2021

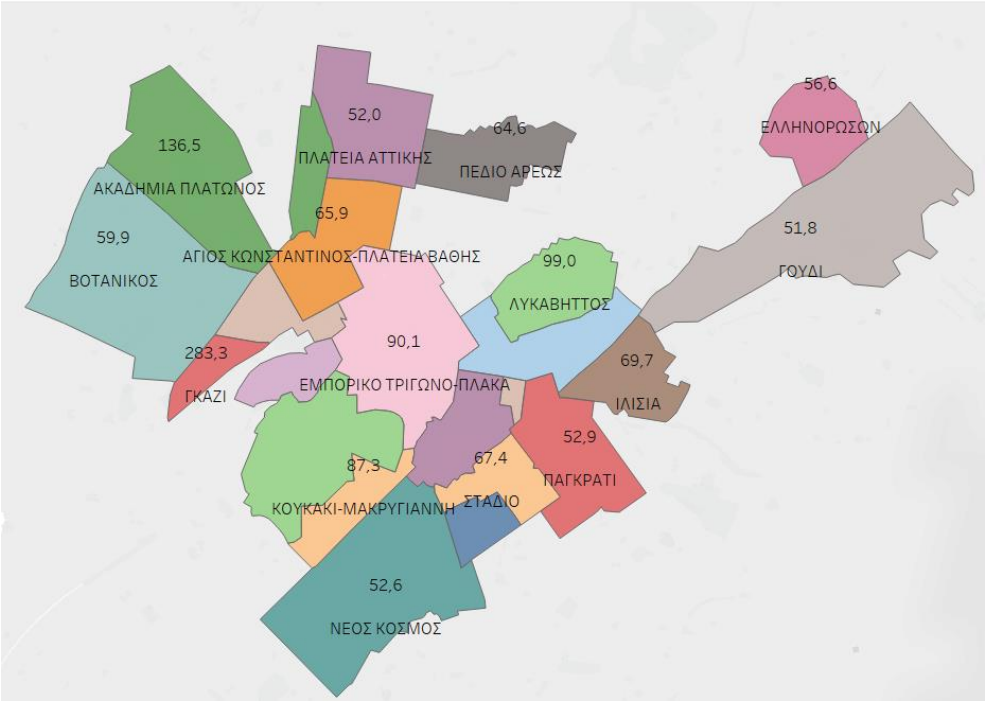
Εξόρυξη δεδομένων

ΣΤΕΒΗΣ ΕΛΕΥΘΕΡΙΟΣ 217131

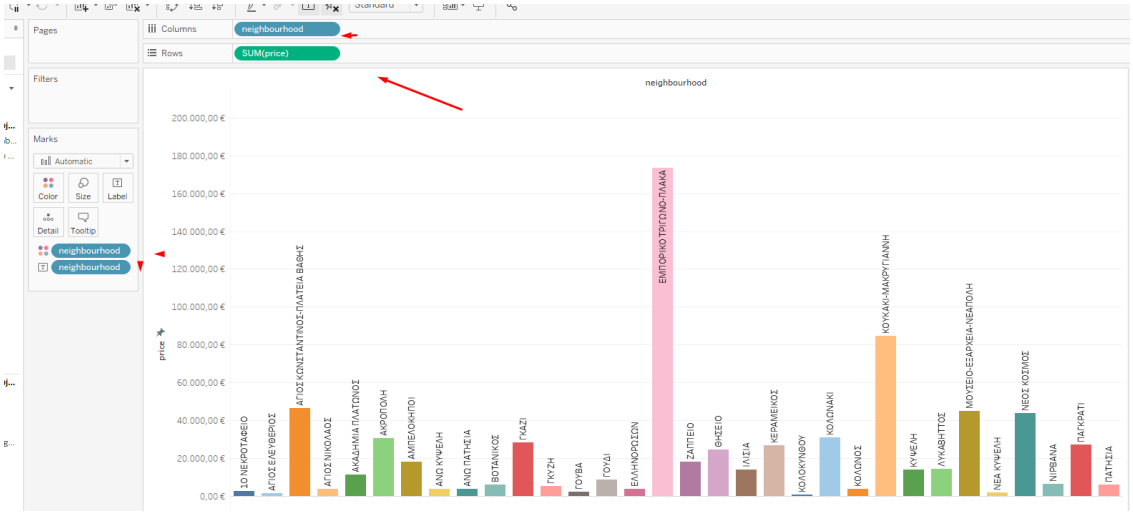
TABLEAU ΓΙΑ ΟΠΤΙΚΟΠΟΙΗΣΗ

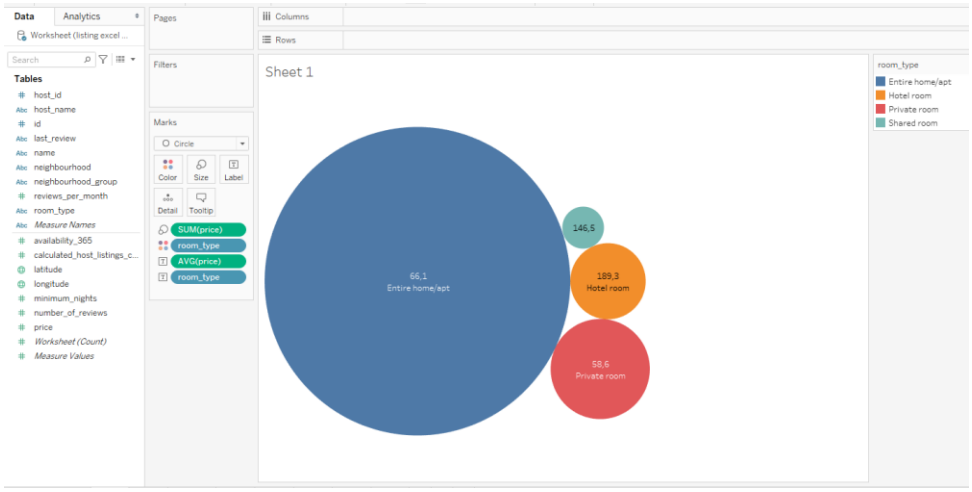
Το πρόγραμμα tableau είναι για την οπτικοποιηση το δεδομένων , είναι ένα διάσημο πρόγραμμα και έχει και όμορφα αποτελέσματα

1. Στην αρχή περνώ τα δεδομένα από το listings.csv και τα αναλύω
2. Έπειτα βλέπω ποια είναι τα πια θα μπορέσουν μου δώσουν ένα ευανάγνωστο αποτέλεσμα ως προς το χρήστη
3. Στην συνέχεια θα προσθέσω ακόμα ένα αρχείο **neighbourhoods.geojson** που είναι ένα αρχείο που θα μας δώσει της διάφορες γειτονίες της Αθήνας η θεσσαλονικείς και θα μας βοηθήσει στην οπτικοποιηση τον δεδομένων

