Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Διατμηματικό ΠΜΣ στα Προηγμένα Συστήματα Υπολογιστών και Επικοινωνιών
1η εργασία για το μάθημα “Τεχνικές Μηχανικής Μάθησης”
Γεωργιάδης Στέφανος (ΑΕΜ 501)
Επιβλέπων Καθηγητής: Ιωάννης Σαράφης

Πρόβλημα 3

Ερώτημα Α

Ανάλυση του Συνόλου Δεδομένων

Το σύνολο δεδομένων (αρχείο covtype) αποτελείται από 581.011 γραμμές και 55 στήλες. Οι στήλες αντιπροσωπεύουν τα χαρακτηριστικά από διάφορους τύπους βλάστησης, όπου συναντούμε σε 4 δασώδεις περιοχές. Η τελευταία στήλη, δηλώνει τον τύπο βλάστησης του κάθε (φυτού/δέντρου) και μπορεί να πάρει 7 τιμές.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581012 entries, 0 to 581011
Data columns (total 55 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   Elevation                           581012 non-null  int64
 1   Aspect                              581012 non-null  int64
 2   Slope                               581012 non-null  int64
 3   Horizontal_Distance_To_Hydrology    581012 non-null  int64
 4   Vertical_Distance_To_Hydrology      581012 non-null  int64
 5   Horizontal_Distance_To_Roadways     581012 non-null  int64
 6   Hillshade_9am                       581012 non-null  int64
 7   Hillshade_Noon                      581012 non-null  int64
 8   Hillshade_3pm                       581012 non-null  int64
 9   Horizontal_Distance_To_Fire_Points  581012 non-null  int64
 10  Wilderness_Area1                    581012 non-null  int64
 11  Wilderness_Area2                    581012 non-null  int64
 12  Wilderness_Area3                    581012 non-null  int64
 13  Wilderness_Area4                    581012 non-null  int64
 14  Soil_Type1                          581012 non-null  int64
 15  Soil_Type2                          581012 non-null  int64
 16  Soil_Type3                          581012 non-null  int64
 17  Soil_Type4                          581012 non-null  int64
 18  Soil_Type5                          581012 non-null  int64
 19  Soil_Type6                          581012 non-null  int64
 20  Soil_Type7                          581012 non-null  int64
 21  Soil_Type8                          581012 non-null  int64
 22  Soil_Type9                          581012 non-null  int64
 23  Soil_Type10                         581012 non-null  int64
 24  Soil_Type11                         581012 non-null  int64
 25  Soil_Type12                         581012 non-null  int64
 26  Soil_Type13                         581012 non-null  int64
 27  Soil_Type14                         581012 non-null  int64
 28  Soil_Type15                         581012 non-null  int64
 29  Soil_Type16                         581012 non-null  int64
 30  Soil_Type17                         581012 non-null  int64
 31  Soil_Type18                         581012 non-null  int64
 32  Soil_Type19                         581012 non-null  int64
 33  Soil_Type20                         581012 non-null  int64
 34  Soil_Type21                         581012 non-null  int64
 35  Soil_Type22                         581012 non-null  int64
 36  Soil_Type23                         581012 non-null  int64
