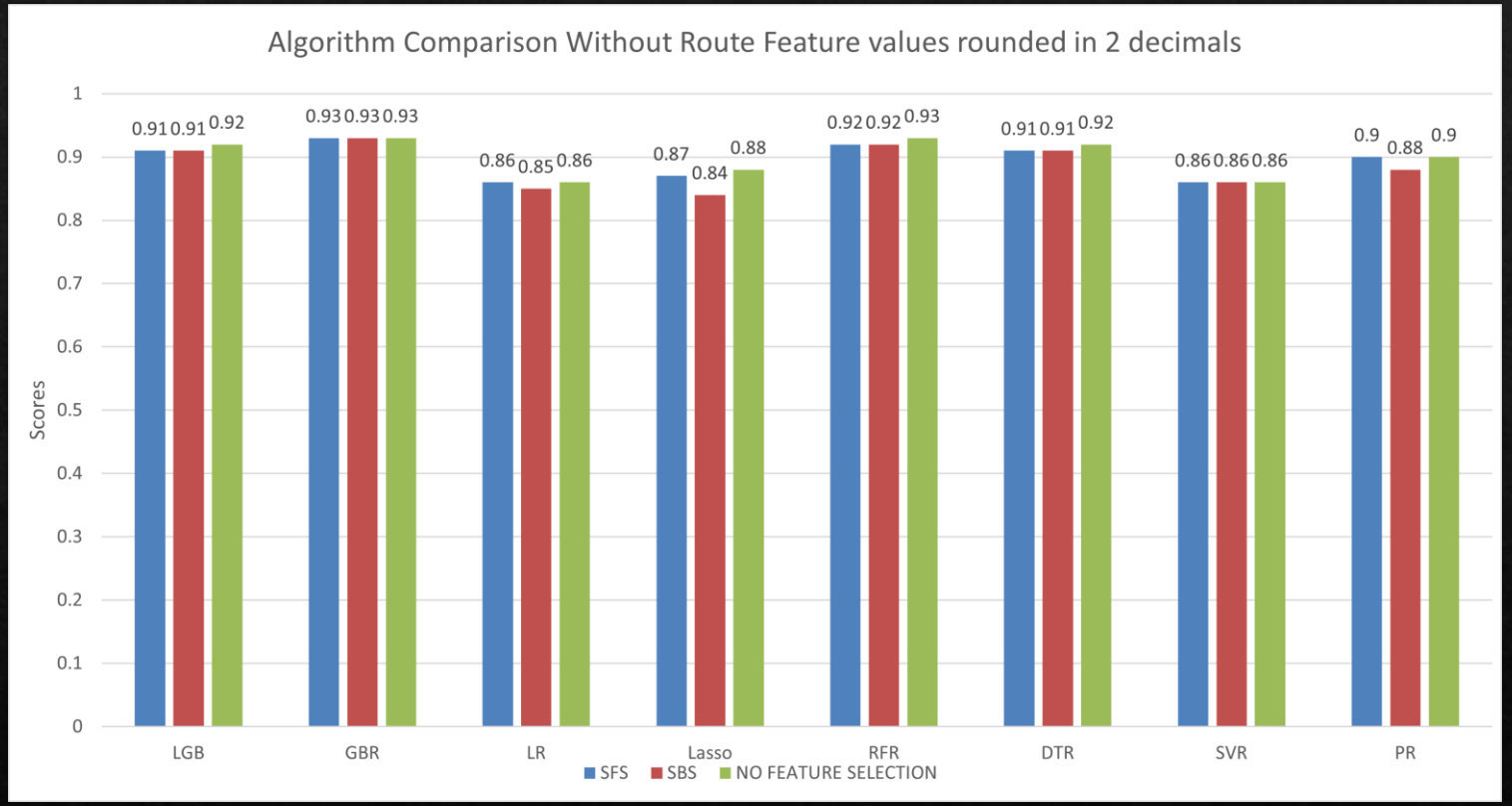
SFS & SBS στους περισσότερους αλγόριθμους έχουν παρόμοια αποτελέσματα.

Ο SBS υστερεί σε lasso, linear και polynomial regressors.

