|  |  |
| --- | --- |
|  | ****Πανεπιστήμιο Πειραιώς, Τμήμα Πληροφορικής  Προπτυχιακό Πρόγραμμα Σπουδών****  ****Ακαδ. έτος 2020-21 (εαρ. εξάµηνο)**** |

|  |
| --- |
| **[Αναλυτική Δεδομένων]** |
| [Technical Report]  Τζιόρβας Αντώνης – Π18153 |
|  |

Table of Contents

[ΜΕΡΟΣ Α 2](#_Toc74594041)

[Προπαρασκευή 2](#_Toc74594042)

[**PCA** (**P**rincipal **C**omponent **A**nalysis) 5](#_Toc74594045)

[**LDA (L**inear **D**iscriminant **A**nalysis**)** 6](#_Toc74594048)

[Clustering 7](#_Toc74594049)

[**KMeans Clustering** 7](#_Toc74594050)

[**DBSCAN Clustering** 7](#_Toc74594051)

[**Agglomerative Clustering** 8](#_Toc74594053)

[**Mean Shift Clustering** 8](#_Toc74594054)

[Classification 9](#_Toc74594055)

[ΜΕΡΟΣ Β 11](#_Toc74594057)

[Προπαρασκευή 11](#_Toc74594058)

[**Exponential Smoothing** 11](#_Toc74594059)

[**Stationarity Check (Augmented Dickey-Fuller)** 12](#_Toc74594060)

[Timeseries Forecasting 13](#_Toc74594061)

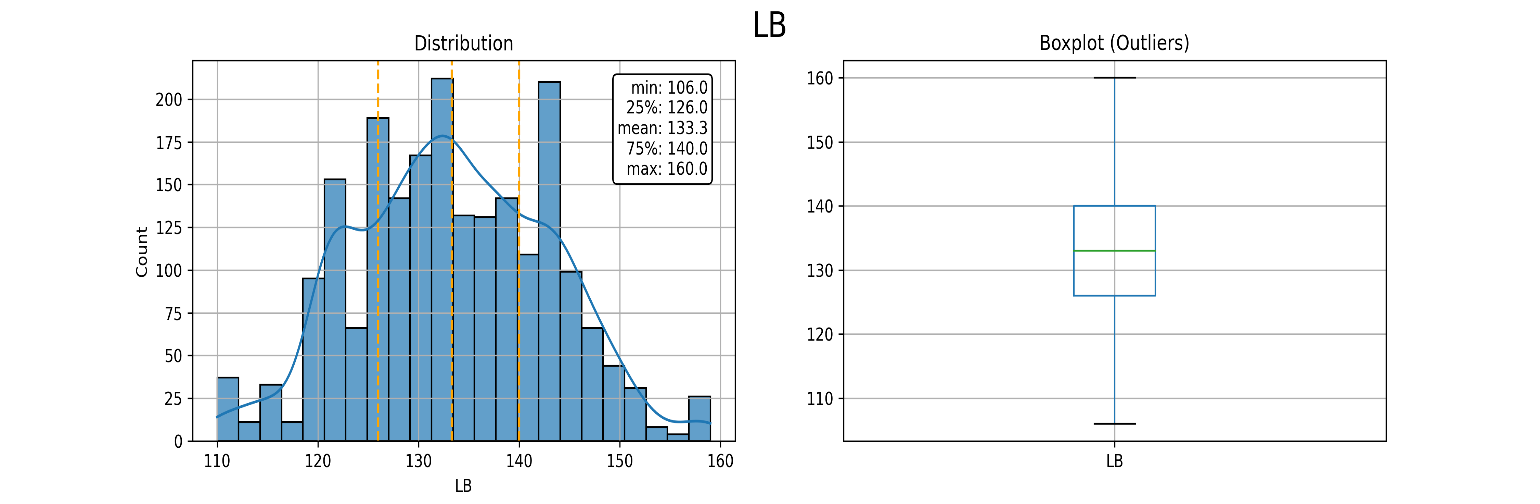
[**MLP Approach** 13](#_Toc74594062)

[**RNN Approach** 14](#_Toc74594064)

[Βιβλιογραφία 15](#_Toc74594066)

