
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ 1984
ΚΕΡΚΥΡΑ
ΚΕΦΑΛΛΟΝΙΑ
ΖΑΚΥΝΘΟΣ
ΛΕΥΚΑΣ
ΙΘΑΚΗ
ΚΥΘΗΡΑ
ΠΑΞΟΙ

Αναφορά Εργασίας στη Διαχείριση Μεγάλου Όγκου Δεδομένων

Κωνσταντίνος Καφτεράνης inf2021090
Στέργιος Μουτζίκος inf2021149
Χρήστος Κωστάκης inf2021115

Ιανουάριος 2025

Περίληψη

Σε αυτήν την αναφορά περιγράφεται η ανάλυση δεδομένων καιρού, περιλαμβάνοντας τον καθαρισμό δεδομένων, την οπτικοποίηση και την εφαρμογή αλγορίθμων μηχανικής μάθησης για την πρόβλεψη του καιρού. Το έγγραφο αποτελεί ένα σύντομο οδηγό για την ανάλυση, με χρήση της Python και του PySpark.

1 Απαιτήσεις και Εισαγωγές

Για την ανάλυση χρησιμοποιήθηκαν οι ακόλουθες βιβλιοθήκες Python:

- NumPy
- Pandas
- Matplotlib
- Seaborn
- Scipy
- Umap
- Scikit-learn
- PySpark

2 Περίληψη Δεδομένων

Το σύνολο δεδομένων που αναλύθηκε αφορά την πρόβλεψη βροχοπτώσεων στην Αυστραλία. Αποτελείται από 23 στήλες (χαρακτηριστικά) και πάνω από 145.000 γραμμές δεδομένων. Περιλαμβάνει πληροφορίες όπως τη θερμοκρασία, την υγρασία, την πίεση και το αν υπήρξε βροχή την επόμενη ημέρα.

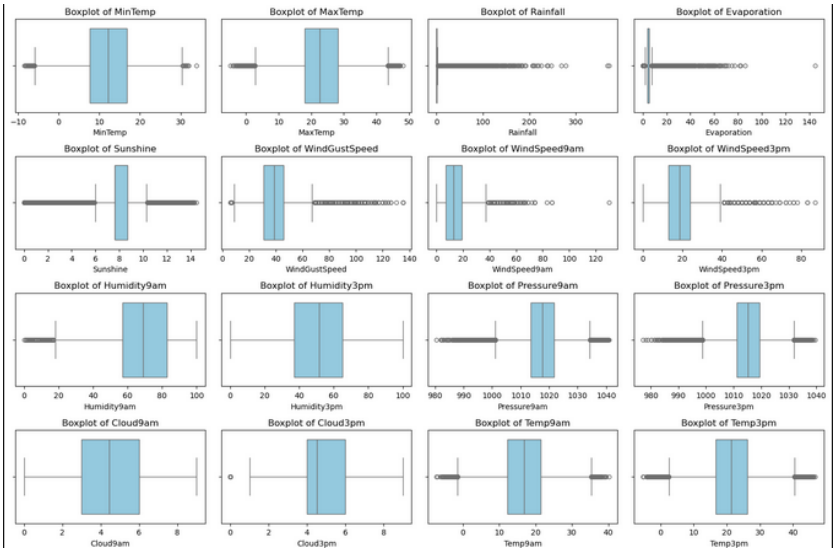

Figure 1: Boxplots

2.1 Missing Values και Outliers

Η επεξεργασία των missing values στο σύνολο δεδομένων απαιτούσε τον διαχωρισμό μεταξύ κατηγορηματικών και αριθμητικών στηλών. Οι αριθμητικές στήλες περιείχαν χαρακτηριστικά όπως θερμοκρασία, υγρασία και πίεση, ενώ οι κατηγορηματικές περιλάμβαναν την τοποθεσία και την κατεύθυνση του ανέμου. Για την αντικατάσταση missing values εφαρμόστηκαν τα εξής:

- Για αριθμητικές στήλες, οι missing values αντικαταστάθηκαν με τον μέσο όρο της στήλης. Αυτό επιλέχθηκε για να διατηρηθεί η συνολική κατανομή των αριθμητικών δεδομένων χωρίς να εισαχθούν μεγάλες αποκλίσεις.

- Για κατηγορηματικές στήλες, αντικαταστάθηκαν με την πιο συχνή τιμή mode της στήλης. Αυτό βοήθησε στη διατήρηση της συνέπειας στις κατηγορίες χωρίς να δημιουργηθούν νέες ή μη ρεαλιστικές τιμές.