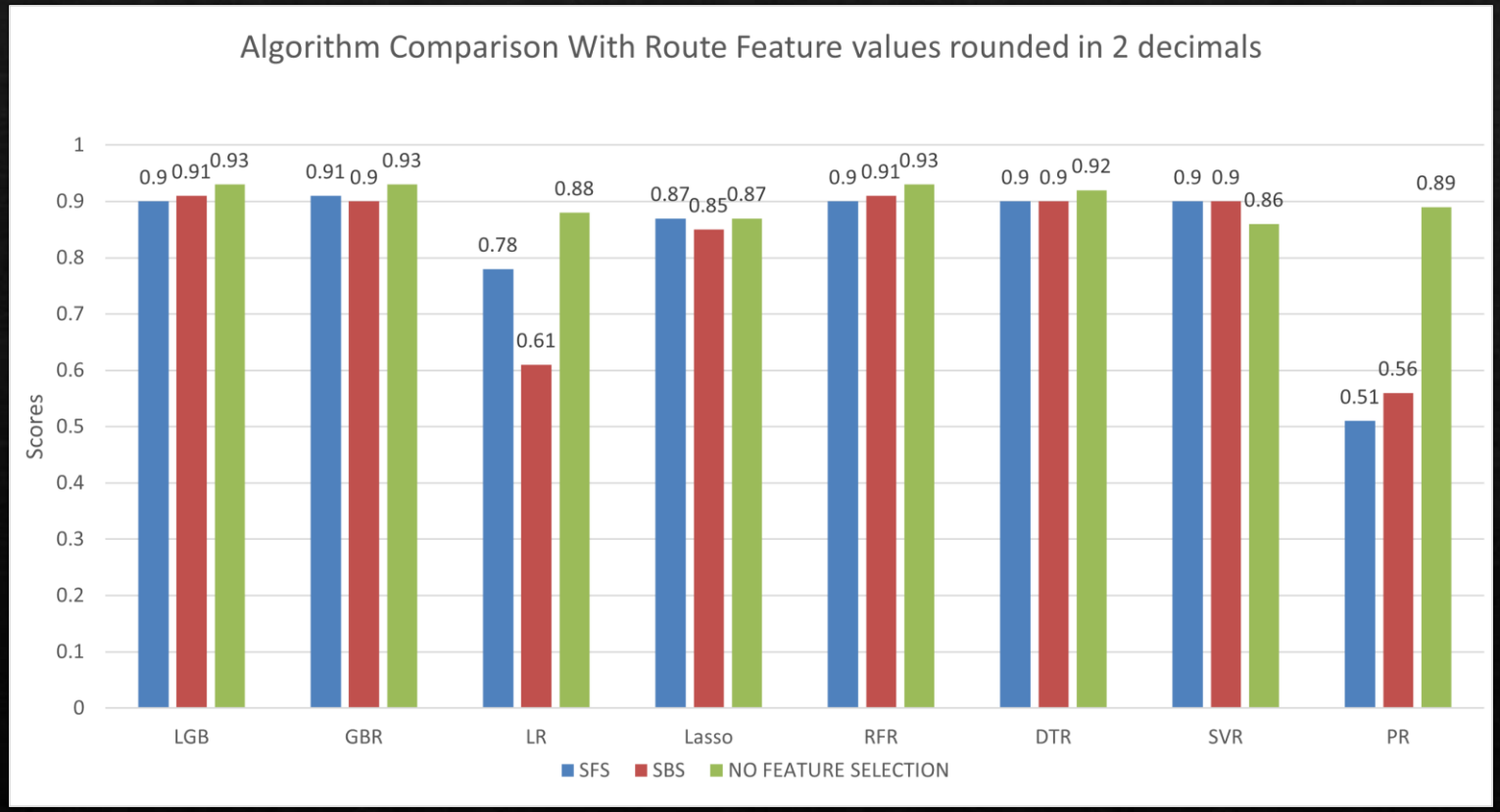


Αποτελέσματα χωρίς την χρήση της στήλης Route

- Ο SBS χρησιμοποιεί περισσότερα features.
- Προτίμηση SFS παρά SBS.
- Χωρίς feature selection, έχουμε καλυτέρα γενικά αποτελέσματα.
- Οι LGB και decision tree, ανεβαίνουν στα 0.92.