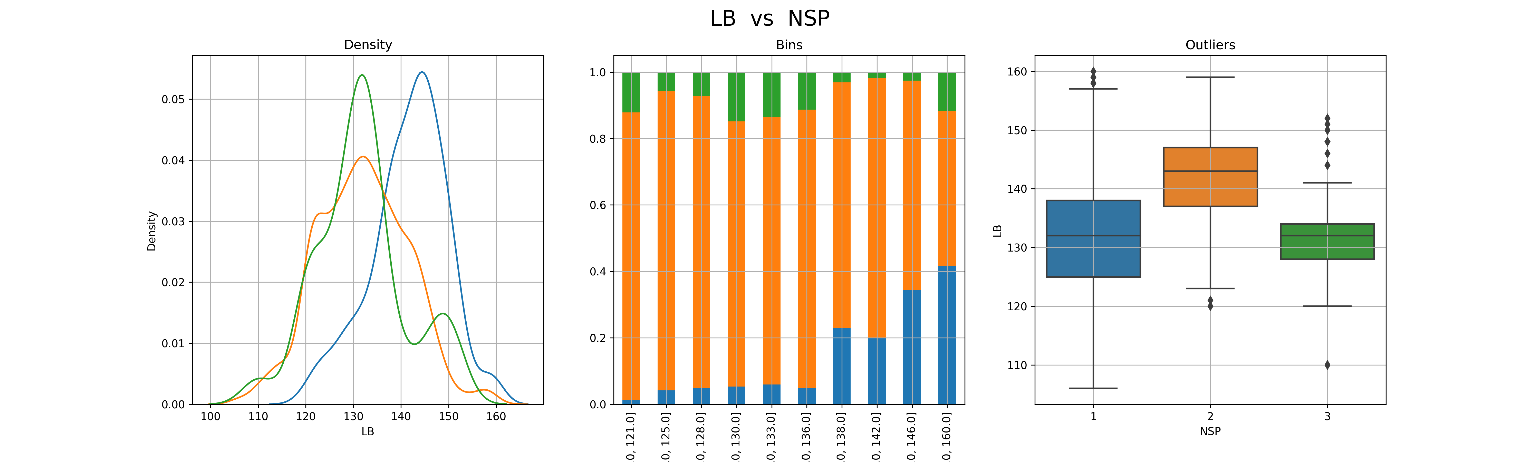
## ΠΛΗΡΕΣ ΦΩΤΟΓΡΑΦΙΚΟ ΥΛΙΚΟ ΕΜΠΕΡΙΕΧΕΤΑΙ ΣΕ ΞΕΧΩΡΙΣΤΟ ΦΑΚΕΛΟ ΣΥΜΠΕΡΙΛΑΜΒΑΝΟΜΕΝΟΥ ΚΑΙ ΤΩΝ ΕΝΔΕΙΚΤΙΚΩΝ ΠΙΝΑΚΩΝ ΤΟΥ REPORT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Αναλυτική Δεδομένων]  [Technical Report] ΜΕΡΟΣ Α To project αυτό αποτελείται από τρία (3) επιμέρους τμήματα που αφορούν την Προπαρασκευή, Συσταδοποίηση και Κατηγοριοποίηση των δεδομένων μας. Προπαρασκευή Πρωταρχικός στόχος είναι να αποκτήσουμε μια γενική εικόνα του συνόλου των δεδομένων μας, καθώς επίσης, να αναγνωρίσουμε τον τύπο κάθε χαρακτηριστικού (κατηγορικό ή αριθμητικό) και να ελέγξουμε για τυχόν τιμές που λείπουν.  Στη συνέχεια μελετάμε την κατανομή (συχνότητα εμφάνισης) των ετικετών ***CLASS*** και ***NSP*** (Target Classes) για να διαπιστώσουμε αν το dataset είναι biased (αν δηλαδή έχουμε περίπου το ίδιο πλήθος εγγραφών για κάθε κλάση)  Μέσω ενός συνδυασμού Ιστογράμματος και Boxplot, θα μελετήσουμε την πυκνότητα της κατανομής του κάθε χαρακτηριστικού και θα εντοπίσουμε πιθανά outliers.  Ενδεικτικά παρατίθεται το χαρακτηριστικό ***LB*** |  | **Ποσοστά Συμμετοχής Κλάσεων**  • • • Type sidebar title • • •   |  |  | | --- | --- | | NSP | Frequency (%) | | Normal | **77.85** | | Suspect | **13.88** | | Pathologic | **8.28** |  |  |  | | --- | --- | | CLASS | Frequency (%) | | A | **27.73** | | B | **18.06** | | C | **15.62** | | D | **11.85** | | E | **9.27** | | AD | **5.03** | | DE | **3.81** | | LD | **3.39** | | FS | **3.25** | | SUSP | **2.49** | |



Συνεχίζοντας την έρευνα γύρω από το χαρακτηριστικό ***LB,*** θα εξετάσουμε το ενδεχόμενο να μπορούμε να εξάγουμε κάποιο συμπέρασμα σχετικά με την κατηγορία ***NSP*** ενός δείγματος με τρεις (3) διαφορετικούς τύπους γραφημάτων:

1. Συγκρίνουμε τις πυκνότητες κατανομής των τριών (3) κατηγοριών και αν παρατηρηθεί κάποια διαφορά, τότε τα δείγματα έχουν διαφορετική συμπεριφορά και συνεπώς μπορούμε μέσω του χαρακτηριστικού αυτού να κάνουμε την πρόβλεψή μας.
2. Χωρίζουμε το σύνολο τιμών του χαρακτηριστικού ***LB*** σε υποσύνολα, εμφανίζουμε τη σύστασή τους και εξετάζουμε εάν οι αναλογίες των δειγμάτων ως προς την ***NSP*** είναι παρόμοια σε όλα τα υποσύνολα.
3. Εμφανίζουμε τα boxplots των τριών (3) κατηγοριών και προσπαθούμε να παρατηρήσουμε κάποια διαφορά στην «συμπεριφορά» των outliers



Και στα τρία γραφήματα παρατηρούμε ότι η συμπεριφορά των δειγμάτων παρουσιάζει μικρές διαφορές, οπότε μπορούμε να θεωρήσουμε σωστή την υπόθεσή μας.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Προτού προχωρήσουμε σε κάποια τεχνική μείωσης διαστάσεων (SVD, PCA, LDA, t-SNE) ας ερευνήσουμε τη συσχέτιση μεταξύ των χαρακτηριστικών μας με τη βοήθεια ενός πίνακα συσχέτισης προκειμένου να «πετάξουμε» πιθανά περιττά χαρακτηριστικά. Ενδιαφερόμαστε στα χαρακτηριστικά που έχουν δείκτη συσχέτισης **Pearson** > 0.8 , γεγονός που δηλώνει ότι τα χαρακτηριστικά είναι στενά συνδεδεμένα μεταξύ τους. Στην περίπτωσή μας τα ζεύγη ***Mean – Median****,* ***Mode – Median****,* ***Min – Width***και ***Mode – Mean***έχουν δείκτη σχεδόν 0.9  Το γεγονός αυτό μας οδηγεί στο συμπέρασμα ότι κάποια από τα χαρακτηριστικά αυτά είναι πλεονάζοντα. Αυτό το ανακαλύπτουμε μέσα από το παρακάτω γράφημα που αναπαριστά το «ποσοστό συμμετοχής» κάθε χαρακτηριστικού στην πλήρη απεικόνιση της πληροφορίας που έχουμε. Η σημασία των χαρακτηριστικών υπολογίζεται από το πόσο κάθε χαρακτηριστικό μειώνει την εντροπία σε ένα δέντρο απόφασης[[1]](#footnote-1).    Με βάση λοιπόν το διάγραμμα μπορούμε να διαπιστώσουμε ότι:   1. Το χαρακτηριστικό ***DS*** δεν φαίνεται να συμβάλλει καθόλου[[2]](#footnote-2) 2. Το χαρακτηριστικό ***Mean*** φαίνεται να είναι μεγαλύτερης σημασίας από τα ***Mode*** και ***Median.*** 3. Το χαρακτηριστικό ***Width*** είναι περιττό καθώς προκύπτει από τη διαφορά   Συνεπώς μπορούμε να «πετάξουμε» τα ***Mode, Median*** και ***Width*** και να διατηρήσουμε το μεγαλύτερο ποσοστό της πληροφορίας. |  | |  |  |  | | --- | --- | --- | | Pearson Correlation | | | | Mean | **Median** | **0.948251** | | Mode | **Median** | **0.933399** | | Min | **Width** | **0.898519** | | Mode | **Mean** | **0.893412** |   Πίνακας 1: Pearson Correlation Scores > 0.8   |  |  | | --- | --- | | VARIABLE | IMPORTANCE | | ASTV | **0.160141** | | ALTV | **0.116019** | | MSTV | **0.094484** | | Μean | **0.085415** | | AC | **0.080397** | | Mode | **0.050176** | | Median | **0.048388** | | DP | **0.046151** | | UC | **0.044288** | | MLTV | **0.040598** | | Min | **0.037856** | | Variance | **0.036667** | | LB | **0.036473** | | Width | **0.031534** | | Max | **0.029131** | | FM | **0.023265** | | Nmax | **0.014867** | | DL | **0.014729** | | Tendency | **0.005998** | | Nzeros | **0.003177** | | DS | **0.000245** |   Πίνακας 2: Feature Importance  **Side Notes**  • • • Type sidebar title • • • |