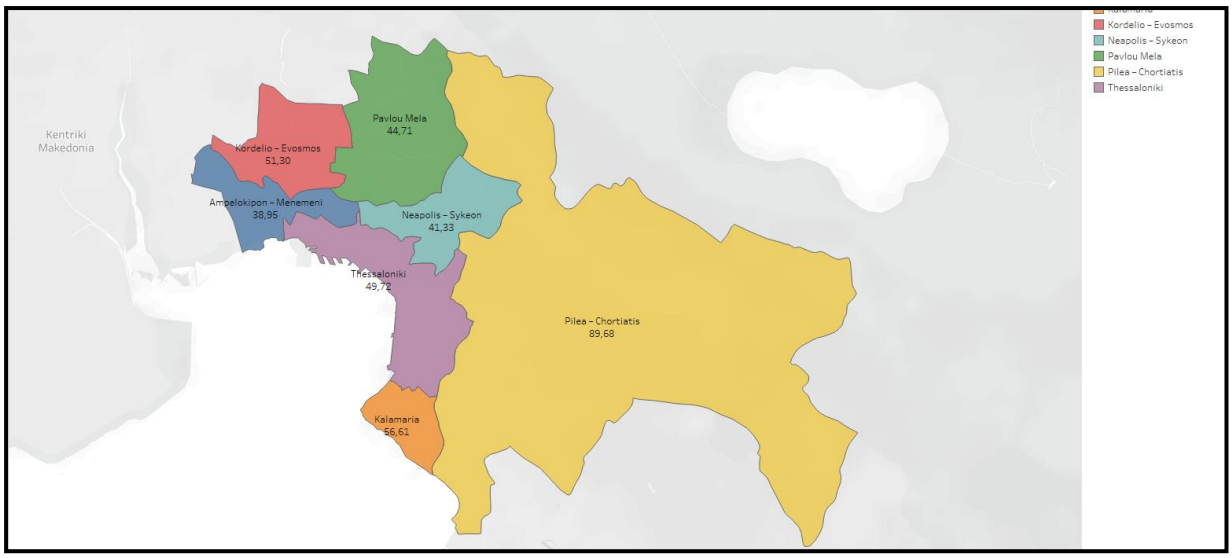


4. Σε άλλο βήμα έχω της γειτονίες από το αρχικό dataset και έπειτα τοποθετώ της τιμές και έτσι λαμβάνω την πιο ακριβή γειτονιά