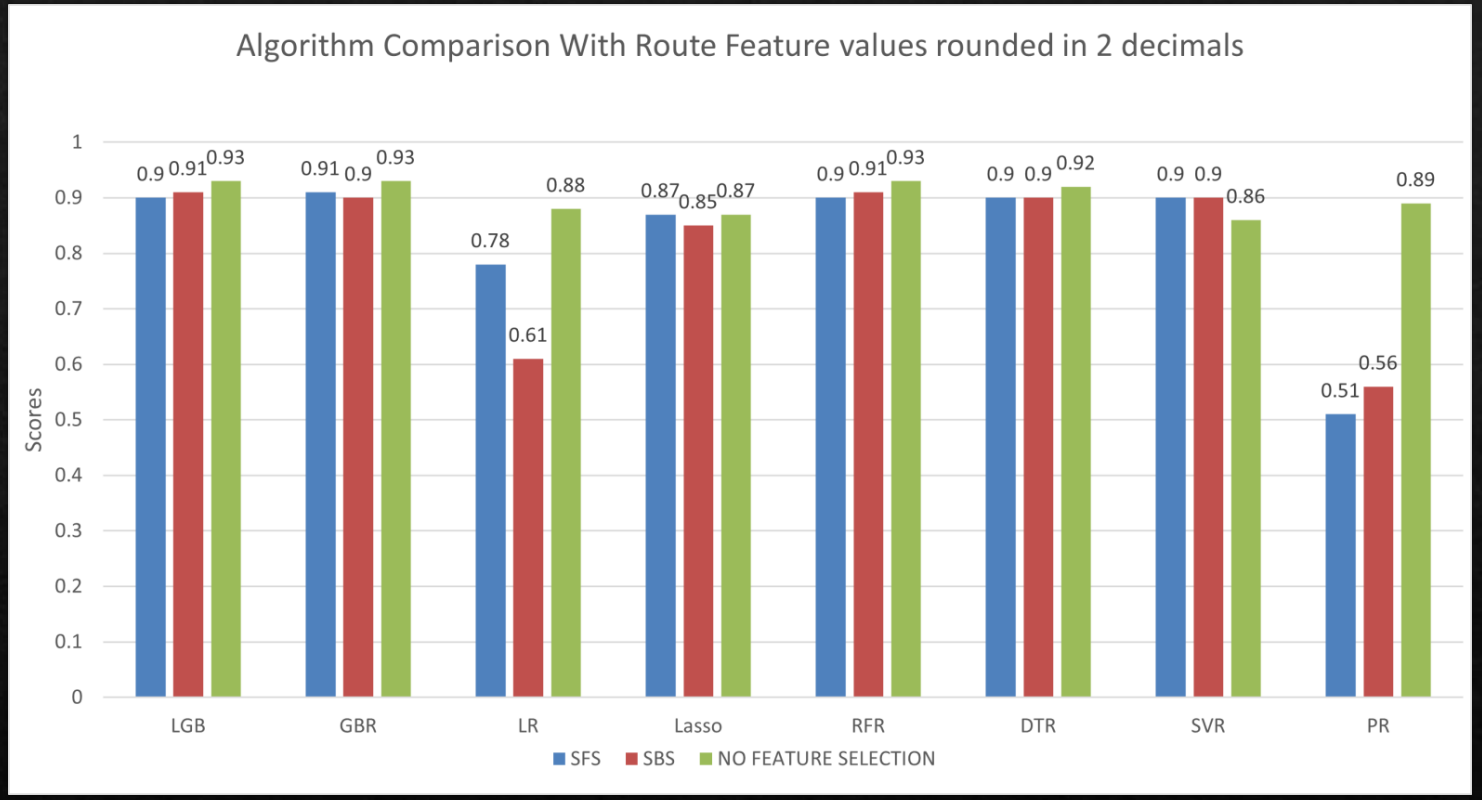
Αποτελέσματα με την χρήση της στήλης Route



- Δεν εκμεταλλεύονται όλοι οι αλγόριθμοι την έξτρα πληροφορία.
- Κάποιοι όμως το λαμβάνουν υπόψιν και φτάνουν στο 0.93.
- Με feature selection ο linear & polynomial regressors έχουν χαμηλά scores.
- Ο συνδυασμός SBS/SFS και Route δίνει μέγιστο score 0.91 και ελάχιστο 0,51.

Αποτελέσματα με την χρήση της στήλης Route

- Χωρίς feature selection, καλύτερα γενικά αποτελέσματα.
- Οι Gradient Boost Regressor και Random Forest Regressor με scores 0.93 εξακολουθούν να είναι οι καλύτεροι.
- Ο LGB φτάνει και αυτός στο 0.93 score.



Thank you!!!