|  |  |  |
| --- | --- | --- |
| Αφού απομακρύναμε τις πλεονάζουσες στήλες, μπορούμε να προχωρήσουμε σε μείωση διαστάσεων με τη χρήση κάποιου αλγορίθμου. Πρώτα πρέπει να κανονικοποιήσουμε τα δεδομένα μας με τη χρήση του [StandardScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html).  Στην περίπτωσή μας θα συγκρίνουμε την απόδοση δυο (2) αλγορίθμων. Για να συγκρίνουμε την απόδοση ακολουθούμε την παρακάτω διαδικασία και για τους δυο (2) αλγορίθμους:   1. Χωρίζουμε το dataset σε training / testing set. 2. Εκτελούμε τον εκάστοτε αλγόριθμο στο training set. 3. Με τη χρήση του Random Forest Classifier πάνω στο testing set κάνουμε τις προβλέψεις μας. 4. Εμφανίζουμε τις μετρικές αξιολόγησης και το confusion matrix.  **PCA** (**P**rincipal **C**omponent **A**nalysis) Αρχικά αναζητούμε το ελάχιστο πλήθος των κύριων συνιστωσών που θα διατηρήσουν το 95% της πληροφορίας.    Από το παραπάνω γράφημα φαίνεται ότι χρειάζονται δώδεκα (12) κύριες συνιστώσες. Στη συνέχεια εκτελούμε PCA στο training set. Για να αξιολογηθεί η ακρίβεια μελετώνται οι μετρικές από το classification report στο testing set, το οποίο παρατίθεται παρακάτω. |  | **TN / True Negative:** when a case was negative and predicted negative  **TP / True Positive:** when a case was positive and predicted positive  **FN / False Negative:** when a case was positive but predicted negative  **FP / False Positive:** when a case was negative but predicted positive  **Precision – Accuracy of positive predictions.**  **Recall – Fraction of positives that were correctly identified**. **F1 score – What percent of positive predictions were correct?** **Classification Metrics**  • • • Type sidebar title • • • |

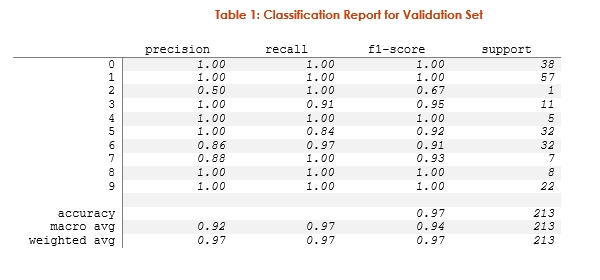
|  |
| --- |
| Πίνακας 3: PCA Classification Report **LDA (L**inear **D**iscriminant **A**nalysis**)** Σε αντίθεση με την PCA, κατά τον μετασχηματισμό του training set χρειάζεται να προσδιορίσουμε και τον πίνακα με τα target labels. Από εκεί και πέρα η διαδικασία αξιολόγησης είναι η ίδια με την PCA και τα αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.    Πίνακας 4: LDA Classification Report  Καθώς η LDA δέχεται και τα targets, απαιτούνται λιγότεροι συντελεστές για να επιτευχθεί η ίδια ακρίβεια. Η ικανότητα πρόβλεψης των δυο μεθόδων παρουσιάζεται στα παρακάτω confusion matrices. |

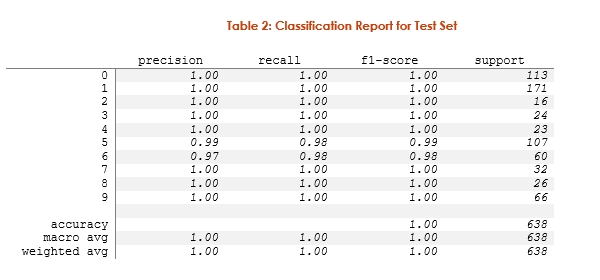
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Όπως φαίνεται από τα confusion matrices, η ακρίβεια των δύο προσεγγίσεων είναι παρόμοια (γύρω στο 89%). Ωστόσο η LDA χρειάστηκε μόνο δύο (2) συντελεστές και για το λόγο αυτό προτιμάται. Clustering Με το νέο dataset τώρα μπορούμε να προχωρήσουμε στη διαδικασία της συσταδοποίησης. Θα προσεγγίσουμε το θέμα αυτό υλοποιώντας τέσσερις (4) διαφορετικούς αλγορίθμους:  KMeans, DBSCAN, Agglomerative Clustering και Mean Shift.  Μετά την εκτέλεση κάθε αλγορίθμου υπολογίζουμε τον αριθμό των clusters που δημιουργήθηκαν για λόγους διασταύρωσης. **KMeans Clustering** Αρχικά πρέπει να αποφασίσουμε τον αριθμό των cluster θα χρησιμοποιηθούν. Θα συγκρίνουμε την ικανότητα του αλγορίθμου να διαχωρίσει τα δεδομένα σε 3, 4, 5 και 10 cluster *(βλ. Πίνακα 5)*. H αξιολόγηση θα γίνει με βάση το Sihlouette Score[[3]](#footnote-3).  Παρατηρείται ότι το καλύτερο score επιτυγχάνεται με 3 clusters. **DBSCAN Clustering** Ο DBSCAN δέχεται αρκετές παραμέτρους, 2 από τις οποίες είναι εξαιρετικά σημαντικές καθώς αλλάζουν ριζικά τη συμπεριφορά του. Οι σημαντικότερες είναι το ***eps*** και το ***min\_samples.***  Το ***eps*** αναπαριστά τη μέγιστη απόσταση στην οποία ένα σημείο επιλέγει τους γείτονές του. Αυτό θα καθορίσει και πόσα cluster θα δημιουργηθούν και συνεπώς ποια σημεία θα θεωρηθούν ως «θόρυβος». Καθώς το eps είναι ανάλογο του αναμενόμενου αριθμού γειτόνων που ανακαλύπτονται, θα χρησιμοποιήσουμε το αλγόριθμο KNN προκειμένου να πετύχουμε όσο το δυνατό ακριβέστερη εκτίμηση για το eps. |  | |  |  | | --- | --- | | # Clusters | Sihlouette Score | | 3 | **0.51** | | 4 | **0.39** | | 5 | **0.41** | | 10 | **0.34** |   Πίνακας 5: KMeans Sihlouette Score   |  |  | | --- | --- | | Algorithm | Sihlouette Score | | KMeans | **0.51** | | DBSCAN | **0.22** | | Agglomerative | **0.48** | | Μean Shift | **0.53** |   Πίνακας 6: Clustering Scores  **Side Notes**  • • • Type sidebar title • • • |