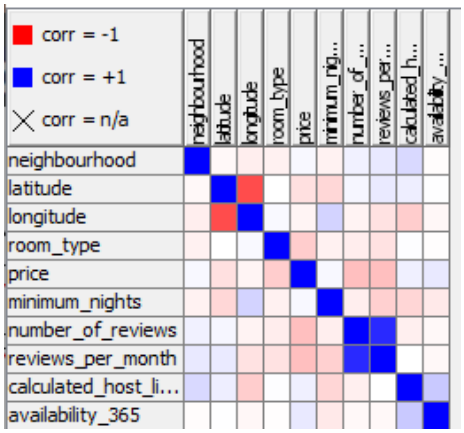


Σε αυτό παράδειγμα έχουμε το μέσο ορό της τιμής ανάλογα το room type

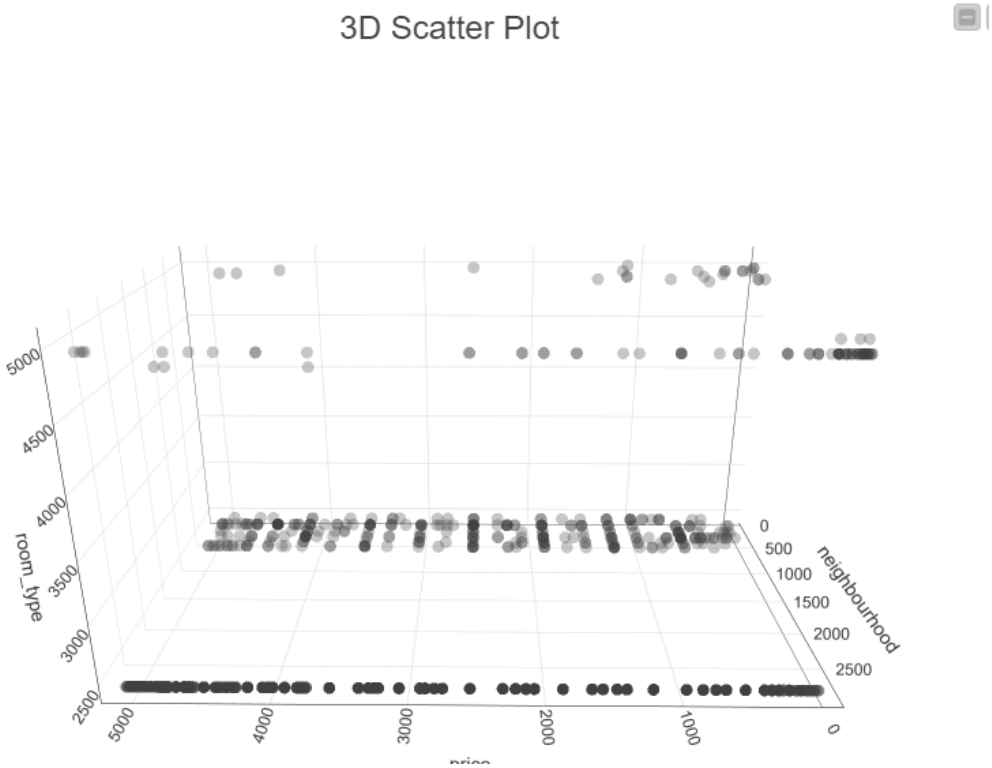


KNIME ΓΙΑ MINING ΚΑΙ GENERALIZATION

