ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής

Εργασία Μαθήματος

«Αναλυτική Δεδομένων (6ο εξ.)»

Όνομα φοιτητή – Αρ. Μητρώου	
Ημερομηνία παράδοσης	

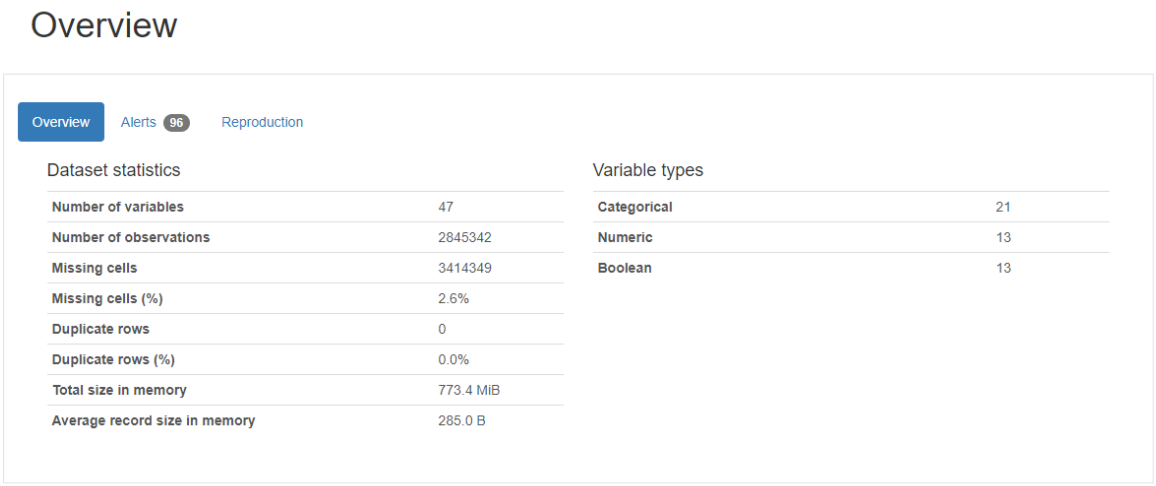
Εισαγωγή

Στόχος της εργασίας είναι να φέρει τον φοιτητή σε επαφή με την διαδικασία της Αναλυτικής Δεδομένων και της Μηχανικής Μάθησης μέσω εφαρμογής τεχνικών σε ένα σύνολο δεδομένων. Πιο συγκεκριμένα, το σύνολο δεδομένων αποτελείται από δεδομένα ατυχημάτων στις Η.Π.Α την περίοδο 2016-2021. Η εργασία υλοποιήθηκε με την βοήθεια του Jupyter Notebook και της γλώσσας Python 3.8. Επιπλέον, βασικές βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι pandas, pandas_profiling, scikit-learn, seaborn και tensorflow. Να σημειωθεί πως για την εκτέλεση του Notebook απαιτείται η δημιουργία κατάλληλου conda environment με τη βοήθεια του requirements.txt.

1. Γνωριμία με τα δεδομένα

Πρώτο βήμα αποτελεί η γνωριμία του αναλυτή με τα δεδομένα. Αφού κατεβάσουμε τα δεδομένα από το Kaggle, δημιουργούμε ένα DataFrame και τα φορτώνουμε (βλ. JN 1.1)[1]. Σε αυτό το σημείο αξίζει να διαβάσουμε το paper που συνοδεύει τα δεδομένα, ώστε να κατανοήσουμε τη σημασία των μεταβλητών, από πού προήλθαν, πώς επεξεργάστηκαν και τί χαρακτηριστικά παρουσιάζουν. Στη συνέχεια και στο πλαίσιο της διερευνητικής ανάλυσης δεδομένων, δημιουργούμε μια αναφορά των δεδομένων με χρήσιμα στοιχεία τους, με τη βοήθεια της βιβλιοθήκης pandas_profiling (βλ. JN 1.2). Η αναφορά περιλαμβάνει για κάθε στήλη δεδομένων γραφήματα, κατανομές, τύπους δεδομένων, στατιστικά, πλήθος ελλιπών τιμών μεταξύ άλλων. Η αναφορά μπορεί να βρεθεί στο αρχείο profile.html.

Παρατηρούμε, λοιπόν, πως το σύνολο δεδομένων αποτελείται από 47 χαρακτηριστικά (στήλες) και 2,845,342 εγγραφές/δείγματα (γραμμές). Τα χαρακτηριστικά αποτελούνται από 21 κατηγορικά, 13 αριθμητικά και 13 boolean.

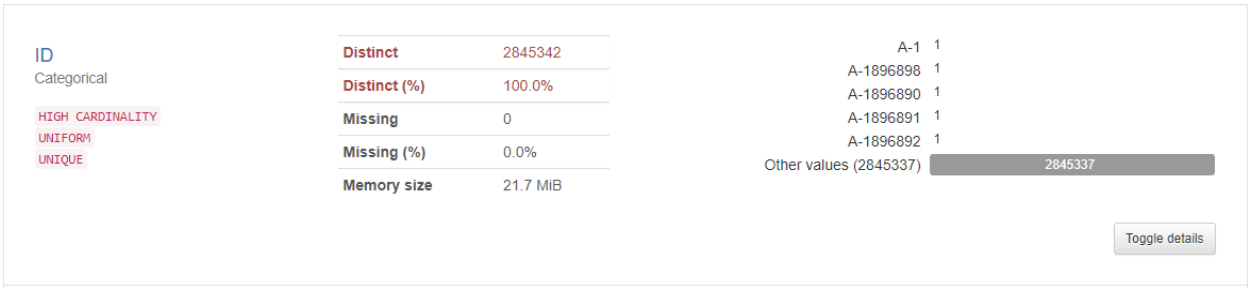


Παρακάτω αναλύονται τα χαρακτηριστικά.

ID

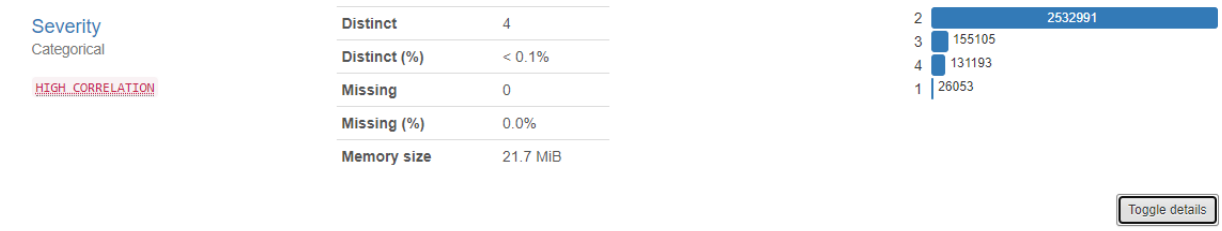
Αποτελεί το μοναδικό αναγνωριστικό μιας εγγραφής.

[1] Βλέπε Jupyter Notebook US_Accidents_Analysis.ipynb την υπο-ενότητα 1.1.



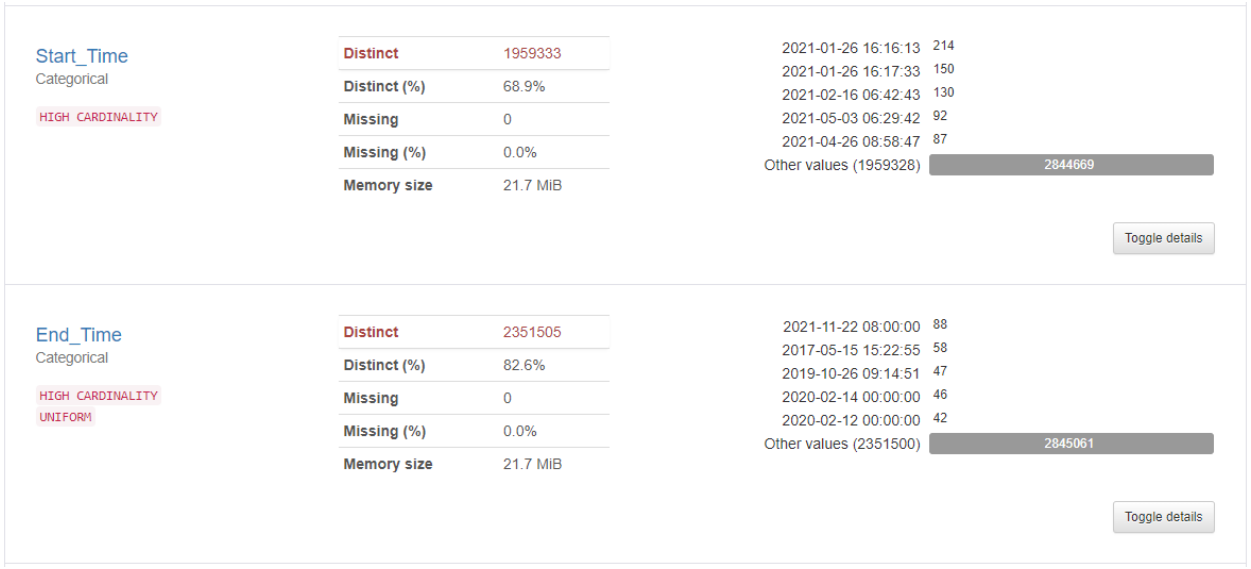
Severity

Η Σοβαρότητα του ατυχήματος είναι μια τακτική (ordinal) κατηγορική μεταβλητή, δηλαδή λαμβάνει τις τιμές 1 μέχρι 4 ανάλογα με τη σοβαρότητα. Παρατηρούμε ότι το 89% των δεδομένων ανήκουν στην κατηγορία σοβαρότητας 2, ενώ δεν υπάρχουν ελλιπή δεδομένα.



Start_Time & End_Time

Αποτελούν την τοπική ώρα αρχή και τέλους του ατυχήματος.