Η βέλτιστη τιμή για το eps θα βρεθεί στο σημείο μέγιστης καμπυλότητας[[4]](#footnote-4).

|  |
| --- |
| Το min\_samples αναπαριστά τον ελάχιστο αριθμό δειγμάτων σε μια γειτονιά προκειμένου ένα σημείο να θεωρηθεί ως πυρήνας. Είναι μια παράμετρος που χρειάζεται δοκιμές προκειμένου να βρεθεί η «βέλτιστη» τιμή. Μετά τo «κούρδισμα» των παραμέτρων μπορούμε να προχωρήσουμε στην εφαρμογή του αλγορίθμου. **Agglomerative Clustering** Είναι μια από τις συχνότερες εφαρμογές Ιεραρχικής Συσταδοποίησης και χρησιμοποιείται για να ομαδοποιήσει τα δεδομένα βάσει ομοιότητας. Όπως και με τον KMeans, προσδιορίζουμε τον αριθμό των clusters. Μία προαιρετική παράμετρος είναι το linkage. Όπως αναφέρεται στη βιβλιογραφία του sklearn:  The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.   * ‘ward’ minimizes the variance of the clusters being merged. * ‘average’ uses the average of the distances of each observation of the two sets. * ‘complete’ or ‘maximum’ linkage uses the maximum distances between all observations of the two sets. * ‘single’ uses the minimum of the distances between all observations of the two sets.   Ύστερα από πειραματισμό με τα διαφορετικές τιμές, αποφάσισα να διατηρήσω την default τιμή ***ward****.* **Mean Shift Clustering** Αρχικά υπολογίζουμε το εύρος (bandwidth) που θα χρησιμοποιήσει για το clustering ο αλγόριθμος και του το περνάμε ως παράμετρο.  Αφού υλοποιήσαμε όλους τους αλγορίθμους, αξιολογούμε τον καθένα ξεχωριστά εμφανίζοντας τα αντίστοιχα Scatter Plot και Sihlouette Plot μαζί με το score. Ενδεικτικά παρουσιάζεται για τον Mean Shift. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classification Χρησιμοποιώντας το dataset στη μορφή που είχε πριν εφαρμόσουμε LDA (2126, 33), θα εκπαιδεύσουμε ένα Νευρωνικό Δίκτυο MLP που θα ταξινομεί τα δεδομένα σε μία από τις δέκα (10) κατηγορίες της στήλης CLASS.  Το πρώτο βήμα είναι να κανονικοποιήσουμε (Normalize)τα δεδομένα μας με τη βοήθεια του MinMaxScaler. Στο παρακάτω γράφημα KDE φαίνεται η κατανομή / πυκνότητα των χαρακτηριστικών πριν (αριστερά) και μετά την κανονικοποίηση (δεξιά) τους.    Στη συνέχεια χωρίζουμε το dataset σε τρία (3) σύνολα   * Training Set με μήκος 60 % του dataset * Validation Set με μήκος 10 % του dataset * Testing Set με μήκος 30 % του dataset   και μπορούμε πλέον να προχωρήσουμε στην κατασκευή του Νευρωνικού Δικτύου. Το δίκτυο αποτελείται από   1. μία (1) στοιβάδα εισόδου, 2. δυο (2) κρυφές στοιβάδες με 30 νευρώνες έκαστος 3. μια (1) στοιβάδα Dropout με ratio 10 % (για να αποφευχθεί overfitting) 4. τη στοιβάδα εξόδου.   Ως loss function χρησιμοποιήθηκε το Categorical Crossentropy ενώ ως optimizer ορίστηκε ο Adam με learning rate 0.001. To ΝΝ εκπαιδεύεται για 30 epochs. Λεπτομέρειες σχετικά με το ΝΝ παρουσιάζονται και στο διπλανό πίνακα ενώ τα Classification Reports φαίνονται στους ακόλουθους πίνακες |  | **Classification  Details**  • • • Type sidebar title • • •   |  |  |  | | --- | --- | --- | | Layer # | Type | Units | | 1 | **Input** | **60** | | 2 | **Dense** | **30** | | 3 | **Dense** | **30** | | 4 | **Dropout** | **-** | | 5 | **Dense** | **10** |   Πίνακας 7: Sequential Model Build  **Input Dimension = 23**  **Learning Rate = 0.001**  **Output Classes = 10**  **Epochs = 30**  **Batch Size = 1**  **Dropout Rate = 10%**   |  |  | | --- | --- | | Metric | Value | | Accuracy | **96.71%** | | Loss | **0.079** | | F1-Score | **0.968** |   Πίνακας 8: Validation Set Scores   |  |  | | --- | --- | | Metric | Value | | Accuracy | **99.53%** | | Loss | **0.017** | | F1-Score | **0.995** |   Πίνακας 9: Test Set Scores |