 37  Soil_Type24                         581012 non-null  int64
 38  Soil_Type25                         581012 non-null  int64
 39  Soil_Type26                         581012 non-null  int64
 40  Soil_Type27                         581012 non-null  int64
 41  Soil_Type28                         581012 non-null  int64
 42  Soil_Type29                         581012 non-null  int64
 43  Soil_Type30                         581012 non-null  int64
 44  Soil_Type31                         581012 non-null  int64
 45  Soil_Type32                         581012 non-null  int64
 46  Soil_Type33                         581012 non-null  int64
 47  Soil_Type34                         581012 non-null  int64
 48  Soil_Type35                         581012 non-null  int64
 49  Soil_Type36                         581012 non-null  int64
 50  Soil_Type37                         581012 non-null  int64
 51  Soil_Type38                         581012 non-null  int64
 52  Soil_Type39                         581012 non-null  int64
 53  Soil_Type40                         581012 non-null  int64
 54  Cover_Type                          581012 non-null  int64
dtypes: int64(55)
```

Σχήμα 1: Οι πληροφορίες για τα χαρακτηριστικά του συνόλου δεδομένων

Οι κατηγορικές μεταβλητές είναι τα χαρακτηριστικά Wilderness Area και Soil Type . Συνολικά υπάρχουν 4 διαφορετικές δασώδεις περιοχές (Wilderness Area) και 40 διαφορετικοί τύποι εδάφους (Soil Type) . Για κάθε τύπο φυτού οι μεταβλητές αυτές δηλώνουν την περιοχή στην οποία βρίσκεται και σε τι τύπο εδάφους έχει φυτρώσει. Η τιμή 1 δηλώνει την παρουσία της συγκεκριμένης περιοχής ή του τύπου εδάφους ενώ η τιμή 0 απουσία. Συνοψίζοντας το σύνολο δεδομένων αποτελείται από 44 κατηγορικές μεταβλητές και 10 μεταβλητές όπου παίρνουν συνεχείς τιμές. Τέλος, η τιμή στόχος παίρνει 7 διαφορετικές τιμές όπου η καθεμία δηλώνει τον τύπο

βλάστησης. Για τον λόγο αυτό το πρόβλημα αυτό ανήκει στα προβλήματα κατηγοριοποίησης. Για παράδειγμα, στο παρακάτω σχήμα φαίνονται οι πρώτες πέντε παρατηρήσεις του συνόλου δεδομένων

	0	1	2	3	4
Elevation	2596	2590	2804	2785	2595
Aspect	51	56	139	155	45
Slope	3	2	9	18	2
Horizontal_Distance_To_Hydrology	258	212	268	242	153
Vertical_Distance_To_Hydrology	0	-6	65	118	-1
Horizontal_Distance_To_Roadways	510	390	3180	3090	391
Hillshade_9am	221	220	234	238	220
Hillshade_Noon	232	235	238	238	234
Hillshade_3pm	148	151	135	122	150
Horizontal_Distance_To_Fire_Points	6279	6225	6121	6211	6172
Cover_Type	5	5	2	2	5
Soil	Type29	Type29	Type12	Type30	Type29
Wilderness	Area1	Area1	Area1	Area1	Area1

Σχήμα2: Τα χαρακτηριστικά για τα 5 πρώτα στοιχεία του συνόλου δεδομένων

Ερώτημα Β

Χωρίζουμε το σύνολο δεδομένων σε σύνολο εκπαίδευσης και σύνολο αξιολόγησης και κατασκευάζουμε ένα μοντέλο λογιστικής παλινδρόμησης χρησιμοποιώντας την συνάρτηση LogisticRegression από την βιβλιοθήκη Sklearn. Ως αλγόριθμο επίλυσης διαλέγουμε τον 'LBFGS', με μέγιστο αριθμό 10000 επαναλήψεων, σύγκλιση στο 10^{-3}, ομαλοποίηση $L2$ και βάρος C = 1.0. Ελέγχουμε την ορθότητα (accuracy) και την ταχύτητα σύγκλισης του μοντέλου, όπως φαίνεται παρακάτω.