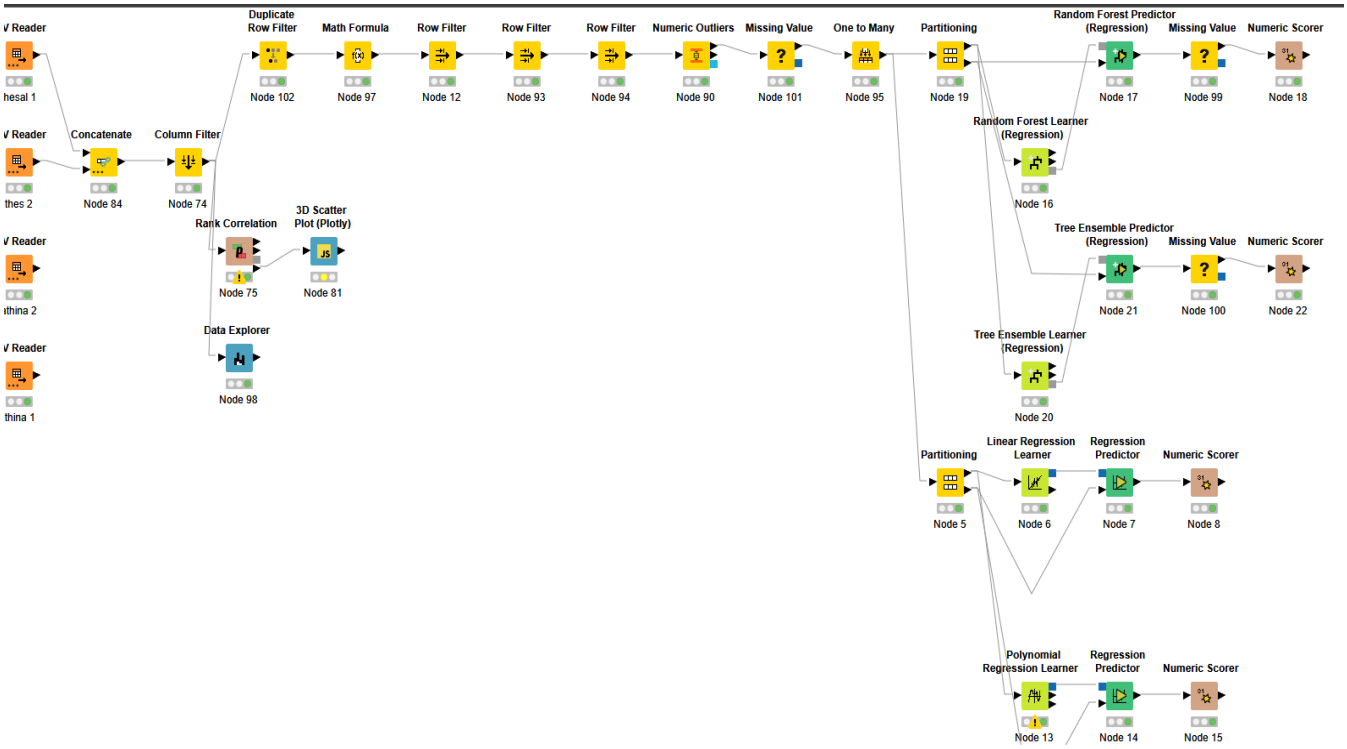
το KNIME είναι εργαλείο που μπορεί να κάνει σχεδόν τα πάντα πάνω σε αυτά που χρειαζόμαστε, το μόνο μειονέκτημα που έχει είναι ότι είναι άσπρο το background



