Κοκκοκύρης Στέλιος - 3160063

Μαυραγάνης Βασίλης - 3160091

Τοσκολλάρι Ρόναλντ – 3160244

ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΑΠΟ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

- Αρχικά εισάγουμε τις απαραίτητες βιβλιοθήκες και διαβάζουμε τα train δεδομένα από το train.csv αρχείο. Στην συνέχεια, πραγματοποιούμε ορισμένους ελέγχους όπως την ύπαρξη null τιμών στο train dataset μας και τους τύπους δεδομένων των features. Όσον αφορά τους τύπους των features πραγματοποιούμε σε ορισμένους από αυτούς (season,year,month,hour,holiday,weekday,workingday,weekday) την αλλαγή από numerical σε categorical ώστε να μας διευκολύνει στην επεξεργασία των δεδομένων (π.χ. στο one hot encoding πιο μετά) και συνεπώς στην παραγωγή καλύτερης ακρίβειας.

- Ένα από τα πιο σημαντικά πράγματα που έπρεπε να κάνουμε ώστε να βελτιώσουμε την ακρίβεια ήταν να απομακρύνουμε τους outliers από το υπόδειγμά μας. Από στατιστικής πλευράς outlier είναι μία παρατήρηση η οποία διαφέρει από τις άλλες. Σε αυτό το σημείο πρέπει να αναφέρουμε ότι κάποιοι από τους outliers είναι επιθυμητοί. Επιθυμούμε να κρατήσουμε αυτούς που δεν είναι από λάθος κατά την συλλογή των δεδομένων. Αρχικά, πρέπει να εντοπίσουμε τους outliers και στον κώδικά μας αυτό το έχουμε πετύχει χρησιμοποιώντας την boxplot όπου από την γραφική παράσταση μπορούμε να εντοπίσουμε τα δεδομένα που δεν ακολουθούν την κατανομή. Όπως παρατηρείτε, έχουμε outliers στα features humidity και windspeed. Αντίθετα, τα features temp και atemp δεν έχουν outliers. Αφού εντοπίσουμε τους outliers χρησιμοποιούμε τη συνάρτηση zscore για να τους απομακρύνουμε. Κατά τον υπολογισμό του z-score η συνάρτηση κάνει rescale και center τα δεδομένα ψάχνοντας τα στοιχεία που απέχουν αρκετά από το 0. Στην εργασία, έπειτα από αρκετές δοκιμές καταλήξαμε ότι όσα στοιχεία έχουν z-score από 3 και πάνω θα αντιμετωπίζονται ως outliers. Ακόμα, αφαιρούμε τους outliers μόνο από το feature humidity και όχι από το windspeed, διότι αφαιρώντας τους outliers του δεύτερου στοιχείου είχαμε χειρότερη πρόβλεψη. Αφού διώξαμε τους outliers, χρησιμοποιήσαμε την countplot για να τσεκάρουμε αν υπάρχουν άκυρα numerical δεδομένα στο υπόδειγμά μας. Παρατηρούμε ότι το windspeed έχει πολλές τιμές 0, οι οποίες μπορεί να είναι NaN, outliers ή σωστές. Επίσης, κάνουμε ανάλυση των

categorical features (season, year, month, hour, holiday, weekday, workingday, weather) με την barplot και βλέπουμε ότι καθένα από αυτά έχει επίδραση στην μεταβλητή count. Το τελευταίο διάγραμμα που φτιάχνουμε πριν δοκιμάσουμε τα μοντέλα δείχνει τη συσχέτιση (corellation) μεταξύ των numerical features. Όπως παρατηρούμε: 1) τα temp και atemp έχουν πολύ μεγάλη συσχέτιση, 2) τα casual και registered περιέχουν άμεσες πληροφορίες για την count (data leakage). Γι' αυτό δεν τις λαμβάνουμε υπ' όψιν στο feature set. 3)Η temp έχει θετική συσχέτιση με την count, ενώ η humidity αρνητική. 4) Τέλος, η windspeed έχει μικρή συσχέτιση με την count.