```
logreg = LogisticRegression(tol=0.001, solver='lbfgs', verbose=1, C=1.0,max_iter=10000)
logreg.fit(X_train, y_train)
print("Test set accuracy with Logistic Regression before scaling: {:.3f}".format(logreg.score(X_test,y_test)))
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
C:\Users\admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed: 73.9min finished

Test set accuracy with Logistic Regression before scaling: 0.704
```

Παρατηρούμε ότι η ορθότητα του μοντέλου (accuracy) παίρνει την τιμή 0.704 και ο χρόνος όπου απαιτείται για την διεργασία αυτή είναι περίπου 74 λεπτά.

Στη συνέχεια επαναλαμβάνουμε την ίδια διαδικασία χρησιμοποιώντας διαφορετικούς αλγόριθμους επίλυσης της συνάρτησης LogisticRegression για διαφορετικές τιμές των υπερπαραμέτρων. Χρησιμοποιώντας μία συνάρτηση πλέγματος επιλέγουμε τις κατάλληλες υπερπαραμέτρους και τον καλύτερο αλγόριθμο επίλυσης. Η διαδικασία αυτή παρά το γεγονός ότι παρέχει ικανοποιητικά αποτελέσματα, έχει ως μειονέκτημα ότι είναι χρονοβόρα και απαιτεί μεγάλη υπολογιστική ισχύς.

Συμπεραίνουμε ότι ο κατάλληλος αλγόριθμος επίλυσης, για μέγιστο αριθμό επαναλήψεων Max_iter=1.000, του συγκεκριμένου προβλήματος είναι ο “sag” με υπερπαραμέτρους:

- C=11.29
- Tol=0.001
- Max Iter=1.000
- Ομαλοποίηση L2

```
logreg = LogisticRegression(tol=0.001, solver='sag', verbose=1, C=11.29,max_iter=1000)
logreg.fit(X_train, y_train)
print("Test set accuracy with Logistic Regression before scaling: {:.3f}".format(logreg.score(X_test,y_test)))
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
convergence after 311 epochs took 193 seconds
Test set accuracy with Logistic Regression before scaling: 0.693
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:  3.2min finished
```

Ερώτημα Γ

Εισάγουμε την συνάρτηση LinearDiscriminantAnalysis από την βιβλιοθήκη Sklearn και επαναλαμβάνουμε την ίδια διαδικασία όπως το ερώτημα 2. Ως αλγόριθμο επίλυσης χρησιμοποιούμε τον “svd” και τις υπόλοιπες μεταβλητές της συνάρτησης by default.

```
1 ldareg = LDA(n_components=None, priors=None, shrinkage=None, solver='svd',store_covariance=False, tol=0.001)
2 ldareg.fit(X_train, y_train)
3 print("Test set accuracy with Logistic Regression before scaling: {:.3f}".format(ldareg.score(X_test,y_test)))
```