Στην αρχή κάνω ένα rank collation και βλέπω που παρέχει μεγάλο correlession δηλαδή το μπλε χρώμα στην φωτογραφία



Επίσης ένα 3d plot me room type neighborhood και price


v Reader
hesal 1
v Reader
thes 2
Concatenate
Node 84
Column Filter
Node 74
Duplicate
Row Filter
Node 102
Math Formula
Node 97
Row Filter
Node 12
Row Filter
Node 93
Row Filter
Node 94
Numeric Outliers
Node 90
Missing Value
Node 101
One to Many
Node 95
Partitioning
Node 19
Random Forest Predictor
(Regression)
Node 17
Missing Value
Node 99
Numeric Scorer
Node 18
Random Forest Learner
(Regression)
Node 16
v Reader
ithina 2
Rank Correlation
Node 75
3D Scatter
Plot (Plotly)
Node 81
Tree Ensemble Predictor
(Regression)
Node 21
Missing Value
Node 100
Numeric Scorer
Node 22
v Reader
thina 1
Data Explorer
Node 98
Tree Ensemble Learner
(Regression)
Node 20
Partitioning
Node 5
Linear Regression
Learner
Node 6
Regression
Predictor
Node 7
Numeric Scorer
Node 8
Polynomial
Regression Learner
Node 13
Regression
Predictor
Node 14
Numeric Scorer
Node 15

Regression

Για το linear regression kai to polynomial regression και την random forest predictor(regressor) για να βγει το σωστό αποτέλεσμα πρέπει να υπάρχει μια προ επεξεργασία στα δεδομένα

1. Το column filter αφαιρεί τα στοιχεία που δεν χρειάζομαι(id,host id)
2. Το duplicate ελέγχει άμα υπάρχουν ιδιά στοιχεία
3. Το math formula προσθέτει
4. $availability_365$*$calculated_host_listings_count και αυτό μας δίνει λίγο πιο καλά αποτελέσματα, αυτό τον πολλαπλάσιο το βρήκα από το rank correlation που έκανα πιο πριν
5. Στα row filter βγάζω μερικές τιμές διότι είναι outliers πιο συγκεκριμένα βγάζω τις τιμές price πάνω από 2000 επίσης τα availability που είναι κάτω από 10 μέρες και τα minimum nights που είναι πάνω από 100
6. Το numeric outliers κάνει και αυτό detect τα outliers
7. Έπειτα προσθέτω τα missing values
8. Έπειτα κάνω one -hot (one to many στο Knime)