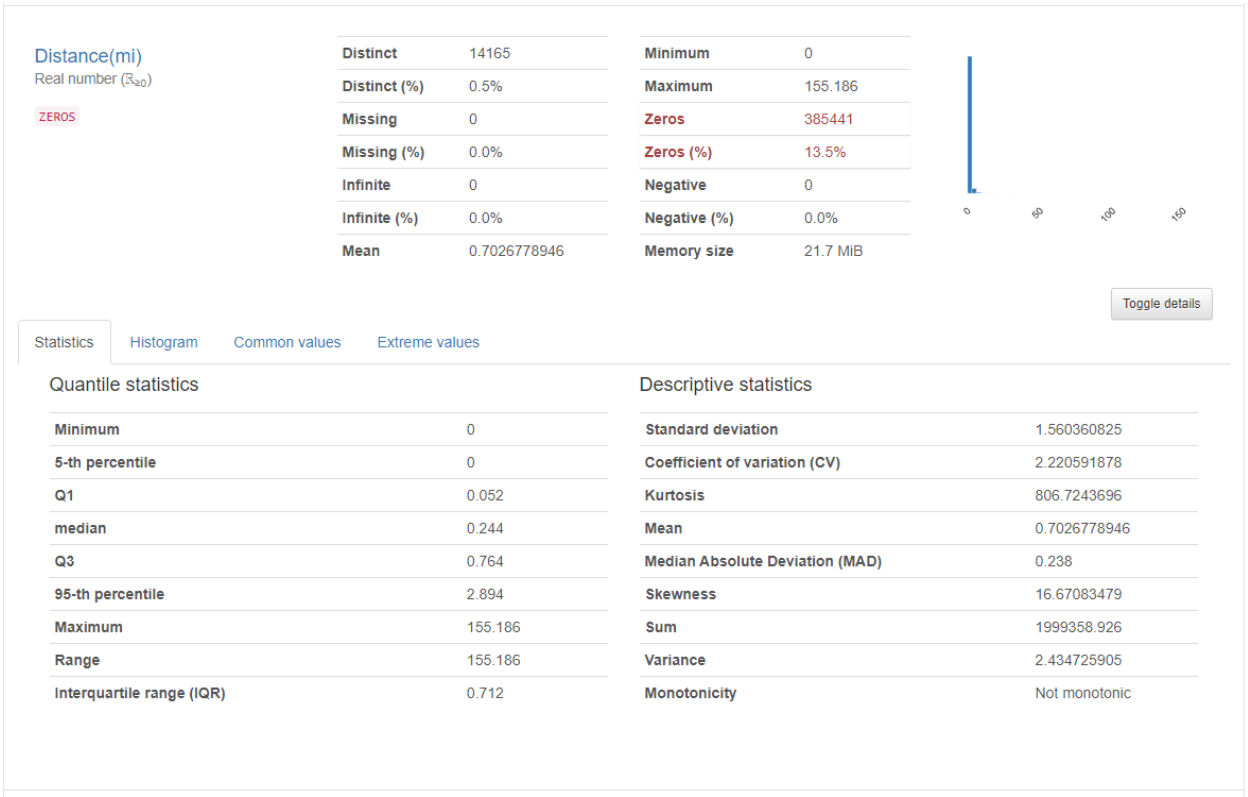


Start_Lat, Start_Lng, End_Lat & End_Lng

Δηλώνουν τις συντεταγμένες αρχής και τέλους του ατυχήματος. Τα στατιστικά τους δεν παρουσιάζονται μιας και δεν διαθέτουν κάποιο αξιόλογο χαρακτηριστικό, πέρα από το γεγονός ότι δεν υπάρχουν ελλιπής τιμές.

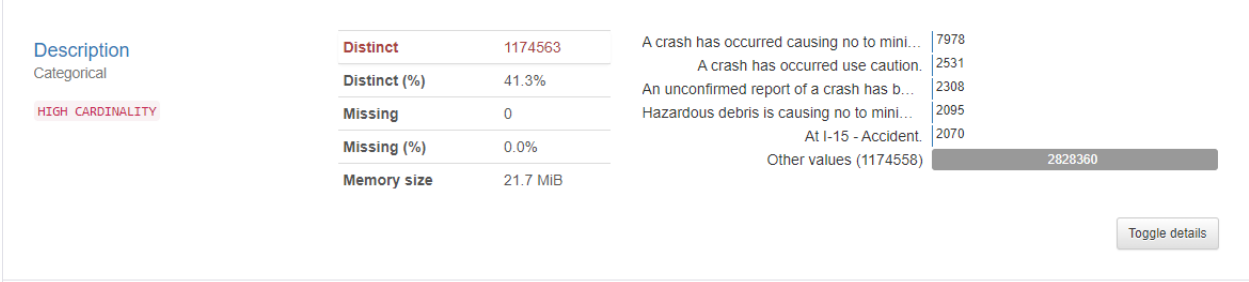
Distance(mi)

Η απόσταση των επιπτώσεων (μποτιλιάρισμα) του ατυχήματος. Οι τιμές της κυμαίνονται κυρίως στο διάστημα [0.052, 2.894], ενώ παρατηρούνται ακραίες τιμές.



Description

Μια συνοπτική περιγραφή του ατυχήματος και των επιπτώσεων. Το αξιοσημείωτο σε αυτό το χαρακτηριστικό που λαμβάνουμε από το paper είναι πως η ομάδα που έφτιαξε το σύνολο δεδομένων έχει ήδη επεξεργαστεί την περιγραφή και έχει εξάγει χρήσιμα χαρακτηριστικά, όπως αναφέρουν στην ενότητα 4.2.3. Τα εξαγόμενα χαρακτηριστικά αποτελούν κυρίως τις Boolean μεταβλητές που θα περιγραφτούν αργότερα.

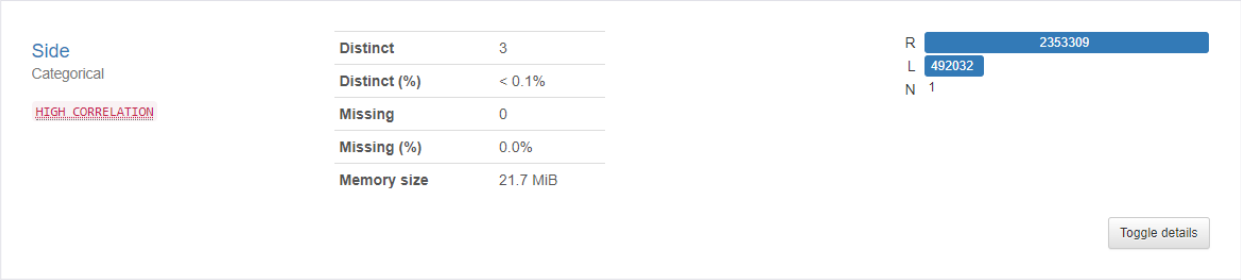


Number, Street, Country, State, Zipcode

Αποτελούν τη διεύθυνση του ατυχήματος.

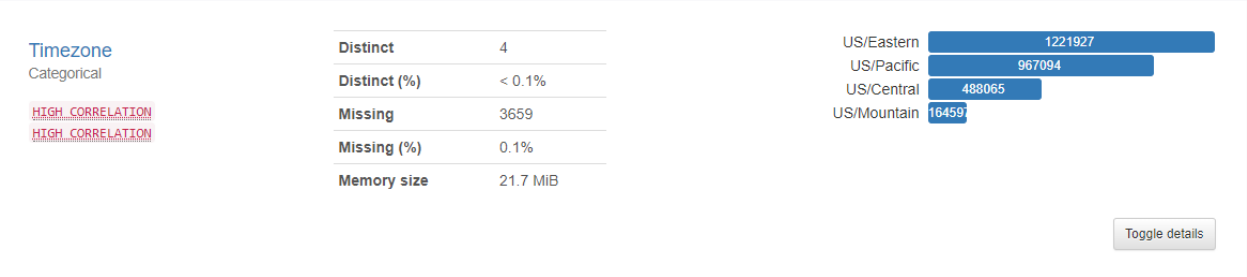
Side

Κατηγορική μεταβλητή που υποδηλώνει την πλευρά του δρόμου που έγινε το ατύχημα και έχει εξαχθεί από την περιγραφή. Παρατηρούμε πως υπάρχει μια εσφαλμένη τιμή και πως το μεγαλύτερο πλήθος έχει λάβει τιμή R δηλαδή δεξιά.



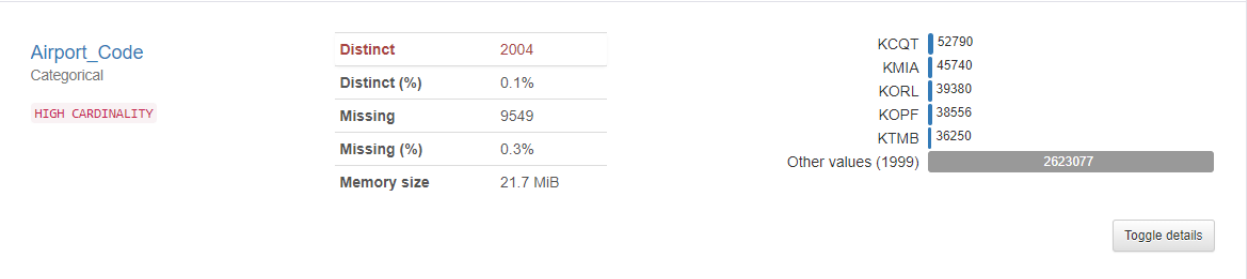
Timezone

Κατηγορική μεταβλητή που υποδηλώνει την ζώνη ώρας που συνέβει το ατύχημα.



Airport_Code

Ο κωδικός του αεροδρομίου από το οποίο συλλέχθηκαν τα δεδομένα καιρού.



Weather_Timestamp

Χρονοσφραγίδα της λήψης των μετεωρολογικών δεδομένων. Όπως αναφέρεται στο συνοδευτικό paper, η ώρα της λήψης των δεδομένων δεν είναι η ίδια με αυτή του ατυχήματος, αλλά η «κοντινότερη» δυνατή λήψη.

Weather_Timestamp
Categorical

HIGH CARDINALITY
MISSING

Distinct	474214
Distinct (%)	17.0%
Missing	50736
Missing (%)	1.8%
Memory size	21.7 MiB

2021-12-17 14:53:00	640
2021-12-23 14:53:00	629
2021-01-26 15:53:00	598
2021-12-06 16:53:00	595
2021-12-03 16:53:00	593
Other values (474209)	2791551

Toggle details

Temperature(F)

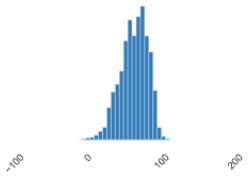
Η τοπική θερμοκρασία την ώρα του ατυχήματος. Παρατηρούμε μια σχετικά κανονική κατανομή με ορισμένες ακραίες τιμές και μερικές ελλιπής.

Temperature(F)
Real number (ℝ)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	788
Distinct (%)	< 0.1%
Missing	69274
Missing (%)	2.4%
Infinite	0
Infinite (%)	0.0%
Mean	61.79355592

Minimum	-89
Maximum	196
Zeros	984
Zeros (%)	< 0.1%
Negative	6444
Negative (%)	0.2%
Memory size	21.7 MiB



Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	-89
5-th percentile	29
Q1	50
median	64
Q3	76
95-th percentile	88.9
Maximum	196
Range	285
Interquartile range (IQR)	26

Descriptive statistics

Standard deviation	18.62262938
Coefficient of variation (CV)	0.3013684697
Kurtosis	0.03002155249
Mean	61.79355592
Median Absolute Deviation (MAD)	13
Skewness	-0.4910794711
Sum	171543113.2
Variance	346.8023252
Monotonicity	Not monotonic

Wind_Chill(F)

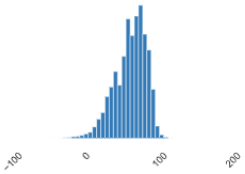
Η θερμοκρασία του αέρα. Παρατηρούνται όμοια χαρακτηριστικά με την προηγούμενη μεταβλητή.