Επιπλέον, αντιμετωπίστηκαν outliers, χρησιμοποιώντας την τεχνική Z-score capping. Με αυτή τη μέθοδο, τιμές που ξεπερνούσαν συγκεκριμένα όρια αντι-

καταστάθηκαν με τις μέγιστες ή ελάχιστες αποδεκτές τιμές, διατηρώντας έτσι τη συνοχή των δεδομένων.

2.2 Outliers

Η διαδικασία εντοπισμού και διαχείρισης των εκτός ορίων τιμών περιελάμβανε τα εξής στάδια:

- **Φόρτωση ενημερωμένου αρχείου δεδομένων**: Το ενημερωμένο αρχείο, όπου είχαν αντικατασταθεί οι ελλιπείς τιμές, φορτώθηκε σε ένα DataFrame χρησιμοποιώντας τη βιβλιοθήκη Pandas. Αν το αρχείο δεν βρισκόταν, εμφανιζόταν μήνυμα σφάλματος και το DataFrame οριζόταν ως None.

- **Εμφάνιση πλήθους δεδομένων και επιλογή αριθμητικών στηλών**: Ο συνολικός αριθμός στοιχείων στο DataFrame υπολογίστηκε, ενώ επιλέχθηκαν μόνο οι αριθμητικές στήλες (float64, int64) για περαιτέρω ανάλυση.

- **Δημιουργία boxplots**: Δημιουργήθηκαν διαγράμματα boxplot, που βοηθούν στην αναγνώριση της κατανομής των δεδομένων και εντοπίζουν εύκολα ακραίες τιμές.

- **Υπολογισμός εκτός ορίων τιμών με βάση το Z-score**: Ο υπολογισμός του Z-score βοήθησε στον εντοπισμό των outliers (δεδομένα με Z-score μεγαλύτερο του 3 σε απόλυτη τιμή). Καταγράφηκε ο αριθμός αυτών των τιμών για κάθε στήλη.

- **Διόρθωση εκτός ορίων τιμών (capping)**: Τα ανώτερα και κατώτερα όρια αποδεκτών τιμών υπολογίστηκαν, και οι τιμές που ξεπερνούσαν τα όρια διορθώθηκαν αντίστοιχα. Για παράδειγμα, μια εξαιρετικά υψηλή θερμοκρασία μειώθηκε στο μέγιστο αποδεκτό όριο.

- **Ενημέρωση στατιστικών**: Μετά τη διόρθωση, τα περιγραφικά στατιστικά (μέσος όρος, τυπική απόκλιση, ελάχιστη και μέγιστη τιμή) επικαιροποιήθηκαν για τις αριθμητικές στήλες.

3 Οπτικοποίηση Δεδομένων

Οι οπτικοποιήσεις ήταν κρίσιμες για την κατανόηση των δεδομένων. Δημιουργήθηκαν τα παρακάτω γραφήματα:

- **Θερμοκρασία**: Διαγράμματα για τη μέση θερμοκρασία ανά ώρα της ημέρας, που έδειξαν διακυμάνσεις κατά τη διάρκεια της ημέρας.
- **Κατανομή Βροχής**: Ιστογράμματα που παρουσίασαν την κατανομή της βροχόπτωσης στις τοποθεσίες.
- **Χάρτης Συσχέτισης**: Διαγράμματα συσχέτισης για την κατανόηση της σχέσης μεταξύ διαφορετικών μεταβλητών, όπως η θερμοκρασία και η υγρασία.