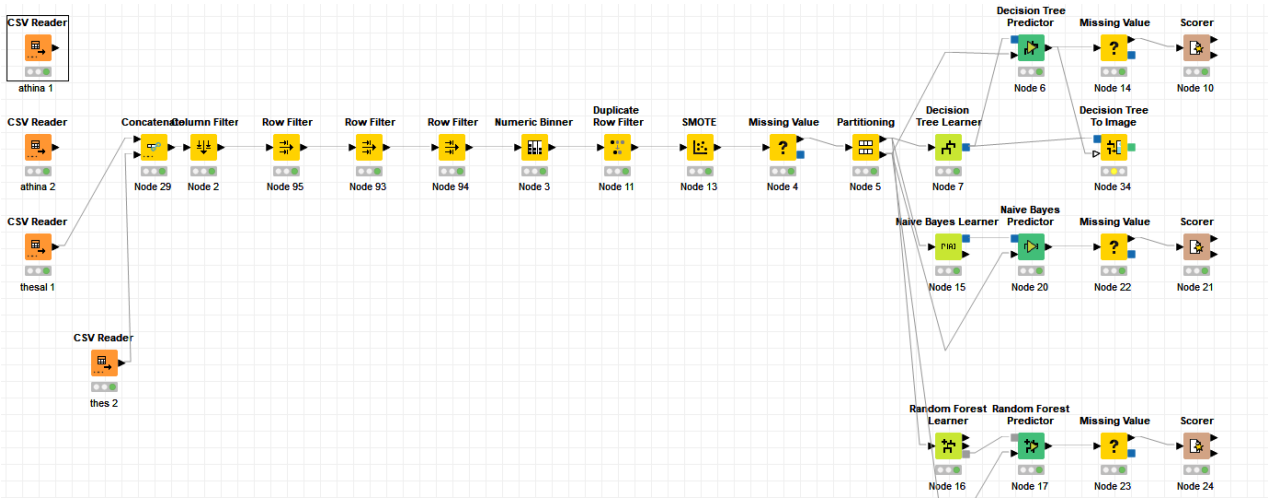
Και τέλος περνώ τα αποτελέσματα σε Αθηνά

	R^2	Mean absolute error	Mean squared error	Root mean squared error	Mean signed difference	Mean absolute error
Linear regression	0,185	17,519	542,11	23,283	-0.165	0,401
Polynomial regression(3)	0,214	17,128	522,756	22,864	-0,257	0,69
Random forest predictor regression	0,317	16,159	486,018	22,048	-0,718	0,353
Tree ensemble predictor	0,375	15,376	444,626	21,086	-0,572	0,332

Και Θεσσαλονίκη

	R^2	Mean absolute error	Mean squared error	Root mean squared error	Mean signed difference	Mean absolute error
Linear regression	0,106	12,636	255,919	15,997	1,583	0,346
Polynomial regression(3)	0,147	12,272	243,975	15,62	1,321	0,334
Random forest predictor regression	0,346	10,599	196,686	14,024	1,225	0,284
Tree ensemble predictor	0,470	9,337	159,609	12,634	0,947	0,247

CLASSIFICATION



Αυτό που αλλάζει σε αυτό το διάγραμμα σε συγκριθεί με το προηγούμενο είναι

1. Το numeric binner που χωρίζει το price σε υποκατηγορίες
2. Το smote κάνει oversampled τα δεδομένα και χρησιμοποίει Nearest neighbor αν και αργεί λίγο να εκτελεστεί έχει μια διαφορά στα αποτελέσματα τα τελικά
3. Στο τέλος είναι όλοι learner μαζί με τα σκορ τους