Wind_Chill(F)
Real number (ℝ)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	897
Distinct (%)	< 0.1%
Missing	469643
Missing (%)	16.5%
Infinite	0
Infinite (%)	0.0%
Mean	59.6582309

Minimum	-89
Maximum	196
Zeros	1322
Zeros (%)	< 0.1%
Negative	20659
Negative (%)	0.7%
Memory size	21.7 MiB



Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	-89
5-th percentile	21
Q1	46
median	63
Q3	76
95-th percentile	88
Maximum	196
Range	285
Interquartile range (IQR)	30

Descriptive statistics

Standard deviation	21.16096744
Coefficient of variation (CV)	0.354703234
Kurtosis	0.3682111381
Mean	59.6582309
Median Absolute Deviation (MAD)	14
Skewness	-0.7082722667
Sum	141729999.5
Variance	447.7865429
Monotonicity	Not monotonic

Humidity(%)

Η τοπική υγρασία την ώρα του ατυχήματος. Εδώ, παρατηρείται μια ιδιαίτερη κατανομή, με θετική κλίση μέχρι τη μέση του πεδίου τιμών, σταθερή για ένα εύρος και ύστερα παρουσιάζει υψηλές διακυμάνσεις. Παρουσιάζονται ελλιπής τιμές.

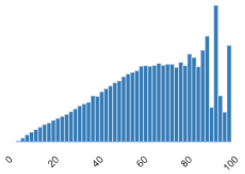
Humidity(%)

Real number (ℝ≥0)

MISSING

Distinct	100
Distinct (%)	< 0.1%
Missing	73092
Missing (%)	2.6%
Infinite	0
Infinite (%)	0.0%
Mean	64.36545225

Minimum	1
Maximum	100
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	21.7 MiB



Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	1
5-th percentile	23
Q1	48
median	67
Q3	83
95-th percentile	96
Maximum	100
Range	99
Interquartile range (IQR)	35

Descriptive statistics

Standard deviation	22.8745681
Coefficient of variation (CV)	0.3553858056
Kurtosis	-0.6827235834
Mean	64.36545225
Median Absolute Deviation (MAD)	17
Skewness	-0.4157629378
Sum	178437125
Variance	523.2458658
Monotonicity	Not monotonic

Pressure(in)

Η πίεση της ατμόσφαιρας, παρουσιάζει ιδιαίτερη κατανομή με ορισμένες ακραίες τιμές και ελλιπής τιμές.

Pressure(in)

Real number (ℝ≥0)

HIGH CORRELATION
MISSING

Distinct	1068
Distinct (%)	< 0.1%
Missing	59200
Missing (%)	2.1%
Infinite	0
Infinite (%)	0.0%
Mean	29.47234438

Minimum	0
Maximum	58.9
Zeros	1
Zeros (%)	< 0.1%
Negative	0
Negative (%)	0.0%
Memory size	21.7 MiB

Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

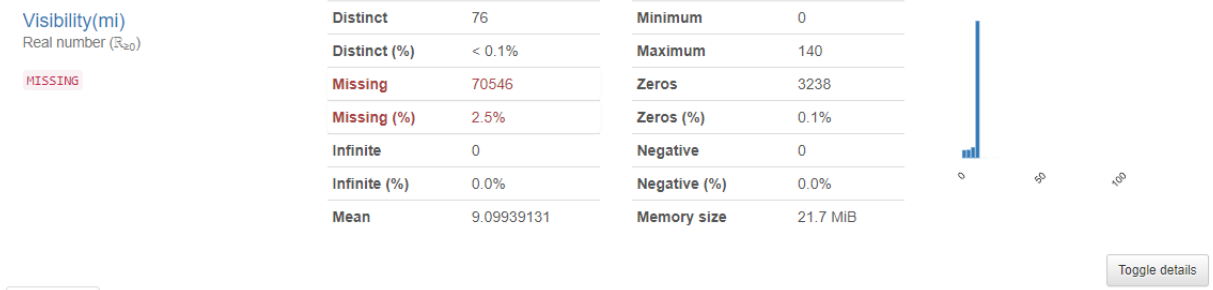
Minimum	0
5-th percentile	27.25
Q1	29.31
median	29.82
Q3	30.01
95-th percentile	30.21
Maximum	58.9
Range	58.9
Interquartile range (IQR)	0.7

Descriptive statistics

Standard deviation	1.045286497
Coefficient of variation (CV)	0.03546668983
Kurtosis	16.72394531
Mean	29.47234438
Median Absolute Deviation (MAD)	0.26
Skewness	-3.257540169
Sum	82114136.52
Variance	1.092623861
Monotonicity	Not monotonic

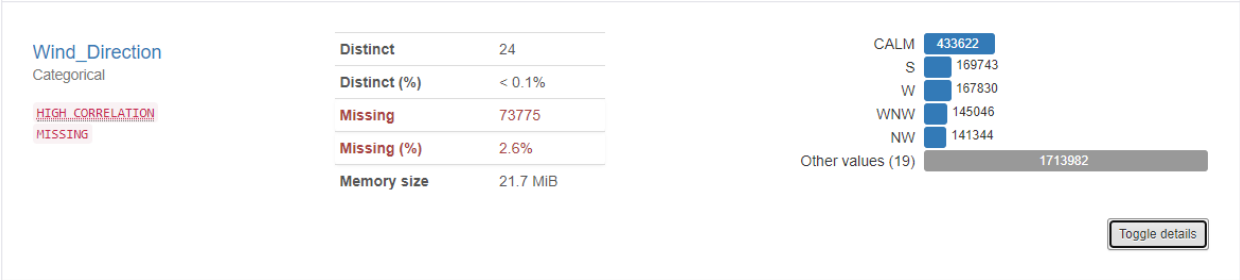
Visibility(mi)

Η τοπική ορατότητα στον δρόμο του ατυχήματος. Ξανά, εμφανίζεται ιδιαίτερη κατανομή με ακραίες και ελλιπής τιμές.



Wind_Direction

Κατηγορική μεταβλητή για την κατεύθυνση του αέρα. Παρατηρείται πως υπάρχουν πολλές κατηγορίες αφού είναι λεπτομερής, π.χ. υπάρχει κατηγορία Βόρειος-Βορειοδυτικός αντί σκέτο Βόρειος.



Wind_Speed(mph)

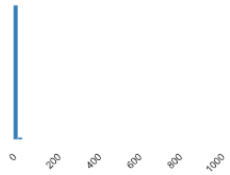
Η ταχύτητα του αέρα, παρουσιάζει ιδιαίτερη κατανομή με τις περισσότερες τιμές να συσσωρεύονται στο εύρος [3.5, 10], ενώ υπάρχουν σχετικά αρκετές ακραίες τιμές.

Wind_Speed(mph)
Real number (ℝ≥0)

MISSING
ZEROS

Distinct	136
Distinct (%)	< 0.1%
Missing	157944
Missing (%)	5.6%
Infinite	0
Infinite (%)	0.0%
Mean	7.395044203

Minimum	0
Maximum	1087
Zeros	433636
Zeros (%)	15.2%
Negative	0
Negative (%)	0.0%
Memory size	21.7 MiB



Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	0
Q1	3.5
median	7
Q3	10
95-th percentile	17
Maximum	1087
Range	1087
Interquartile range (IQR)	6.5

Descriptive statistics

Standard deviation	5.52745395
Coefficient of variation (CV)	0.7474538081
Kurtosis	1508.031366
Mean	7.395044203
Median Absolute Deviation (MAD)	3
Skewness	10.24972664
Sum	19873427
Variance	30.55274717
Monotonicity	Not monotonic

Precipitation(in)

Η κατακρήμνιση την ώρα του ατυχήματος. Το 74% των δεδομένων λαμβάνει τιμή 0, ενώ υπάρχουν αρκετές ακραίες και ελλιπής τιμές.

Precipitation(in)
Real number (ℝ≥0)

MISSING
SKEWED
ZEROS

Distinct	230
Distinct (%)	< 0.1%
Missing	549458
Missing (%)	19.3%
Infinite	0
Infinite (%)	0.0%
Mean	0.00701693988

Minimum	0
Maximum	24
Zeros	2104242
Zeros (%)	74.0%
Negative	0
Negative (%)	0.0%
Memory size	21.7 MiB

Toggle details

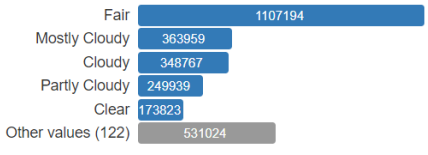
Weather_Condition

Κατηγορική μεταβλητή η οποία περιγράφει την κατάσταση του καιρού. Αποτελείται από 127 κατηγορίες που μπορούν να θεωρηθούν αρκετά λεπτομερείς, για παράδειγμα Cloudy, Partial Cloudy, Mostly Cloudy. Υπάρχουν ελλιπής τιμές.

Weather_Condition
Categorical

HIGH CARDINALITY
MISSING

Distinct	127
Distinct (%)	< 0.1%
Missing	70636
Missing (%)	2.5%
Memory size	21.7 MiB



Toggle details

Amenity, Bump, Crossing, Give_Way, Junction, No_Exit, Railway, Roundabout, Station, Stop, Traffic_Calming, Traffic_Signal, Turning_Loop

Boolean μεταβλητές οι οποίες παρέχουν πληροφορίες για τον δρόμο του ατυχήματος. Όπως αναφέρεται στο συνοδευτικό paper, οι πληροφορίες αυτές έχουν εξαχθεί από την περιγραφή του ατυχήματος. Δεν υπάρχουν ελλιπής τιμές, ενώ το 99% των εγγραφών έχουν την τιμή false.

Sunrise_Sunset, Civil_Twilight, Nautical_Twilight, Astronomical_Twilight

Κατηγορικές μεταβλητές που περιγράφουν την θέση του ήλιο και την κατάσταση φωτεινότητας, με δυνατές τιμές Day και Night. Υπάρχουν ελάχιστες ελλιπής τιμές. Σε όλες τις μεταβλητές, η κατηγορία Day εμφανίζεται περίπου στο 75% των δειγμάτων.

2. Μηχανική δεδομένων

Έχοντας μια ποιοτική και ποσοτική εικόνα των δεδομένων, μπορούμε να συνεχίσουμε με τη διαδικασία της μηχανικής δεδομένων. Η διαδικασία αυτή αποτελεί ένα είδος προεπεξεργασίας των δεδομένων, ώστε να καθαριστούν και να μετατραπούν σε μια αποδεκτή μορφή για επεξεργασία. Δεν χρησιμοποιείται ο όρος pre-processing αφού η διαδικασία αυτή συμβαίνει όταν προετοιμάζουμε τα δεδομένα μας για ένα συγκεκριμένο μοντέλο. Γενικά στο χώρο υπάρχουν διάφορες ερμηνείες.