|  |
| --- |
| ΜΕΡΟΣ Β To project αυτό αποτελείται από τρία (3) επιμέρους τμήματα που αφορούν την Προπαρασκευή, Κατασκευή MLP για πρόβλεψη χρονοσειρών και Κατασκευή RNN για πρόβλεψη χρονοσειρών. Επιλέγουμε να ασχοληθούμε με την πρόβλεψη της τιμής ***“ISE\_USD\_BASED”.*** Προπαρασκευή Προτού ξεκινήσουμε την προετοιμασία των δεδομένων μας, θα μελετήσουμε τη συμπεριφορά τους στο χρόνο. Συγκεκριμένα θα ασχοληθούμε με τη μεταβολή της μέσης τιμής και διακύμανσης στο πέρασμα του χρόνου.   **Exponential Smoothing** Μία πρώτη απόπειρα πρόβλεψης μπορεί να πραγματοποιηθεί μέσω της τεχνικής *exponential smoothing.* Βασιζόμενη στην υπόθεση ότι το μέλλον θα είναι **περίπου** ίδιο με το πρόσφατο παρελθόν - και συνεπώς όσο πιο πίσω στο χρόνο πάμε, τόσο λιγότερη επιρροή έχει στην πρόβλεψή μας - μπορούμε να αποκτήσουμε μία πρώτη ιδέα για την συμπεριφορά των δεδομένων μας. Ο μαθηματικός τύπος μπορεί να εκφραστεί ως: |

* είναι το ποσοστό «βαρύτητας» που δίνει το μοντέλο στην προηγούμενη πρόβλεψη
* είναι η προηγούμενη πρόβλεψη επί την βαρύτητα
* αναπαριστά το πόσο θυμάται το μοντέλο από την προηγούμενη πρόβλεψη