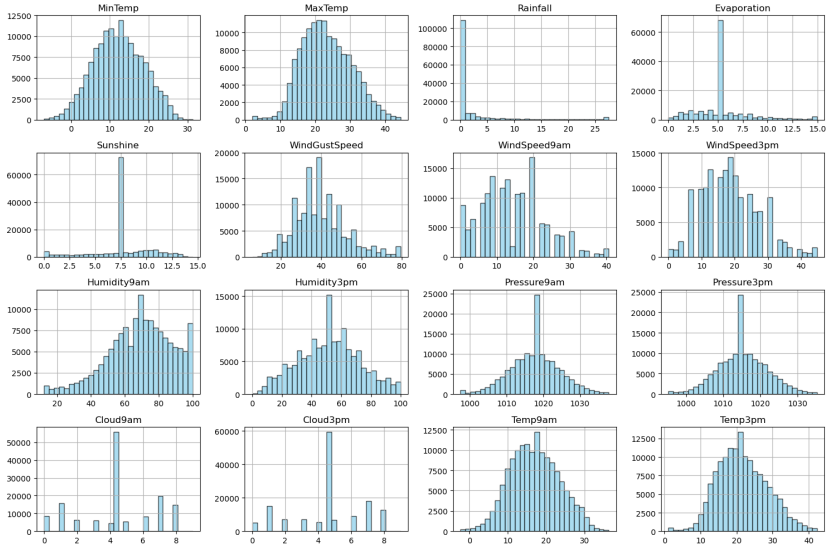

Figure 2: Histogram

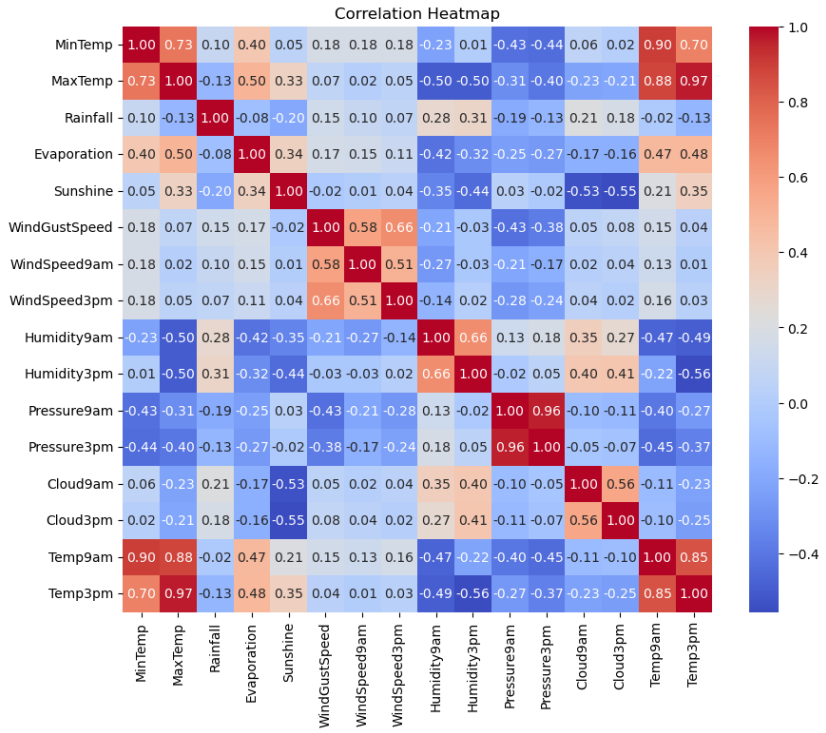

Figure 3: Heatmap

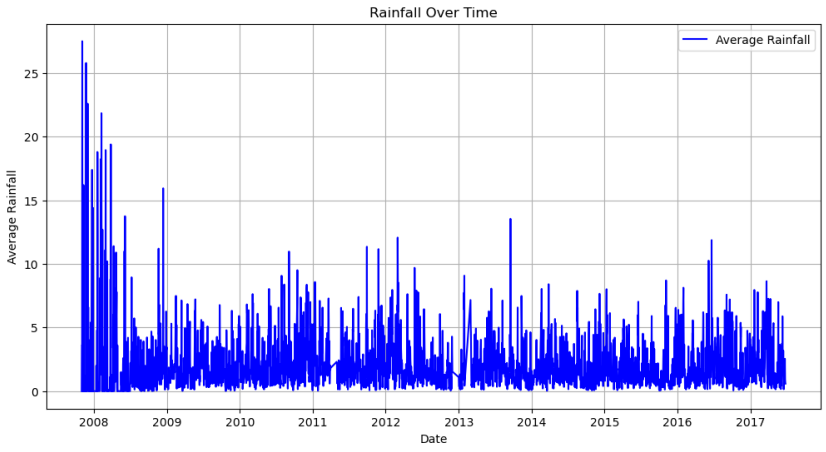

Figure 4: Rain Over Time

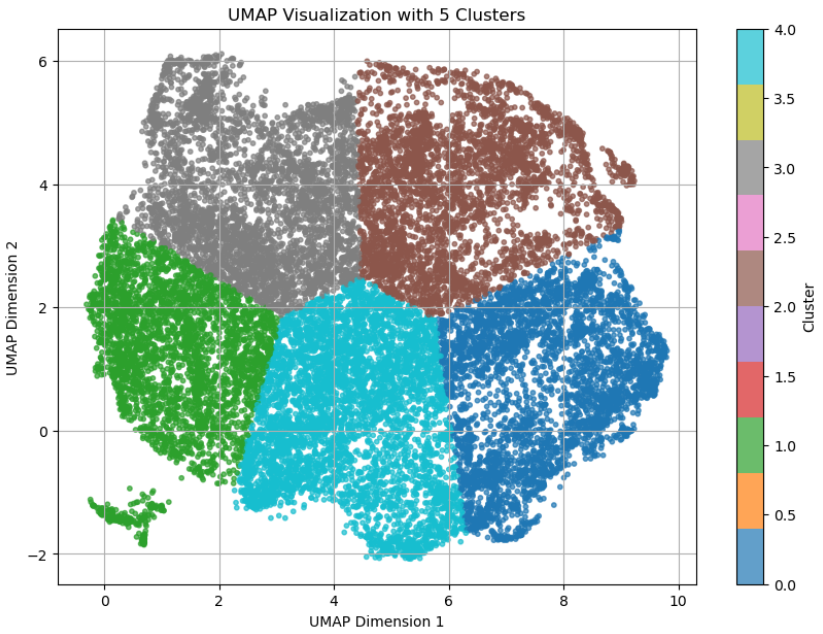