Στη φάση αυτή, οι διαδικασίες που υλοποιούμε είναι:

2.1. Διαγραφή περιττών χαρακτηριστικών

Διαγράφουμε τα χαρακτηριστικά/στήλες που δεν θα χρησιμοποιήσουμε στις μελλοντικές διαδικασίες μας. Πιο συγκεκριμένα, θα αγνοήσουμε τις στήλες:

- ID: αδιάφορο χαρακτηριστικό
- Description: η πληροφορία του αποτυπώνεται στις εξαγόμενες Boolean μεταβλητές, όπως περιγράφεται στο συνοδευτικό paper στην ενότητα 4.2.3
- Number, Street, County, Country, State, Zipcode: λαμβάνουμε αντίστοιχη πληροφορία από τις Boolean μεταβλητές, π.χ. ένας συγκεκριμένος δρόμος έχει κόμβους, σήματα στοπ κλπ. σε κάθε εγγραφή με τον αντίστοιχο δρόμο. Επιπλέον, η πληροφορία συσχετίζεται εύκολα με τις συντεταγμένες του ατυχήματος. Να σημειωθεί πως κρατάμε το City ώστε να αποδώσουμε τιμές στα ελλιπή δεδομένα καιρού.
- Airport_Code: αδιάφορο χαρακτηριστικό
- Timezone: αδιάφορο αφού θέλουμε τις χρονοσφραγίδες σε τοπικές ώρες ώστε ενθυλακώνονται οι σχετικές πληροφορίες του ατυχήματος. Π.χ. τα περιβαλλοντικά χαρακτηριστικά είναι παρόμοιά στις 8 τοπική πρωινή ώρα στην ανατολική και δυτική ακτή, αλλά διαφορετικά σε απόλυτη ώρα.
- Weather_Timestamp: θα μπορούσε να γίνει ανάλυση της απόκλισης με την χρονοσφραγίδα του ατυχήματος αλλά έχει ήδη γίνει στο συνοδευτικό paper, ενότητα 4.2.2.
- Wind_Chill: παρατηρείται υψηλή γραμμική συσχέτιση με τη μεταβλητή Temperature, οπότε κρατάμε την τελευταία.
- Wind_Direction: υποθέτουμε πως δεν θα βοηθήσει στην ανάλυσή μας. Επιπλέον, διαθέτει πολλές κατηγορίες με περιττή πληροφορία, ενώ η μείωση των κατηγοριών απαιτεί χρονοβόρα δουλεία με χαμηλό όφελος.
- Turning_Loop: σταθερό σε όλες τις εγγραφές

- Sunrise_Sunset, Civil_Twilight, Nautical_Twilight: υποθέτουμε πως η μεταβλητή Astronomical_Twilight ενθυλακώνει αρκετά καλά την πληροφορία των υπολοίπων.

```
data_df = us_acc.drop(['ID', 'Description', 'Number', 'Street',
                       'County', 'State', 'Zipcode', 'Country',
                       'Timezone', 'Weather_Timestamp', 'Airport_Code', 'Turning_Loop', 'Wind_Direction',
                       'Wind_Chill(F)', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight'], axis=1)
```

2.2. Μετατροπή δεδομένων

Τα δεδομένα δεν είναι πάντα στη μορφή ή/και στον τύπο δεδομένων που θέλουμε. Οπότε, εκτελούμε κατάλληλες εντολές για να τα μετατρέψουμε στην επιθυμητή μορφή. Πιο συγκεκριμένα, μετατρέπουμε τον τύπο δεδομένων των Start_Time και End_Time από object σε pandas.DateTime.

```
data_df['Start_Time'] = pd.to_datetime(data_df['Start_Time']).dt.date
data_df['End_Time'] = pd.to_datetime(data_df['End_Time']).dt.date
```

Στη συνέχεια, μετονομάζουμε ορισμένες στήλες για ευκολία.

```
data_df = data_df.rename({'Distance(mi)': 'Distance',
                          'Temperature(F)': 'Temperature',
                          'Humidity(%)': 'Humidity',
                          'Pressure(in)': 'Pressure',
                          'Visibility(mi)': 'Visibility',
                          'Wind_Speed(mph)': 'Wind_Speed',
                          'Precipitation(in)': 'Precipitation'}, axis=1)
```

Έπειτα, διαγράφουμε την εσφαλμένη εγγραφή με την τιμή ‘N’ στη στήλη ‘Side’ και τη μετατρέπουμε, μαζί με την Astronomical_Twilight σε Boolean μεταβλητές.

```
# a 'N' value exists, so we drop row + to boolean
data_df.loc[:, 'Side'] = data_df.loc[data_df['Side'] != 'N']['Side'].replace({'R': 1, 'L': 0})
data_df.loc[:, 'Astronomical_Twilight'] = data_df['Astronomical_Twilight'].replace({'Day': 1, 'Night': 0})
```

Μειώνουμε τις κατηγορίες στο πεδίο Weather_Condition κάνοντας ορισμένες ποιοτικές συγχωνεύσεις, π.χ. Cloudy, Partial Cloudy, Mostly Cloudy → Cloudy, καταλήγοντας σε 7 κατηγορίες. Έπειτα, τις αντιστοιχούμε σε μια άτυπη αλλά με νόημα κατάταξη.

```
# 'Tornado' appears in 8 records, pretty low representation, so we drop it.
data_df.loc[:, 'Weather_Condition'] = data_df.loc[data_df['Weather_Condition'] != 'Tornado', 'Weather_Condition']

# Reducing the number of possible categorical values
data_df.loc[:, 'Weather_Condition'] = data_df['Weather_Condition'].replace(to_replace={
                                r'^.*(Rain|Squalls|Drizzle|Showers|Thunderstorm|Precipitation|Wintry|Sleet).*$': 'Rain',
                                r'^.*(Cloud|Overcast).*$': 'Cloudy',
                                r'^.*(Snow|Pellets|Hail).*$': 'Snow',
                                r'^.*(Fog|Haze|Mist|Smoke).*$': 'Fog',
                                r'^.*(Thunder|T-Storm).*$': 'Thunder',
                                r'^.*(Dust|Sand|Ash).*$': 'Dust',
                                r'^.*(Clear|Fair).*$': 'Fair'}, regex=True)

# Specific label categorical encoding, from good weather to bad
data_df.loc[:, "Weather_Condition"] = data_df['Weather_Condition'].replace({"Fair": 0,
                                                                           "Cloudy": 1,
                                                                           "Fog": 2,
                                                                           "Dust": 3,
                                                                           "Rain": 4,
                                                                           "Thunder": 5,
                                                                           "Snow": 6})
```

Τέλος, πραγματοποιούμε μικρές μετατροπές τύπου δεδομένων και διαγράφουμε τα κενά.

```
    # There are few nans so we can drop the respective rows
    data_df = data_df.dropna(subset=['Weather_Condition', 'Side', 'Astronomical_Twilight'])
    data_df.loc[:, 'Side'] = data_df['Side'].astype(int)
    data_df.loc[:, 'Astronomical_Twilight'] = data_df['Astronomical_Twilight'].astype(int)

    # Bool variables to 0/1
    bool_lbls = ['Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
                 'Roundabout', 'Station', 'Stop','Traffic_Calming', 'Traffic_Signal']
    data_df[bool_lbls] = data_df[bool_lbls].astype(int)
```

2.3. Καθαρισμός δεδομένων από ακραίες τιμές

Οι ακραίες τιμές παρατηρούνται στα δεδομένα καιρού. Ελέγχοντας τη κατανομή των καιρικών μεταβλητών είναι δυνατό να ορίσουμε αποδεκτά εύρη και να απορρίψουμε τις ακραίες τιμές.

```
    data_df = data_df.loc[(data_df['Temperature'] < 120) &
                          (data_df['Pressure'] < 50) &
                          (data_df['Pressure'] > 10) &
                          (data_df['Visibility'] < 100) &
                          (data_df['Wind_Speed'] < 300) &
                          (data_df['Precipitation'] < 15)]
```

2.4. Απόδοση ελλιπών τιμών

Παρατηρούμε πως υπάρχουν αρκετές ελλιπής τιμές στα δεδομένα καιρού. Μια απλή λύση είναι να υπολογίσουμε την μέση τιμή κάθε πόλης και να γίνει αντίστοιχη απόδοση των ελλιπών τιμών βάσει της πόλης. Τέλος, υπάρχει η πιθανότητα να αποδοθούν κενές τιμές σε πόλεις που δεν είχαν κάποια τιμή, δηλαδή η μέση τιμής της πόλης να είναι nan. Για αυτό το λόγο, διαγράφουμε τις εγγραφές με κενή τιμή μετά την απόδοση.

```
labels = ['Temperature', 'Humidity', 'Pressure', 'Visibility', 'Wind_Speed', 'Precipitation']
average_per_city = data_df.groupby('City')[labels].mean().reset_index()
for label in labels:
    data_df.loc[pd.isna(data_df[label]), label] = pd.merge(data_df
                                .loc[pd.isna(data_df[label]), 'City'], average_per_city, on='City')[label]

# After the imputation, there may be nan values due to lack of data for a city, so we finally drop them
data_df = data_df.dropna(subset=labels)
```

3. Συσταδοποίηση

Το πλήθος των χαρακτηριστικών καθιστούν την εύρεση ομοιοτήτων των εγγραφών μια δύσκολη διαδικασία. Οι ομοιότητες δεδομένων θα δημιουργούσαν ομάδες στο πολυδιάστατο χώρο των μεταβλητών. Οι ομάδες θα ήταν εύκολα ορατές σε ένα καρτεσιανό πλέγμα αν είχαμε 2 με 3 μεταβλητές, όμως το πρόβλημά μας διαθέτει πάνω από 20! Οι μέθοδοι συσταδοποίησης μας επιτρέπουν να εντοπίσουμε αυτές τις ομάδες στον πολυδιάστατο χώρο χρησιμοποιώντας μετρικές αποστάσεων. Μερικές μέθοδοι τείνουν να δημιουργούν ομάδες υπερσφαίρας, ενώ άλλες έχουν την δημιουργούν ομάδες διαφόρων σχημάτων, βάσει της πυκνότητας των δεδομένων. Αξίζει να σημειωθεί πως στο τέλος αυτές οι ομάδες ενδέχεται να παρουσιάζουν ιδιαιτερότητες των δεδομένων και αλληλεπιδράσεις των μεταβλητών που δεν είναι εύκολα εντοπίσιμες με άλλες μεθόδους, π.χ. στατιστικές.