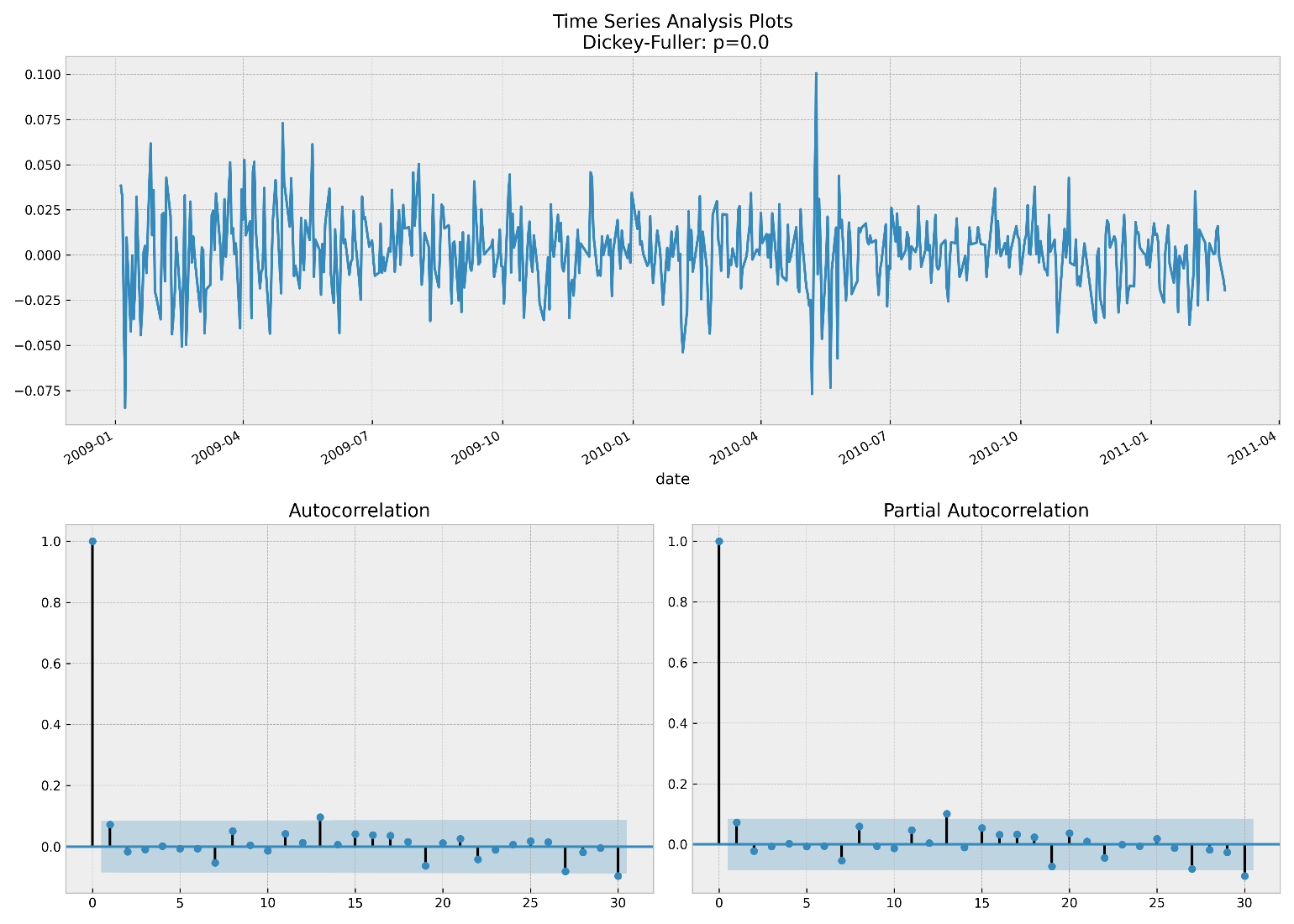
### 

### **Stationarity Check (Augmented Dickey-Fuller)**

Στη συνέχεια ελέγχουμε τη «στατικότητα» (**stationarity**) των δεδομένων μας. Για να θεωρηθεί μια χρονοσειρά «στατική» πρέπει να έχει:

1. Σταθερό μέσο
2. Σταθερή διακύμανση
3. Σταθερή συνδιακύμανση μεταξύ περιόδους ίδιου μήκους

Ο έλεγχος αυτός μπορεί εύκολα να πραγματοποιηθεί μέσω του τεστ ***Dickey-Fuller***.