Model 1 (Linear Regression)

Αρχικά διαγράφουμε τις στήλες casual, registered. Έπειτα, στον άξονα των X εκχωρούμε όλες τις εναπομείναντες στήλες εκτός της count, ενώ στον άξονα των Y εκχωρούμε την στήλη count.Στη συνέχεια, εφαρμόζουμε τον αλγόριθμο Linear Regression ως μία πρώτη προσέγγιση στον πρόβλημά μας. Πριν, όμως, από την εφαρμογή του Linear Regression επιλέγουμε να χωρίσουμε τα δεδομένα μας σε k=10 ομάδες για μεγαλύτερη ακρίβεια πρόβλεψης με τη μέθοδο k-fold cross validation. Έτσι, πέτυχαμε να έχουμε overall accuracy 0.38 και μετά την εφαρμογή του Linear Regression ένα σφάλμα της τάξεως του 1.27852 (RMSLE) και R2: 0.42505. Τέλος, επιλέξαμε να δημιουργήσουμε ένα regression plot για καλύτερη κατανόηση της απόδοσης του μοντέλου αυτού.

Model 2 (Linear Regression με τροποποιημένα δεδομένα)

Στο μοντέλο 2 σκεφτήκαμε να χωρίσουμε τα δεδομένα μας με βάση την κατηγορία των χρηστών. Οπότε, δημιουργήσαμε δύο data frames, ένα για τους registered users και ένα για τους casual users. Για κάθε ένα από αυτά τα data frames αρχικά μετονομάζουμε τη στήλη casual ή registered σε count, εντοπίζουμε και αφαιρούμε τυχόν outliers με τη μέθοδο Z-score και έπειτα εφαρμόζουμε one-hot encoding. Με τη συγκεκριμένη τεχνική μετατρέπουμε τα μη-δυαδικά categorical features σε πολλαπλά υπό-γνωρίσματα δυαδικής μορφής. Έτσι, κάθε υπό-γνώρισμα δείχνει πότε αληθεύει ή όχι. Με αυτό τον τρόπο επιτυγχάνουμε καλύτερη πρόβλεψη για το μοντέλο μας. Στη συνέχεια, για κάθε data frame χωρίζουμε τις στήλες μας όπως ακριβώς στο model 1. Όλες οι στήλες εκτός της count θα προστεθούν στον άξονα των X και η count στον άξονα των Y. Ακολούθως, εφαρμόζουμε Linear Regression σε κάθε ένα από τα data frames, αφού πρώτα έχουμε εφαρμόσει τη μέθοδο k-fold cross validation με k=10. Έτσι, επιτυγχάνουμε overall accuracy 0.62207 για casual users και 0.66176 για registered users. Έχουμε ένα σφάλμα της τάξεως του 0.97988 για casual users και 1.0382 για registered users, ενώ η ακρίβεια είναι 0.66440 και 0.69100 για casual και registered users αντίστοιχα.

Model 3 (Random Forest)

Στο μοντέλο αυτό θα χρησιμοποιήσουμε τον αλγόριθμο Random Forest Regression.Ο Random Forest Regression ανήκει στους ensemble algorithms και μέσω αυτού θα προσπαθήσουμε να επιτύχουμε μικρότερο σφάλμα πρόβλεψης. Σε αυτό το μοντέλο η τεχνική one-hot encoding δεν μας χρειάζεται. Έτσι, αρχικά χωρίζουμε τα δεδομένα μας σε casual και registered users επιτυγχάνοντας με αυτό τον τρόπο μικρότερο σφάλμα πρόβλεψης, όπως φαίνεται και από το μοντέλο 2. Έτσι, για κάθε ένα data frame εντοπίζουμε και αφαιρούμε τα outliers με τη μέθοδο z-score.Έπειτα με τη μέθοδο log1p πραγματοποιούμε κανονικοποίηση στην count.Με αυτόν τον τρόπο έχουμε μεγαλύτερες πιθανότητες για μια πιο ακριβή πρόβλεψη. Τέλος, έχοντας πρώτα χωρίσει τα γνωρίσματα στους άξονες X,Y όπως ακριβώς περιγράφηκε στα προηγούμενα μοντέλα, εφαρμόζουμε τον αλγόριθμο Random Forest Regression. Το σφάλμα για τους casual και τους registered users είναι 0.51041 και 0.3612 αντίστοιχα, ενώ η ακρίβεια R2 είναι 0.84726 για τους casual και 0.92494 για τους registered. Σχεδιάσαμε και τα αντίστοιχα regression plots σχετικά με την απόδοση του μοντέλου αυτού.