3.1. Προεπεξεργασία δεδομένων

Για τη συγκεκριμένη ανάλυση θα επιλεχθεί ένα υποσύνολο των μεταβλητών του συνόλου δεδομένων. Επιπλέον, κρατάμε ξεχωριστά τη μεταβλητή Severity ως μεταβλητή στόχο για μελλοντικές διαδικασίες. Πιο συγκεκριμένα, αγνοούμε τις μεταβλητές Start_Time και End_Time αφού δεν θέλουμε να κάνουμε χρονική συσταδοποίηση. Επιπλέον, αγνοούμε τις μεταβλητές συντεταγμένων. Στόχος μας είναι η συσταδοποίηση βάσει των υπολοίπων χαρακτηριστικών, ενώ στην περίπτωση που λαμβάναμε υπόψιν το χρόνο και το χώρο, η συσταδοποίηση θα έδινε μεγάλη έμφαση σε αυτά τα χαρακτηριστικά.

```
columns = ['Distance', 'Side', 'Temperature', 'Humidity',
       'Pressure', 'Visibility', 'Wind_Speed', 'Precipitation',
       'Weather_Condition', 'Amenity', 'Bump', 'Crossing', 'Give_Way',
       'Junction', 'No_Exit', 'Railway', 'Roundabout', 'Station', 'Stop',
       'Traffic_Calming', 'Traffic_Signal', 'Astronomical_Twilight']

X = data_df[columns].values
y = data_df['Severity'].values
```

Η συσταδοποίηση απαιτεί τη χρήση συναρτήσεων απόστασης, αλλά υπάρχουν περιπτώσεις που χρησιμοποιούνται συναρτήσεις που δεν υπακούν στα μετρικά αξιώματα. Παρόλα αυτά, θα σταθούμε στην πρώτη περίπτωση. Τα αποτελέσματα των μετρικών επηρεάζονται ιδιαίτερα από την κλίμακα των μεταβλητών. Για παράδειγμα, μια μεταβλητή με μέγεθος 10 λαμβάνει λιγότερη σημασία σε σχέση με μια μεγέθους 100 όταν χρησιμοποιούμε την ευκλείδεια απόσταση. Για αυτό το λόγω, τα δεδομένα μας πρέπει να κανονικοποιούνται, δηλαδή να μεταφέρονται σε μια κοινή κλίμακα. Σε αυτή την περίπτωση, θα μας βοηθήσει η κλάση StandardScaler της βιβλιοθήκης scikit-learn η οποία μετατρέπει τη κατανομή κάθε μεταβλητής σε κανονική κατανομή. Βέβαια, εδώ παρατηρούμε πως αν κανονικοποιηθούν Boolean μεταβλητές απλά θα λαμβάνουν δυο σταθερές τιμές εξαρτώμενες από τη μέση τιμή των άσσων. Οπότε, μπορούμε να τις εξαιρέσουμε από την κανονικοποίηση.

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_scaled = sc.fit_transform(X[:, :9])
X_scaled = np.concatenate([X_scaled, X[:, 9:]], axis=1)
```

Τέλος, αξίζει να αναφερθεί πως οι Boolean μεταβλητές δεν αποδίδουν βέλτιστα αποτελέσματα στις συσταδοποιήσεις, αφού οι αποστάσεις σε μοναδιαίες συντεταγμένες στον πολυδιάστατο χώρο δεν ερμηνεύονται σωστά από την ευκλείδεια απόσταση. Ένα παράδειγμα παρουσιάζεται στο άρθρο της IBM[2]. Μια λύση που προτείνεται είναι η χρήση τεχνικών μείωσης διαστάσεων, ώστε οι παραγόμενες συνιστώσες να εκφράσουν, όσο είναι δυνατό, τη μοναδιαία πληροφορία σε λιγότερες διαστάσεις. Όμως, σε αυτό το σημείο επιθυμούμε να βρούμε σχέσεις μεταξύ των μεταβλητών ή/και χαρακτηριστικά των συστάδων. Χρησιμοποιώντας τεχνικές μείωσης διαστάσεων θα χάσουμε την απόδοση σημασίας στις αντίστοιχες μεταβλητές.

3.2. Αναζητώντας χρήσιμες πληροφορίες μέσω συσταδοποίησης

Ένας τρόπος για την εύρεση και κατανόηση των χαρακτηριστικών των συστάδων αποτελεί η διαδικασία απόδοσης συντελεστών σημασίας μέσω διαδικασίας επιβλεπόμενης ταξινόμησης. Πιο συγκεκριμένα, για διάφορες συνθέσεις υπερ-παραμέτρων, εκπαιδεύονται μοντέλα συσταδοποίησης χωρίς επίβλεψη. Ύστερα, έχοντας τις ετικέτες των συστάδων, για κάθε μια συστάδα εκπαιδεύεται ένα μοντέλο ταξινόμησης με επίβλεψη, θέτοντας ως στόχο 1 αν ανήκει στην εκάστοτε συστάδα, ή 0 διαφορετικά[3]. Σημαντική είναι η επιλογή μοντέλου ταξινόμησης , το οποίο θα πρέπει να αποδίδει συντελεστές «σημασίας» στις μεταβλητές, για παράδειγμα ένα δέντρο αποφάσεων. Στην περίπτωσή μας, επιλέχθηκε ο αλγόριθμος συσταδοποίησης KMeans, διότι αποτελεί τη γρηγορότερη μέθοδο συσταδοποίησης, ενώ παράλληλα ο τεράστιος όγκος δεδομένων καθιστά άλλους αργούς αλγορίθμους απαγορευτικούς. Για την ταξινόμηση επιλέχθηκε ο DecisionTreeClassifier αφού προσφέρει ικανοποιητικά τους συντελεστές «σημασίας» των μεταβλητών. Να σημειωθεί πως κατά τη διάρκεια των δοκιμών παρατηρήθηκε μεγάλη ανισορροπία στις ετικέτες των συστάδων με αποτέλεσμα να χρησιμοποιήσουμε μέθοδο under sampling.

```
from sklearn.tree import DecisionTreeClassifier
from imblearn.under_sampling import RandomUnderSampler

def get_variable_importances(X, labels):
    importances = []
    for lbl in np.unique(labels):
        y_bool= np.where(labels == lbl, 1, 0)
        X_und, y_und = RandomUnderSampler().fit_resample(X, y_bool)
        dtc = DecisionTreeClassifier()
        dtc.fit(X_und, y_und)
        importances.append(dtc.feature_importances_)
    return np.array(importances)
```

[2] https://www.ibm.com/support/pages/clustering-binary-data-k-means-should-be-avoided

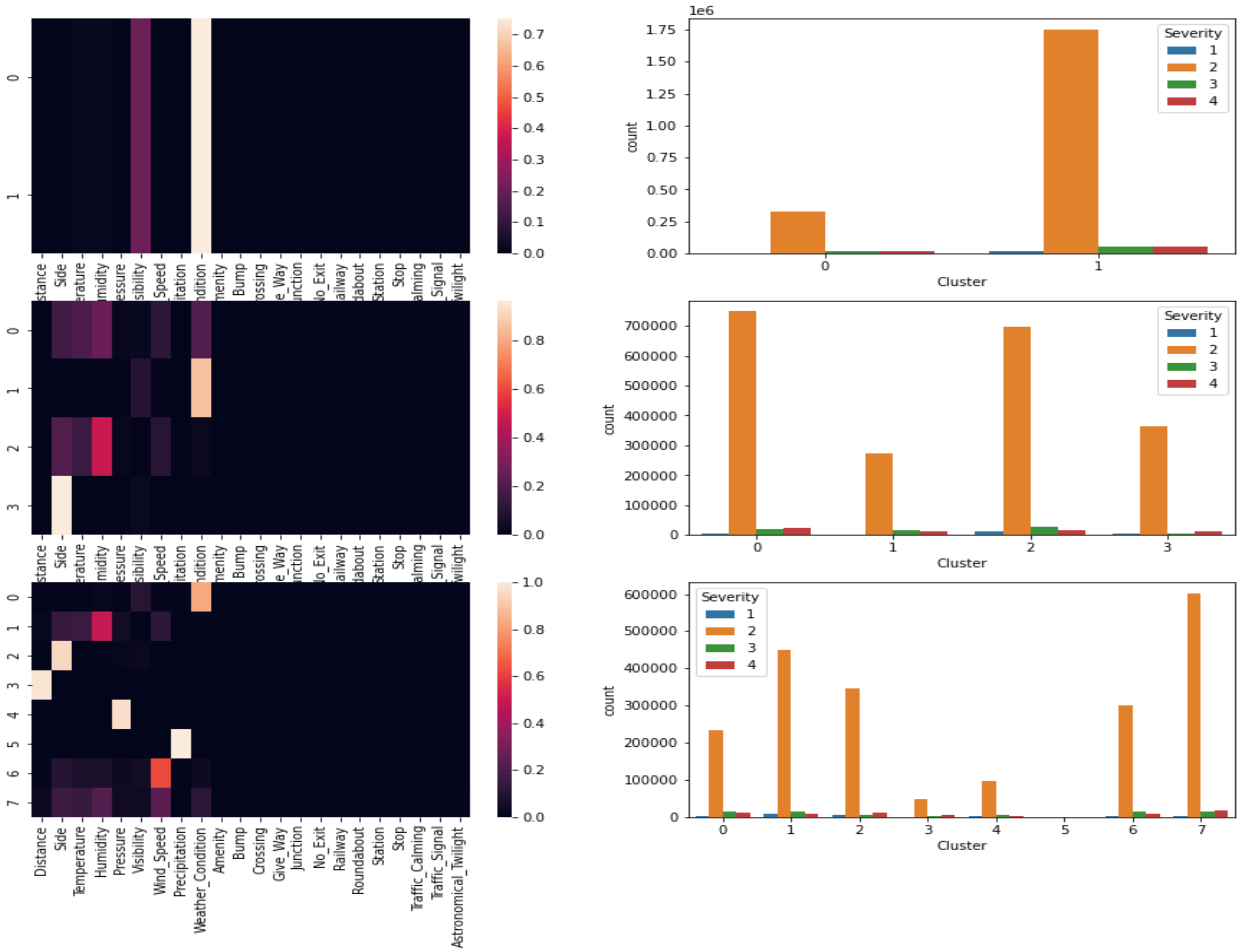
[3] https://www.quora.com/How-do-I-understand-the-characteristics-of-each-cluster-when-doing-a-K-Means-clustering-algorithm

Έγιναν τρεις δοκιμές για n_clusters = [2, 4, 8]