```
Test set accuracy with Logistic Regression before scaling: 0.681
```

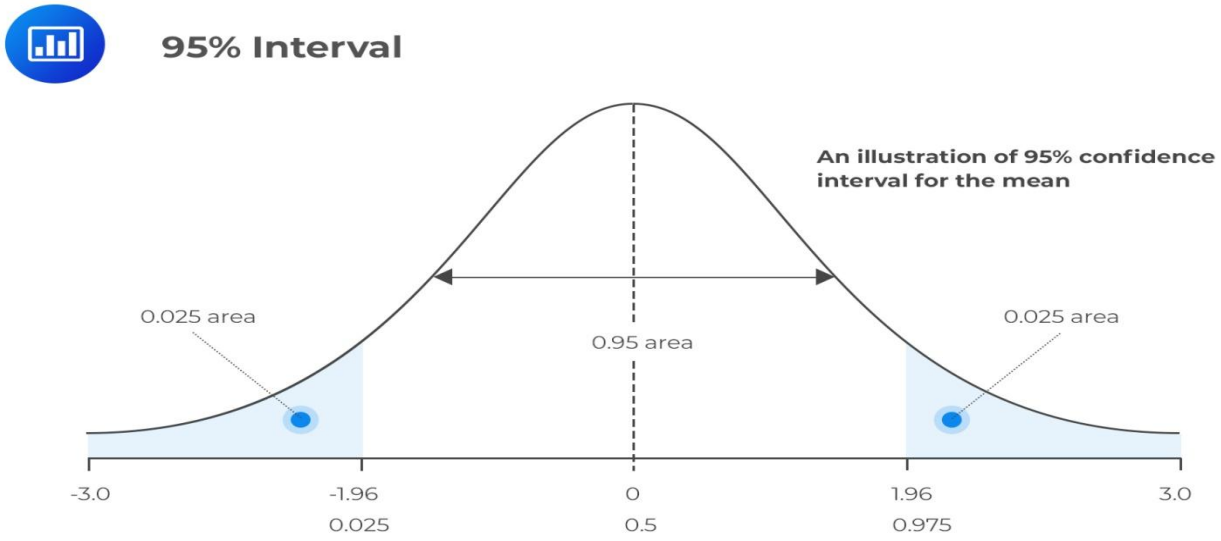
Η ορθότητα του μοντέλου δοκιμαζόμενη στο σύνολο αξιολόγησης παίρνει την τιμή acc=0.681

Ερώτημα Δ

Παρατηρούμε ότι η απόδοση των δύο συναρτήσεων LogisticRegression και LDA είναι περίπου ίδια. Αντιθέτως, η διαφορά στην ταχύτητα σύγκλισης των δύο μοντέλων είναι χαοτική. Στο συγκεκριμένο σύνολο δεδομένων όταν χρησιμοποιούμε την συνάρτηση LogisticRegression, παρατηρούμε ότι δεν είναι ευαίσθητο στην αλλαγή των υπερπαραμέτρων, ενώ η διαδικασία εύρεσης των κατάλληλων υπερπαραμέτρων είναι χρονοβόρα. Αντιθέτως, η διαδικασία που εκτελέσαμε στο Ερώτημα Γ δίνει παρόμοια αποτελέσματα σε πολύ μικρότερο χρόνο.

Ερώτημα Ε

Χρησιμοποιούμε την LinearDiscriminantAnalysis (LDA) για να εκπαιδεύσουμε και να αξιολογήσουμε το μοντέλο. Αυξάνουμε σταδιακά τον αριθμό των παραδειγμάτων που χρησιμοποιούμε για το σύνολο αξιολόγησης και υπολογίζουμε το εύρος του σφάλματος αξιολόγησης. Για κάθε ένα από τα παραδείγματα χρησιμοποιούμε διαστήματα εμπιστοσύνης 95%.



Σχήμα 3: Απεικόνιση του διαστήματος εμπιστοσύνης 95%

Υπολογίζουμε το $Interval = z * \sqrt{(\text{accuracy} * (1 - \text{accuracy})) / \text{n})}$ **εξ. Ε. 1**

Όπου z=1.96 για διαστήματα εμπιστοσύνης 95% και n ο μεταβλητός αριθμός παραδειγμάτων

Και το έυρος του συστήματος αξιολόγησης θα είναι $standar\ error \pm interval$. Παρακάτω βλέπουμε τις μετρήσεις για τα διαφορετικά δείγματα αξιολόγησης με άυξουσα σειρά

```
accuracy=0.678
interval=0.004
lower=31.791, upper=32.550
accuracy=0.683
interval=0.003
lower=31.430, upper=31.966
accuracy=0.681
interval=0.002
lower=31.671, upper=32.109
accuracy=0.681
interval=0.002
lower=31.704, upper=32.083
accuracy=0.678
interval=0.002
lower=32.031, upper=32.371
accuracy=0.680
interval=0.002
lower=31.862, upper=32.172
accuracy=0.681
interval=0.001
lower=31.783, upper=32.069
accuracy=0.678
interval=0.001
lower=32.036, upper=32.305
accuracy=0.680
interval=0.001
lower=31.859, upper=32.112
accuracy=0.676
interval=0.001
lower=32.243, upper=32.490
```

Συμπεράσματα:

- Στους αλγόριθμους που έχουν περιορισμένο θεωρητικό υπόβαθρο συχνά πρέπει να πειραματιστούμε χωρίς καθοδήγηση για να πετύχουμε κάποιο καλό αποτέλεσμα. Μπορούμε να χρησιμοποιήσουμε μία συνάρτηση πλέγματος για να βρούμε τις κατάλληλες υπερπαραμέτρους και τον καλύτερο αλγόριθμο.
- Παρατηρούμε ότι το $confidence\ interval$ αυξάνεται με τη χωρητικότητα G και μειώνεται με την αύξηση των δεδομένων εκπαίδευσης. Επίσης, αύξηση της χωρητικότητας G οδηγεί σε μείωση του $Rtrain$
- Στην αξιολόγηση των αλγορίθμων, παρατηρούμε ότι ο αλγόριθμος ο οποίος είναι καλύτερος έχει μικρότερο άνω όριο σφάλματος αξιολόγησης από το κάτω όριο του άλλου αλγορίθμου.