Όπως βλέπουμε το *p-value* είναι 0 και αυτό σημαίνει ότι η χρονοσειρά μας είναι στατική, οπότε μπορούμε να προχωρήσουμε στην υλοποίηση του Νευρωνικού Δικτύου.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timeseries Forecasting Προτού τροφοδοτήσουμε το νευρωνικό δίκτυο με τα δεδομένα μας, μετασχηματίζουμε τη χρονοσειρά σε supervised πρόβλημα. Το αρχικό dataset αποτελούνταν μόνο από τη στήλη ***ISE\_USD\_BASED.*** Αυτό σημαίνει ότι η τιμή ***ISE\_USD\_BASED*** σε μία χρονική στιγμή ***t*** προβλέπεται από την τιμή ***ISE\_USD\_BASED*** στις χρονικές στιγμές ***t-1, t-2, t-3 και t-4.*** **MLP Approach** Αφού μετατρέψαμε τη χρονοσειρά, χωρίζουμε το dataset σε training και testing set με αναλογίες 80/20 %. Το MLP μας έχει την ακόλουθη δομή:   1. Ένα (1) Input layer με 5 νευρώνες 2. Δυο (2) Dense layers με 2 νευρώνες το καθένα 3. Ένα (1) Output layer   Ως optimizer ορίζουμε τον *Adam με learning rate 0.0014,* ενώ για συνάρτηση σφάλματος χρησιμοποιούμε το μέσο τετραγωνικό σφάλμα.  Εκπαιδεύουμε το Νευρωνικό Δίκτυο αλλάζοντας των αριθμό των epochs[[5]](#footnote-5) κάθε φορά και συγκρίνουμε την ακρίβειά του μέσω της μετρικής r2-score. Παρατηρούμε ότι το καλύτερο score επιτυγχάνεται στα 25 epochs και από εκεί και πέρα μειώνεται ξανά το score. |  | **MLP  Details**  • • • Type sidebar title • • •   |  |  |  | | --- | --- | --- | | Layer # | Type | Units | | 1 | **Input** | **5** | | 2 | **Dense** | **2** | | 3 | **Dense** | **2** | | 4 | **Dense** | **1** |   Πίνακας 7: MLP Neural Network Build   |  |  | | --- | --- | | Parameter | Value | | Input Dimension | **4** | | Learning Rate | **0.0014** | | No. Epochs | **25** | | Batch Size | **1** |   Πίνακας 8: Best Model Parameters  **R2-Score: 0.003** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RNN Approach** Παρόμοια με το MLP, μετασχηματίζουμε τη χρονοσειρά σε supervised πρόβλημα. Ωστόσο, σε αυτή την περίπτωση χρειάζεται να προσδιορίσουμε τον αριθμό των εγγραφών που θα χρησιμοποιηθούν για training (samples), τον αριθμό  Στη συνέχεια κατασκευάζουμε ένα RNN με:   1. Ένα (1) SimpleRNN layer με 5 νευρώνες 2. Δυο (2) Dense layers με 3 νευρώνες το καθένα 3. Ένα (1) Output layer   Ως optimizer ορίζουμε τον *Adam με learning rate 0.0014,* ενώ για συνάρτηση σφάλματος χρησιμοποιούμε το μέσο τετραγωνικό σφάλμα.  Όπως και πριν, εκπαιδεύουμε το Νευρωνικό Δίκτυο αλλάζοντας των αριθμό των epochs[[6]](#footnote-6) κάθε φορά και συγκρίνουμε την ακρίβειά του μέσω της μετρικής r2-score. Παρατηρούμε ότι το καλύτερο score επιτυγχάνεται στα 30 epochs. |  | **SimpleRNN  Details**  • • • Type sidebar title • • •   |  |  |  | | --- | --- | --- | | Layer # | Type | Units | | 1 | **SimpleRNN** | **5** | | 2 | **Dense** | **3** | | 3 | **Dense** | **3** | | 4 | **Dense** | **1** |   Πίνακας 9: RNN Neural Network Build   |  |  | | --- | --- | | Parameter | Value | | Input Shape |  | | Learning Rate | **0.002** | | No. Epochs | **25** | | Batch Size | **1** |   Πίνακας 10: Best Model Parameters  **R2-Score: -0.016** |

## Βιβλιογραφία

Mallawaarachchi, Vijini. “How to Use DBSCAN Effectively.” *Medium*, 21 June 2020, towardsdatascience.com/how-to-use-dbscan-effectively-ed212c02e62.

Peixeiro, Marco. “The Complete Guide to Time Series Analysis and Forecasting.” *Medium*, 22 May 2021, towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775?gi=f0a4f6f6cc67.

Pietro, Mauro Di. “Machine Learning with Python: Classification (Complete Tutorial).” *Medium*, 8 Dec. 2020, towardsdatascience.com/machine-learning-with-python-classification-complete-tutorial-d2c99dc524ec?gi=e29e80af774f.

Rajaratne, Maneesha. “Data Pre Processing Techniques You Should Know.” *Medium*, 2 Dec. 2018, towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6.

Vandeput, Nicolas. “Simple Exponential Smoothing.” *Medium*, 1 Apr. 2021, towardsdatascience.com/simple-exponential-smoothing-749fc5631bed.

‌

‌

1. Στην περίπτωσή μας χρησιμοποιείται ο [Random Forest Classifier](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) [↑](#footnote-ref-1)
2. Βλ. «Πίνακας 2: Feature Importance» για το ακριβές νούμερο [↑](#footnote-ref-2)
3. Silhouette Coefficient ή Silhouette Score είναι μια μετρική που χρησιμοποιείται για τον υπολογισμό της ικανότητας ενός αλγορίθμου συσταδοποίησης που κυμαίνεται στο σύνολο [-1, 1]. Όσο πιο κοντά στο 1, τόσο καλύτερη είναι η συσταδοποίηση. [↑](#footnote-ref-3)
4. Δεδομένου ότι γνωρίζουμε πως τα δεδομένα μας χωρίζονται σε 3 cluster, θα προτιμηθεί η τιμή του eps που δίνει 3 cluster ανεξάρτητα από το αν είναι η βέλτιστη ή όχι. [↑](#footnote-ref-4)
5. Δοκιμάστηκαν οι τιμές 20, 25, 50 και 100 [↑](#footnote-ref-5)
6. Δοκιμάστηκαν οι τιμές 20, 25, 30 και 50 [↑](#footnote-ref-6)