```
from sklearn.cluster import KMeans

importances_list = []
cluster_severity_list = []
n_clsuters = [2, 4, 8]
for n in n_clsuters:
    kmeans = KMeans(n_clusters=n)
    kmeans.fit(X_scaled)
    importances = get_variable_importances(X_scaled, kmeans.labels_)
    importances_list.append(pd.DataFrame(data=importances, columns=columns))
    cluster_severity_list.append(pd.DataFrame(data={"Cluster": kmeans.labels_, "Severity": y}))
```

Παρακάτω παρουσιάζονται ένα heatmap με τους συντελεστές των μεταβλητών για κάθε συστάδα, καθώς και ένα ραβδόγραμμα με τα πλήθη των κατηγοριών Severity για κάθε συστάδα. Το πρώτο διάγραμμα μας προσφέρει πληροφορίες σχετικά με τη βαρύτητα κάθε μεταβλητής σε κάθε συστάδα, π.χ. η συστάδα Α χαρακτηρίζεται κυρίως από την θερμοκρασία (Temperature), την απόσταση των επιπτώσεων (Distance) και την ύπαρξη σηματοδότη Stop (Stop). Το δεύτερο διάγραμμα μας δίνει μια δευτερεύον πληροφορία όσον αφορά τη δυνατότητα της συσταδοποίησης χωρίς επίβλεψη να δημιουργήσει ομάδες που να εκφράζουν την σοβαρότητα του ατυχήματος μέχρι κάποιο βαθμό.



Για n_clusters=2 παρατηρούμε πως και οι δυο συστάδες χαρακτηρίζονται από το Visibility και το Weather_Condition. Για n_clusters=4 έχουμε τις συστάδες 0 και 2να χαρακτηρίζονται από τα χαρακτηριστικά, όμως με ορισμένες διαφορές στις τιμές των συντελεστών. Η τρίτη συστάδα διαφέρει από τις υπόλοιπες και βασίζεται έντονα στο Weather_Condition και το Visibility, ενώ η τέταρτη χαρακτηρίζεται από την πλευρά του ατυχήματος. Για n_clusters=8 παρατηρούμε ιδιαίτερους συνδυασμούς σε κάθε συστάδα, χωρίς ιδιαίτερες επαναλήψεις, πράγμα το οποίο σημαίνει πως οι συστάδες «ξεχωρίζουν» αρκετά καλά μεταξύ τους. Αξίζει να σημειωθεί πως σε όλες τις δοκιμές, οι δυαδικές τιμές δεν λαμβάνουν καθόλου βαρύτητα που επιβεβαιώνει τα προηγούμενα σχόλια περί συσταδοποίησης με Boolean μεταβλητές. Τελευταία αλλά εξίσου σημαντική παρατήρηση αποτελεί η όμοιες κατανομές του Severity στις συστάδες. Τα μοντέλα δεν κατάφεραν να δημιουργήσουν συστάδες που να εκφράζουν τη σοβαρότητα με αυτές τις παραμετροποιήσεις, δίνοντας πάτημα για την επόμενη ενότητα.

3.3. Συσταδοποίηση χωρίς επίβλεψη

Στόχος μας σε αυτή τη διαδικασία είναι να δημιουργήσουμε τρία μοντέλα συσταδοποίησης τα οποία να ταξινομούν τα δεδομένα ως προς την σοβαρότητα του ατυχήματος (Severity). Λάβαμε μια πρώτη εικόνα με σχετικές συσταδοποιήσεις στο προηγούμενο βήμα, ωστόσο χωρίς ιδιαίτερα αποτελέσματα από την πλευρά της ταξινόμησης. Σε αυτή τη διαδικασία θα εκτελέσουμε ορισμένες τεχνικές μείωσης των διαστάσεων ώστε να αντιμετωπίσουμε το πρόβλημα των Boolean μεταβλητών. Παρόλα αυτά, οι διαδικασίες μείωσης διαστάσεων, καθώς και οι αλγόριθμοι συσταδοποίησης, δεν ήταν αποδοτικές ως προς τον χρόνο εκτέλεσης, αφού ο όγκος των δεδομένων είναι τεράστιος (> 2εκ δείγματα!). Παρατηρώντας, παράλληλα, και την ανισορροπία στις κλάσεις αποφασίσαμε πως ο RandomUnderSampler, ο οποίος μειώνει τυχαία το πλήθος των δειγμάτων κάθε κλάσης μέχρι να ισούται με το πλήθος της μειονοτική κλάσης, θα ήταν καλή λύση. Πέτυχαμε να μειώσουμε τα δείγματα στα 94 χιλιάδες, όμως οι αλγόριθμοι αδυνατούσαν πάλι να αποδώσουν σε εύλογο χρόνο εκτέλεσης. Οπότε, μειώσαμε ξανά το πλήθος σε 20 χιλιάδες.

```
# Reduces to 94224 samples
sampler = RandomUnderSampler()
X_reduced, y_reduced = sampler.fit_resample(X_scaled, y)

# but we need to reduce more
N = 20_000
random_index = np.random.choice(X_reduced.shape[0], N, replace=False)
X_sampled, y_sampled = X_reduced[random_index], y_reduced[random_index]
```

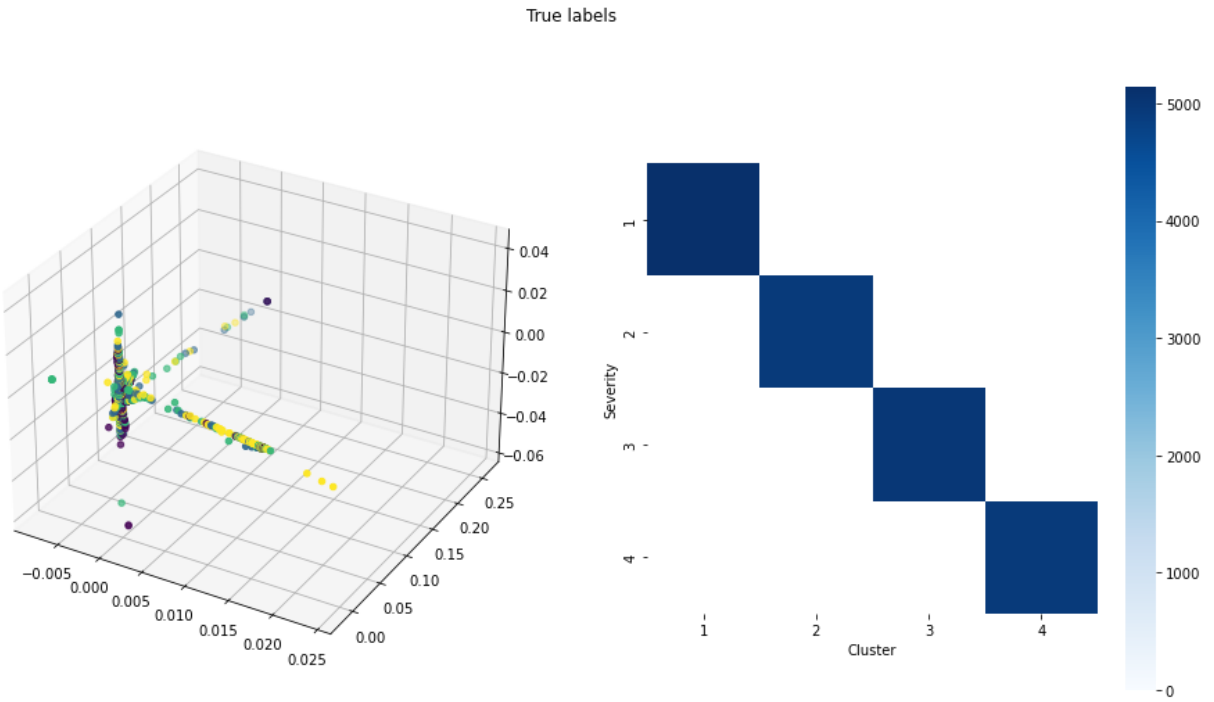
```
Size before: 2240862
Counts before: {1: 23556, 2: 2080281, 3: 71130, 4: 65895}