Figure 5: Umap

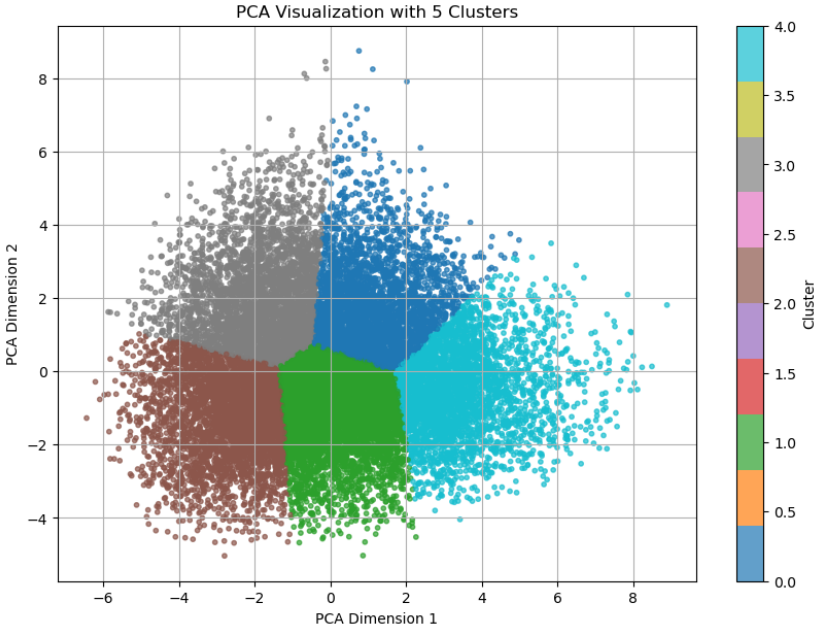

Figure 6: PCA

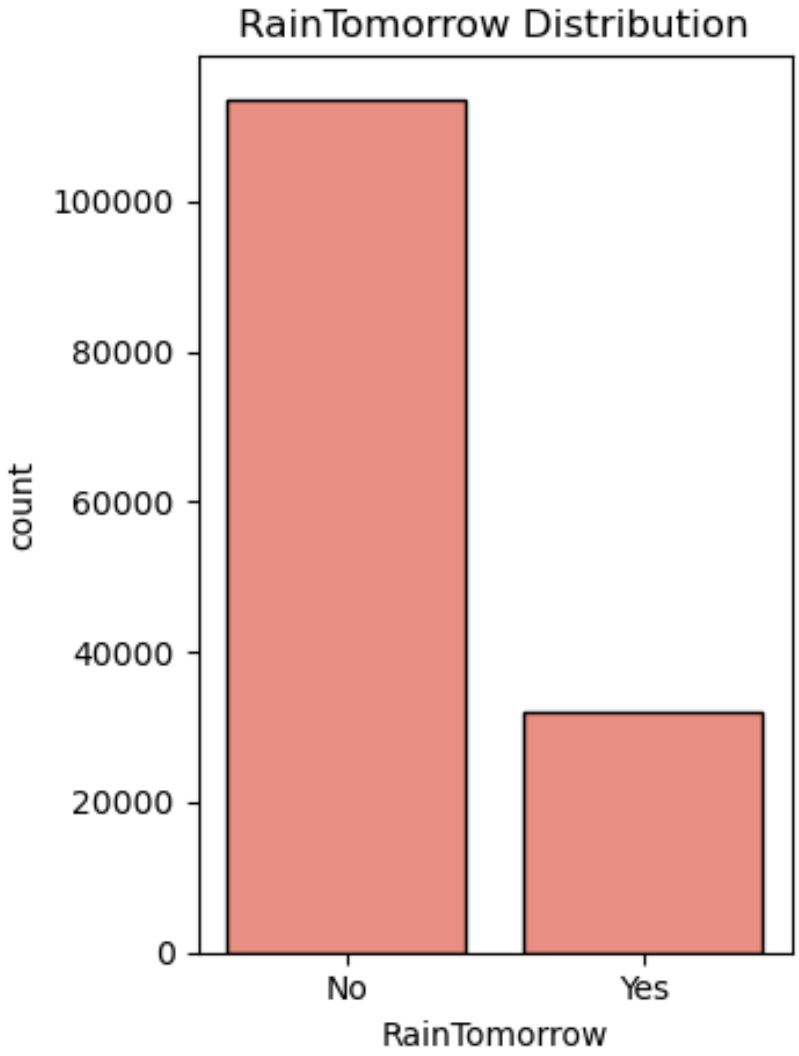

Figure 7: Rain Tomorrow

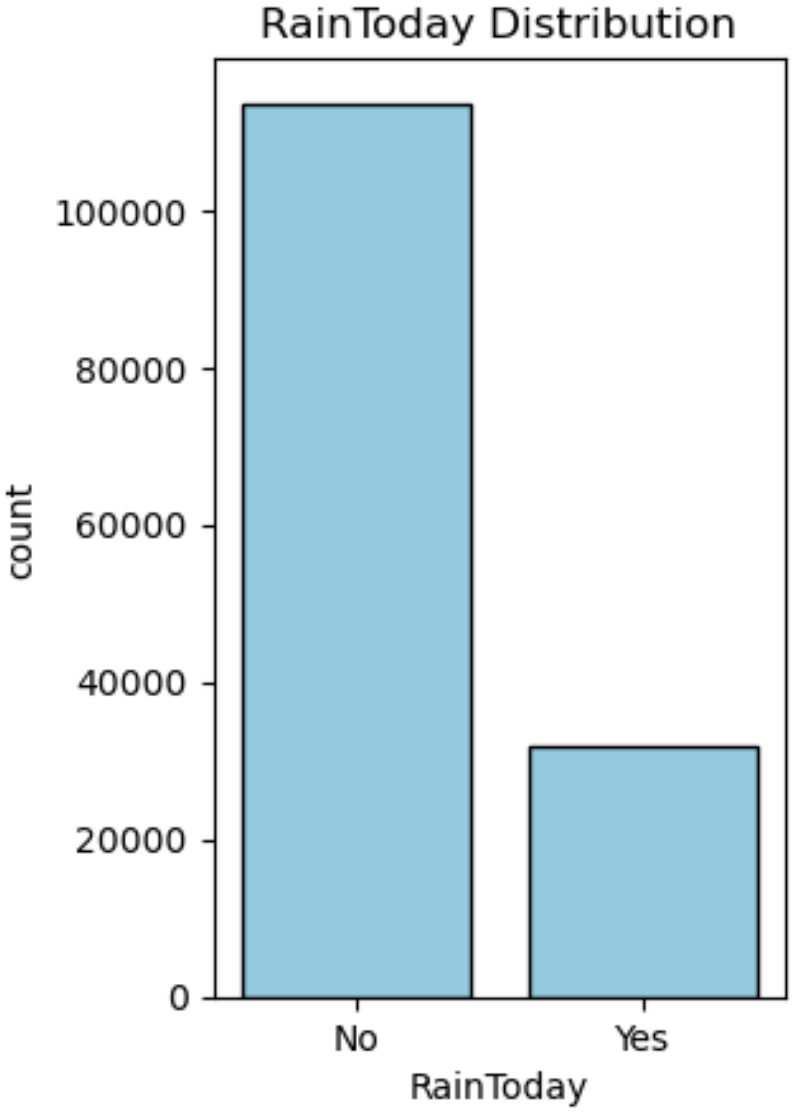

Figure 8: Rain Today

4 Μηχανική Μάθηση

Η ανάλυση περιλάμβανε την εφαρμογή αλγορίθμων μηχανικής μάθησης για την πρόβλεψη του αν θα υπάρξει βροχή την επόμενη ημέρα. Ο αλγόριθμος Logistic Regression εφαρμόστηκε μέσω των παρακάτω σταδίων:

4.1 Logistic Regression

Ο αλγόριθμος Logistic Regression χρησιμοποιήθηκε για την ταξινόμηση, όπου η πρόβλεψη ήταν δυαδική (βροχή ή όχι βροχή). Οι βασικές ενέργειες περιλάμβαναν:

- **Εκκίνηση Spark Session**: Αρχικοποιήθηκε μια συνεδρία Spark, ώστε να επιταχυνθούν οι υπολογισμοί.
- **Μετατροπή σε PySpark DataFrame**: Το Pandas DataFrame μετατράπηκε σε PySpark DataFrame για επεξεργασία.
- **Κωδικοποίηση κατηγορηματικών μεταβλητών**: Οι κατηγορηματικές μεταβλητές, όπως οι τοποθεσίες και η κατεύθυνση του ανέμου, κωδικοποιήθηκαν ως αριθμοί χρησιμοποιώντας το String Indexer.
- **Συγκέντρωση χαρακτηριστικών**: Οι μεταβλητές, όπως η θερμοκρασία και η υγρασία, συγκεντρώθηκαν σε έναν πίνακα χαρακτηριστικών (features vector) με τη βοήθεια του Vector Assembler.
- **Κωδικοποίηση στόχου**: Η στήλη "RainTomorrow" κωδικοποιήθηκε σε 0 (όχι βροχή) και 1 (βροχή).
- **Διαχωρισμός συνόλου δεδομένων**: Τα δεδομένα χωρίστηκαν σε εκπαιδευτικό και δοκιμαστικό σύνολο.
- **Εκπαίδευση μοντέλου**: Το μοντέλο Logistic Regression εκπαιδεύτηκε στο εκπαιδευτικό σύνολο δεδομένων.
- **Πρόβλεψη**: Το μοντέλο έκανε προβλέψεις για το δοκιμαστικό σύνολο, κατηγοριοποιώντας τις ημέρες σε 'Ναι' ή 'Οχι' για βροχή.
- **Αξιολόγηση μοντέλου**: Η απόδοση του μοντέλου αξιολογήθηκε με χρήση του Area Under Curve (AUC), το οποίο έδειξε εξαιρετική ακρίβεια 0.85.

4.2 Random Forest

Ο αλγόριθμος Random Forest εφαρμόστηκε για να βελτιωθεί η ακρίβεια της πρόβλεψης. Οι βασικές ενέργειες περιλάμβαναν:

- **Εκκίνηση Spark Session**: Όπως και με τη λογιστική παλινδρόμηση, ξεκίνησε μια συνεδρία Spark για επεξεργασία.

- **Δημιουργία δέντρων απόφασης**: Ο Random Forest δημιούργησε πολλαπλά δέντρα απόφασης σε τυχαία υποσύνολα δεδομένων.
- **Συνδυασμός αποτελεσμάτων**: Τα αποτελέσματα των δέντρων συνδυάστηκαν για να βελτιώσουν την ακρίβεια της πρόβλεψης.
- **Αξιολόγηση μοντέλου**: Ο AUC για τον Random Forest ήταν 0.66.

4.3 Gradient Boosted Trees (GBT)

Ο αλγόριθμος Gradient Boosted Trees χρησιμοποιεί σταδιακή εκμάθηση για βελτιστοποίηση. Οι βασικές ενέργειες περιλάμβαναν:

- **Εκκίνηση Spark Session**: Η επεξεργασία έγινε στο Spark, όπως και για τους υπόλοιπους αλγορίθμους.
- **Δημιουργία σταδίου δέντρων**: Κάθε νέο δέντρο προστίθεται για να διορθώσει σφάλματα του προηγούμενου.
- **Αξιολόγηση μοντέλου**: Ο AUC του GBT ήταν 0.70, καλύτερος από τον Random Forest.