Model 4 (Gradient Boost)

Το μοντέλο 4 είναι αυτό το οποίο εν τέλει κρατήσαμε για την πρόβλεψη μας. Σε αυτό το μοντέλο χρησιμοποιούμε τον ensemble αλγόριθμο Gradient Boost. Ακολουθούμε ακριβώς τα ίδια βήματα με το μοντέλο 3. Κρατάμε τα δεδομένα μας σε 2 data frames με βάση την κατηγορία χρήστη, χρησιμοποιούμε τη μέθοδο log1p για την κανονικοποίηση του γνωρίσματος count και τέλος εφαρμόζουμε τον αλγόριθμο Gradient Boost Regressor.Επιτυγχάνουμε τα μικρότερα έως τώρα σφάλματα. Για τους casual users έχουμε RMSLE: 0.47934 και ακρίβεια R2: 0.91058 , ενώ για τους registered users έχουμε RMSLE:0.29276 και R2: 0.95230.Παρακάτω παρατίθενται τα αντίστοιχα performance regression plots.

Model 5 (Neural Network)

Το πέμπτο και τελευταίο μοντέλο που δοκιμάζουμε είναι ένα neural network και συγκεκριμένα το keras από την βιβλιοθήκη tensorflow. Πριν ξεκινήσουμε τη δημιουργία του νευρωνικού δικτύου κάνουμε κάποια προεπεξεργασία στα δεδομένα για να έχουμε καλύτερο fit στο μοντέλο και καλύτερα αποτελέσματα. Η προεπεξεργασία αυτή είναι one-hot encoding στις categorical μεταβλητές, διότι αυτό βοηθάει το νευρωνικό δίκτυο στην εκπαίδευσή του. Ξεκινάμε τώρα τη δημιουργία του νευρωνικού δικτύου. Αρχικά, κάνουμε split τα δεδομένα σε train/test.Μετά κάνουμε ένα νευρωνικό δίκτυο με 3 layers.Στα 2 πρώτα layers ορίζουμε 200 νευρώνες (neurons), ενώ στο 3ο

layers έχουμε μόνο 1 νευρώνα. Ορίζουμε σαν activation function την relu, διότι με αυτήν είχε καλύτερη απόδοση το δίκτυο. Στην συνέχεια, κάνουμε compile το neural network και εκπαιδεύουμε το δίκτυο. Τέλος, υπολογίζουμε τα test δεδομένα, το σφάλμα και την ακρίβεια (RMSLE/R2).

TESTING TEST DATASET

Διαβάζουμε το test dataset και κάνουμε κατάλληλη προεπεξεργασία στα δεδομένα στο μοντέλο μας, το οποίο θα είναι το Gradient Boost Regressor.Τέλος, κάνουμε πρόβλεψη για τους casual/registered users και παίρνουμε το άθροισμά τους για την τελική μας πρόβλεψη. Τέλος, ετοιμάζουμε το αρχείο το οποίο ανεβάσαμε στο Kaggle.

Επιπλέον πράγματα που δοκιμάσαμε για την βελτιώση της πρόβλεψης

1) Lasso Regression, Ridge Regression, XGBoost, KNN - neighbors
2) GridSearchCV για να βρούμε τις καλύτερες παραμέτρους.
3) Να κάνουμε πρόβλεψη χρησιμοποιώντας συνδυασμό των μοντέλων.