Size after: 20000
Counts after: {1: 5143, 2: 4925, 3: 4997, 4: 4935}
```

Έπειτα από δοκιμές τεχνικών μείωσης διαστάσεων, αλγορίθμων συσταδοποίησης και των αντίστοιχων παραμέτρων τους, καταλήξαμε στα εξής: Locally Linear Embedding (n_components=3) για μη γραμμική μείωση διαστάσεων διατηρώντας τις τοπικές αποστάσεις, KMeans (n_clusters=6), DBSCAN (eps=0.0015, min_samples=25) και AgglomerativeClustering (n_clusters=6, linkage='ward'). Να σημειωθεί πως δεν είναι αναγκαίο το n_clusters να είναι ίσο με το πλήθος των κλάσεων που θέλουμε να ταξινομήσουμε,

δηλαδή 4. Αντιθέτως, μπορεί να είναι οποιοδήποτε νούμερο, αρκεί, σε μια επιτυχημένη συσταδοποίηση, να συγχωνεύσουμε τις συστάδες που αντιστοιχούν στην εκάστοτε κλάση. Π.χ. η συστάδα Α αντιστοιχεί στην κλάση 1, η συστάδα Β στην κλάση 2 και η συστάδα Γ στην κλάση 1, μπορούμε να πούμε πως το δείγμα ανήκει στην κλάση 1 αν ταξινομηθεί στην συστάδα Α ή Γ.

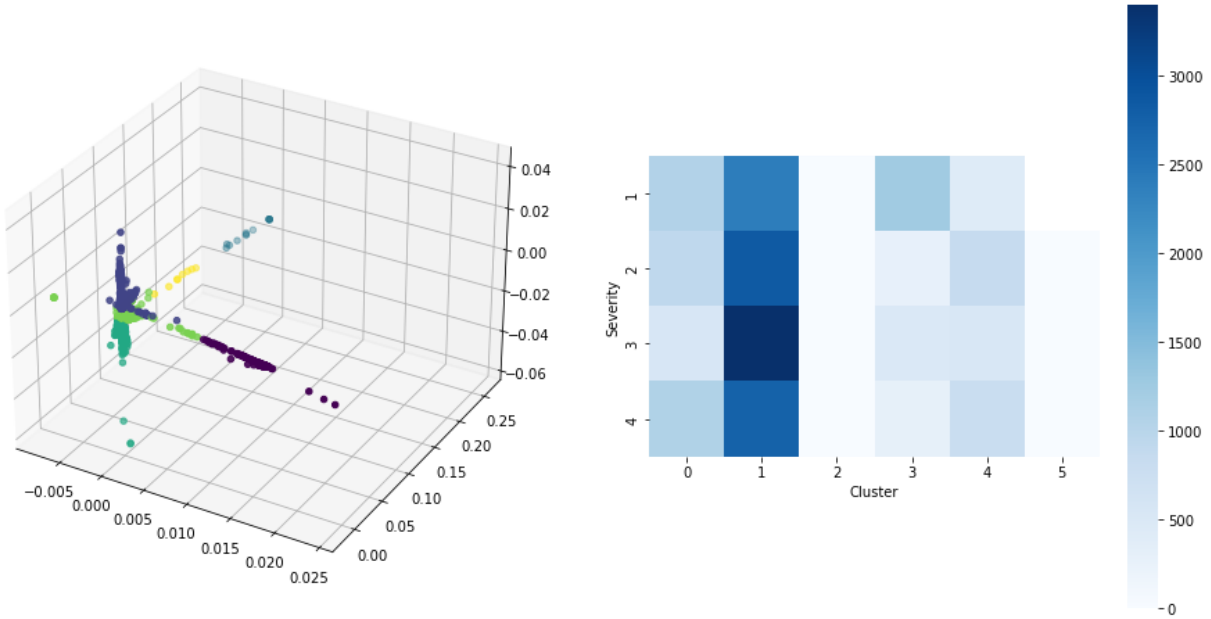
Για την οπτικοποίηση των αποτελεσμάτων χρησιμοποιήθηκαν scatterplots των μειωμένων δεδομένων με χρώμα βάσει της συστάδα και heatmaps των συστάδων έναντι των κλάσεων. Αρχικά αξίζει να παρουσιαστεί ο μειωμένος χώρος που παράχθηκε από τον LLE.



Αξιόλογη είναι η μη διαχωρισιμότητα των κλάσεων, όπως φαίνεται στο scatterplot. Μας προϊδεάζει πως σε αυτό το χώρο είναι δύσκολη η δημιουργία ενός αποδοτικού μοντέλου. Το heatmap, σε αυτή την περίπτωση δεν χρησιμεύει, όμως μπορούμε να σχολιάσουμε πως μια ικανοποιητική συσταδοποίηση θα είχε ένα αντίστοιχο αποτέλεσμα, χωρίς να είναι απαραίτητη η διαγώνιος.

Παρακάτω φαίνονται τα αποτελέσματα του KMeans.

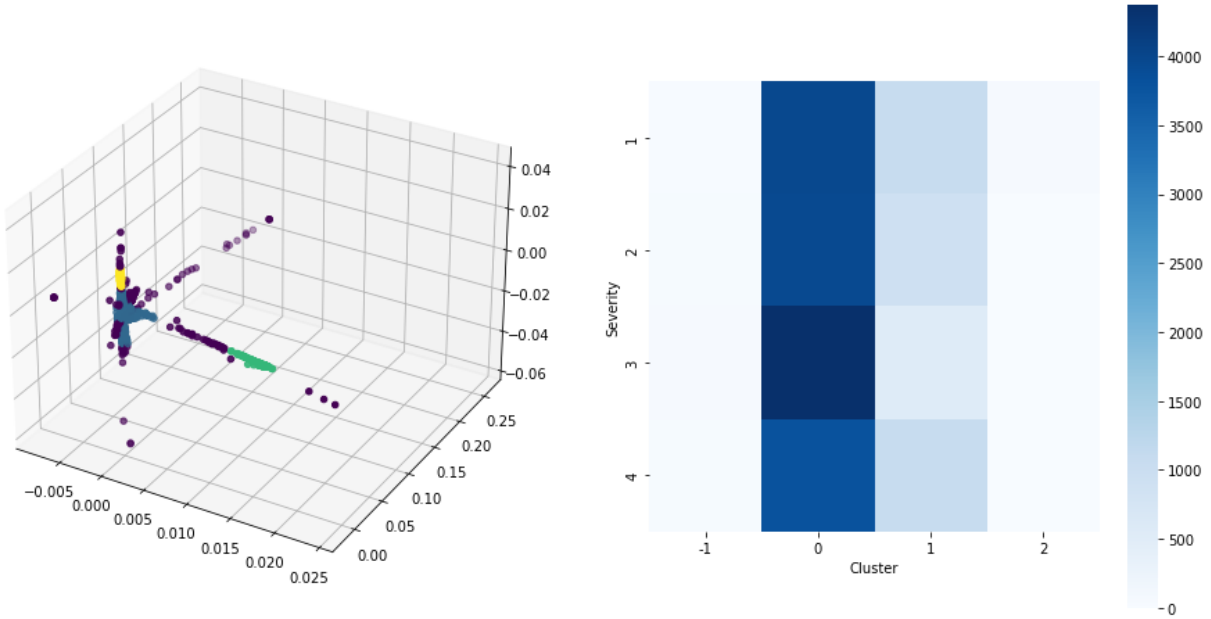
KMeans labels



Είναι φανερό πως τα αποτελέσματα της συσταδοποίησης δεν αντιστοιχούν στις πραγματικές κλάσεις. Η συστάδα 1 περιέχει τα περισσότερα δεδομένα, τα οποία ανήκουν σε όλες τις κλάσεις.

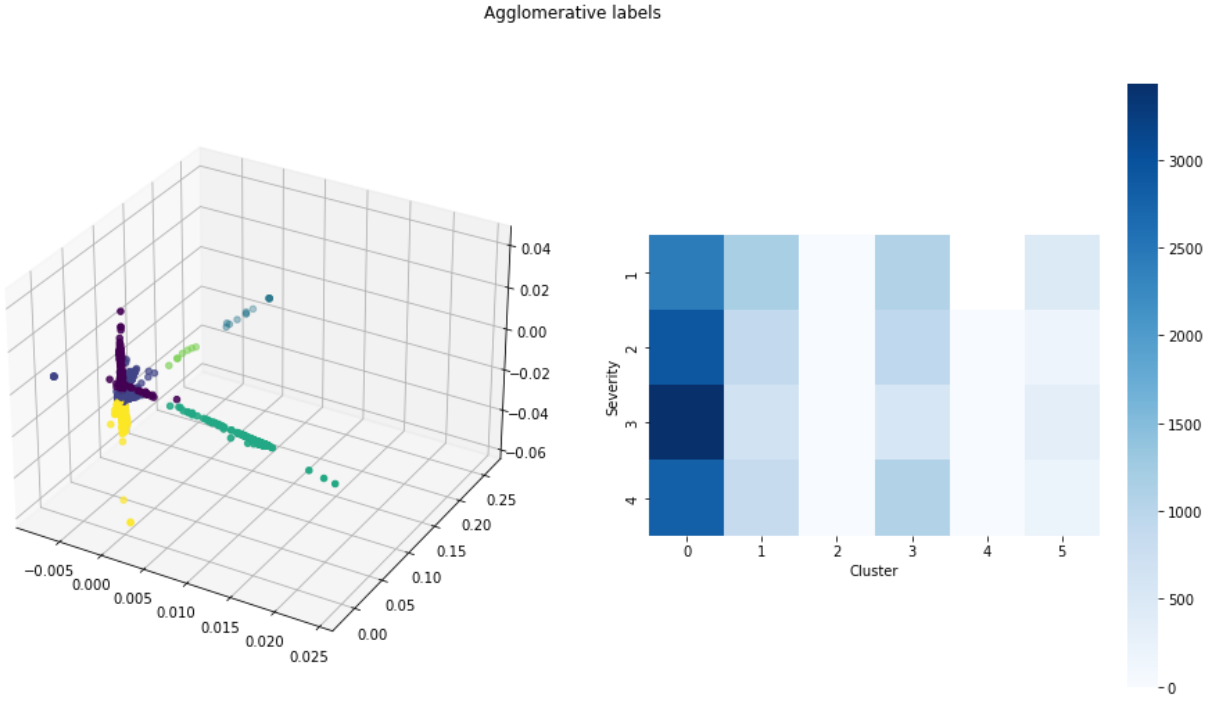
Όμοια αποτελέσματα λαμβάνουμε και από το DBSCAN.

DBSCAN labels



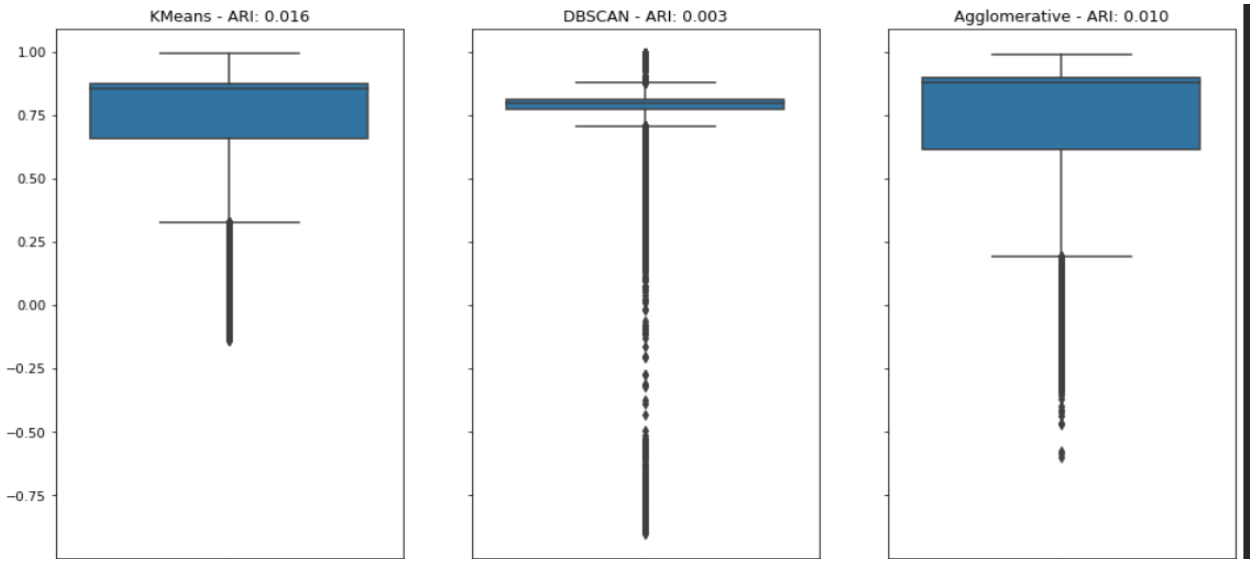
Σε αυτή την περίπτωση η κλάση -1 αποτελεί το θόρυβο που αναγνώρισε ο αλγόριθμος. Τα περισσότερα δεδομένα ανήκουν στην συστάδα 0, ανεξαρτήτως πραγματικής κλάσης.

Τέλος, τα αποτελέσματα του AgglomerativeClustering.



Ομοίως, τα αποτελέσματα δεν είναι ικανοποιητικά.

Τέλος, χρησιμοποιούμε δύο δείκτες που αφορούν την ποιότητα της συσταδοποίησης, Silhouette Score και Adjusted Rand Index. Ο πρώτος χρησιμοποιείται χωρίς τις πραγματικές κλάσεις και θα μπορούσαμε να πούμε με απλά λόγια πως μετρά την διαχωρισημότητα των συστάδων. Στην περίπτωση μας, υπολογίσαμε το silhouette score για κάθε δείγμα και δημιουργήσαμε boxplots για κάθε μοντέλο. Ο Adjusted Rand Index μετρά την ομοιότητα των ετικετών των συστάδων και των ετικετών των πραγματικών κλάσεων, αγνοώντας τις μεταθέσεις των ετικετών.



Παρόλο που και τα τρία μοντέλα παρουσιάζουν όμοια διάμεση τιμή shillouette score, οι κατανομές τους διαφέρουν αρκετά. Οι υψηλές τιμές των Q1, Q2, Q3 μπορεί να παραπλανήσουν κάποιον πως υπάρχει καλή συσταδοποίηση. Η αλήθεια είναι πως υποδηλώνουν εύρωστες και καλά ορισμένες συστάδες, πράγμα το οποίο συμβαίνει αν παρατηρήσουμε τα παραπάνω scatterplots. Όμως, αυτό δεν αρκεί και ο ARI μας δείχνει την πραγματικότητα, δηλαδή δεν έχει γίνει καλή αντιστοίχιση των ετικετών.

Για να μην είμαστε τελείως απαισιόδοξοι, τα αποτελέσματα δεν είναι ικανοποιητικά για αυτές τις συνθέσεις παραμέτρων. Ενδέχεται για μια μεγάλη τιμή n_clusters και κατάλληλη συγχώνευση στον KMeans να λύσει το πρόβλημα της μη διαχωρισημότητας, ή ένα άλλος συνδυασμός αλγορίθμου μείωσης διαστάσεων, με άλλον random_state και υπόλοιπες παραμέτρους να αποδώσουν καλύτερα αποτελέσματα.

4. Ταξινόμηση με Νευρωνικά Δίκτυα

Σε αυτή την ενότητα, παρουσιάζουμε τρία μοντέλα νευρωνικών δικτύων που εκπαιδεύονται με επίβλεψη για ταξινόμηση των δειγμάτων ως προς τη μεταβλητή στόχο Severity. Τα νευρωνικά δίκτυα δημιουργήθηκαν με την βιβλιοθήκη tensorflow. Πιο συγκεκριμένα, τα τρία μοντέλα διαφέρουν στις υπερ-παραμέτρους τους και συγκρίνεται η αποδοτικότητά τους.

4.1. Προεπεξεργασία δεδομένων

Για την διαδικασία της ταξινόμηση θα χρησιμοποιήσουμε τα ισορροπημένα δεδομένα της ενότητας 3.3 αφού η μεγάλη ανισορροπία των κλάσεων των αρχικών δεδομένων θα οδηγήσει σε overfitting. Σε αυτή την προεπεξεργασία, κωδικοποιούμε τις ετικέτες των κλάσεων σε One-Hot Encoding, δηλαδή μια λίστα μεγέθους ίσο με το πλήθος των κλάσεων (ν), στην οποία υπάρχουν ν-1 μηδενικά και 1 άσσος ανάλογα με την κλάση. Επιπλέον, χωρίζουμε τα δεδομένα μας σε 3 μέρη, δεδομένα για training, validation και testing με ποσοστά επί του συνολικού πλήθους 70%, 12% και 18% αντίστοιχα.

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split


enc = OneHotEncoder()
y_enc = enc.fit_transform(y_reduced.reshape(-1,1)).toarray()

X_train, X_val, y_train, y_val = train_test_split(X_reduced, y_enc, test_size=0.3)
X_val, X_test, y_val, y_test = train_test_split(X_val, y_val, test_size=0.6)
```