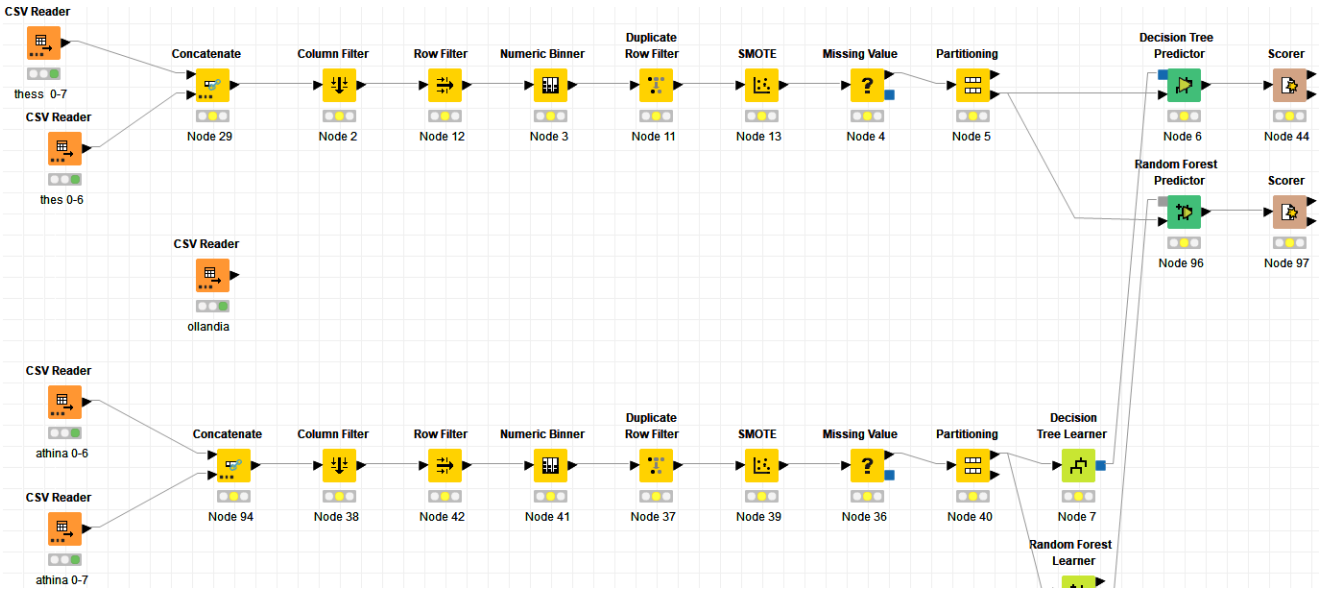
Αθηνά σε όλα τα δεδομένα decision tree και random forest

	Correct classified	Wrong classified	Accuracy	Error	Cohen kappa(k)
decision tree prediction	3861	1701	69,41%	30,58%	0,47
random forest predictor	4967	595	89,30%	10,69%	0,818
naïve bayes	3125	2437	56,18%	43,81%	0,25

Θεσσαλονίκη σε όλα τα δεδομένα decision tree και random forest

	Correct classified	Wrong classified	Accuracy	Error	Cohen kappa(k)
decision tree prediction	1215	347	77,78%	22,21%	0,50
random forest predictor	1459	103	93,406	6,59%	0,86
naïve bayes	917	645	58,707	41,293	0,19

generalization



Αρχικά έχουμε τα δεδομένα τοποθετώ για train την Αθηνά και για τεστ την Θεσσαλονίκη

Τα nodes είναι σχεδόν ιδιά με τα προηγούμενα, και εδώ αφαιρώ τα price που είναι επάνω από 2000 και επίσης αφαιρώ τα χαμηλά availability

Έτσι περνώ τα παρακάτω αποτελέσματα

	Correct classified	Wrong Classified	Accuracy	Error	Cohen kappa(k)
Decision tree learner	1465	2565	36,35%	63,64%	0,029
Random forest Learner	2644	1386	65,60	34,39	0,127

Έπειτα βάζω στους ιδίους node αλλά αλλάζω τα δεδομένα βάζω Ιούνιο Αθηνά και Θεσσαλονίκη μαζί και Ιούλιο Αθηνά και Θεσσαλονίκη μαζί

	Correct classified	Wrong Classified	Accuracy	Error	Cohen kappa(k)
Decision tree learner	6512	3400	65,69%	34,30%	0,363
Random forest Learner	7815	2127	78,54%	21,45%	0,599

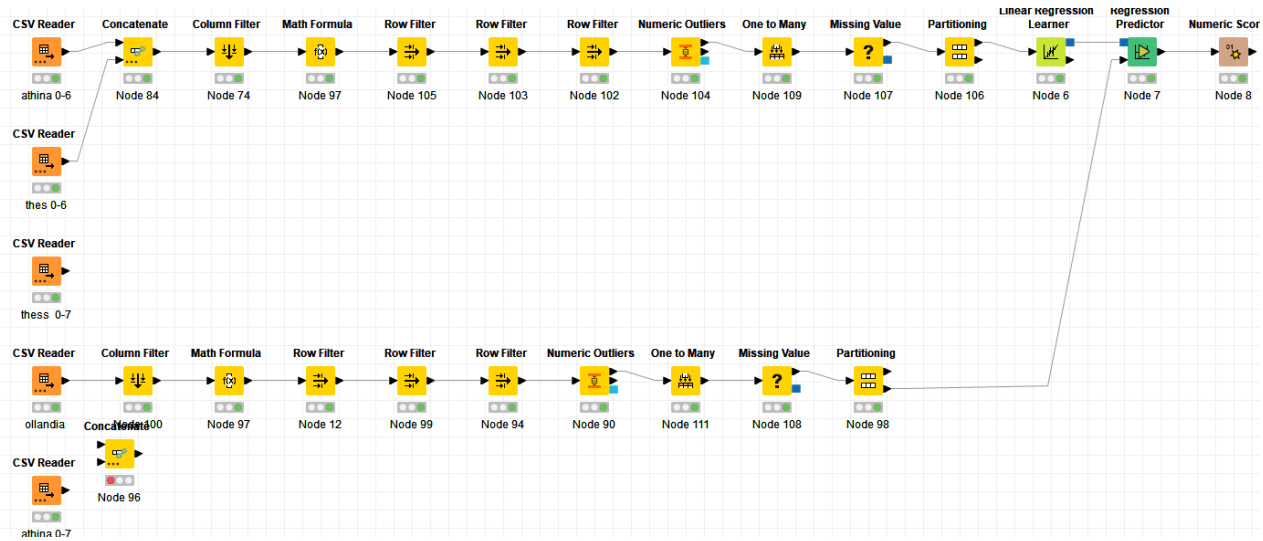
Ο random forest όπως ήταν αναμμένο είναι πιο καλός αλγόριθμος Το τελευταίο generation που έκανα είναι αναμεσά στην Αθηνά και Θεσσαλονίκη το μηνά Ιούνιο και τα δεδομένα του AMSTERDAM(TEST)

	Correct classified	Wrong Classified	Accuracy	Error	Cohen kappa(k)	
Decision tree learner	2218	10858	16,96%	89,08	0,19	
Random forest Learner	874	12.202	6,68	93,31%	-0,07	

Εδώ βλέπουμε ότι ο Decision tree learner είναι πιο αποδοτικός αλγόριθμος

Επρεπε καπου να k fold cross validation όμως δεν τα καταφερα με το knime

Regression και generalization



Κάνουμε ότι και στο regression πιο πριν όμως εδώ δεν έχουμε τα ιδιά ποσοστά.

απλός έχουμε 2 φορές τα ιδιά node γιατί έχουμε το generalization.

Generalization train δεδομένα Αθήνας test σε Θεσσαλονίκης

	R2	Mean absolute error	Mean squared error	Root mean squared error	Mean signed difference	Mean absolute error
Linear regression	-0.096	15,93	372,67	19,305	6,66	0,45

ΙΟΥΝΙΟΣ VS ΙΟΥΛΙΟΣ

	R2	Mean absolute error	Mean squared error	Root mean squared error	Mean signed difference	Mean absolute error
Linear regression	0,044	18,529	619,714	24,894	-1,708	0,425

ΙΟΥΝΙΟΣ(ΑΘΗΝΑ ΚΑΙ ΘΕΣΣ) VS AMSTERDAM(test)

	R2	Mean absolute error	Mean squared error	Root mean squared error	Mean signed difference	Mean absolute error
Linear regression	-1.529	96,911	14,894.207	122.024	-96.308	0.598