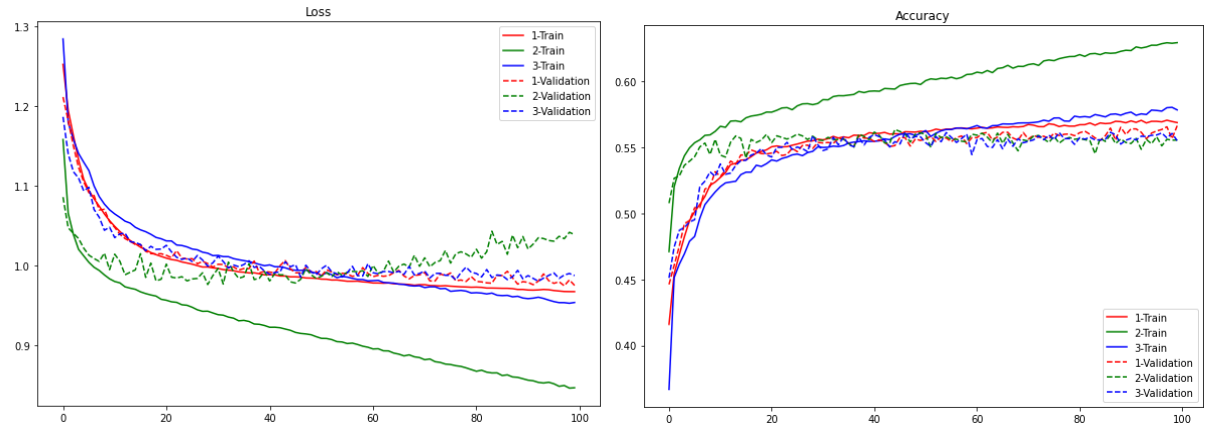
4.2. Εκπαίδευση μοντέλων

Αν και έγιναν διάφορες δοκιμές υπερπαραμέτρων, παρουσιάζονται τρεις που εμφανίζουν ενδιαφέροντα χαρακτηριστικά. Πιο συγκεκριμένα, τα μοντέλα είναι:

Name	Layer	Units	Activation	Batch_Size
model_1	Input	22	-	32
	Dense	50	sigmoid	
	Dense	35	sigmoid	
	Dense	4	softmax	

model_2	Input	22	-	68
	Dense	125	relu	
	Dense	86	relu	
	Dense	49	sigmoid	
	Dense	30	relu	
	Dense	4	softmax	
model_3	Input	22	-	25
	Dense	250	sigmoid	
	Dropout	0.2	-	
	Dense	160	tanh	
	Dense	35	sigmoid	
	Dense	23	tanh	
	Dense	45	sigmoid	
	Dropout	0.2	-	
	Dense	10	sigmoid	
	Dense	4	softmax	

Σε όλα τα μοντέλα χρησιμοποιήθηκαν ο optimizer adam, το loss categorical crossentropy, το metric accuracy και 100 epochs. Αναλυτικότερα, ο adam αποτελεί μια πολύ συχνή και καλή επιλογή για optimizer. Χρησιμοποιούμε categorical crossentropy αφού στόχος μας είναι η ταξινόμηση με one hot ετικέτες. Το accuracy αποτελεί καλή επιλογή για παρακολούθηση της εκπαίδευσης αφού τα δεδομένα μας είναι ισορροπημένα. Θέσαμε epochs=100 σε όλα τα μοντέλα για να παρατηρήσουμε από κοινού την πορεία των losses και accuracies.



Μετά την εκπαίδευση των μοντέλων, λαμβάνουμε τις τιμές των losses και accuracies και τα εμφανίζουμε σε διάγραμμα. Παρατηρούμε πως το model_3 ξεκινάει με χαμηλό loss και υψηλό accuracy, τόσο στο training όσο και στο validation. Πολύ γρήγορα, όμως, οι γραμμές train και validation αποκλίνουν μεταξύ τους, πράγμα το οποίο υποδεικνύει overfitting, δηλαδή το μοντέλο «μαθαίνει» πάρα πολύ καλά τα δεδομένα εκπαίδευσης όμως δεν μπορεί να γενικευτεί και να αποδώσει καλά σε άγνωστα δεδομένα. Οι καμπύλες των δεικτών των άλλων μοντέλων ακολουθούν όμοια πορεία μεταξύ τους, όμως το model_2 (μπλε γραμμή) παρουσιάζει μικρή απόκλιση (training με validation_ όσο περνάνε τα epochs. Τελικά, όλα

τα μοντέλα παρουσιάζουν όμοια σύγκλιση του validation accuracy, παρόλο που οι training μετρικές δεν ακολουθούν ίδια σύγκλιση.

Μένει να συγκρίνουμε τα μοντέλα ως προς την απόδοσή τους σε άγνωστα δεδομένα, δηλαδή ταξινομώντας το υποσύνολο δεδομένων test. Για αυτή τη διαδικασία θα μας βοηθήσει η συνάρτηση classification_report της βιβλιοθήκης scikit-learn, η οποία δέχεται τις πραγματικές και τις εκτιμώμενες κλάσεις και εκτυπώνει έναν πίνακα με μετρικές precision, recall, f1-score για κάθε κλάση και το συνολικό accuracy. Έτσι, έχουμε:

model_1

	precision	recall	f1-score	support
0	0.70	0.76	0.73	4292
1	0.53	0.44	0.48	4297
2	0.50	0.50	0.50	4144
3	0.48	0.52	0.50	4228
accuracy			0.56	16961
macro avg	0.55	0.55	0.55	16961
weighted avg	0.55	0.56	0.55	16961

model_2

	precision	recall	f1-score	support
0	0.68	0.81	0.74	4292
1	0.51	0.49	0.50	4297
2	0.49	0.46	0.48	4144
3	0.50	0.44	0.47	4228
accuracy			0.55	16961
macro avg	0.54	0.55	0.55	16961
weighted avg	0.54	0.55	0.55	16961

model_3

	precision	recall	f1-score	support
0	0.65	0.82	0.73	4292
1	0.51	0.49	0.50	4297
2	0.50	0.50	0.50	4144
3	0.52	0.41	0.46	4228
accuracy			0.56	16961
macro avg	0.55	0.55	0.55	16961
weighted avg	0.55	0.56	0.55	16961

Είναι προφανές πως οι τιμές των μετρικών είναι σχεδόν ίδιες μεταξύ των μοντέλων! Ένα απλό μοντέλο (model_1) έχει την ίδια απόδοση με ένα περίπλοκο (model_3) ενώ το κόστος εκπαίδευσης του πρώτου είναι πολύ χαμηλότερο από τα υπόλοιπα. Βέβαια, αναρωτιόμαστε γιατί το περίπλοκο μοντέλο δεν κατάφερε να αποδώσει καλύτερα. Δεν υπάρχει σίγουρη και πλήρης απάντηση. Δυο πιθανές απαντήσεις

είναι πως το πρόβλημα είναι αρκετά περίπλοκο από τη φύση του ή δεν διαθέτουμε τις κατάλληλες μεταβλητές και υπερπαραμέτρους για μια αποτελεσματική ταξινόμηση.

4.3. Καλύτερο μοντέλο και χάρτης εκτίμησης

Ως καλύτερο μοντέλο θα επιλέξουμε το πρώτο αφού μεταξύ ίσων αποδόσεων, το επόμενο κριτήριο επιλογής θα μπορούσε να είναι ο αριθμός των παραμέτρων ή η απλότητα του μοντέλου. Με βάση, λοιπόν, αυτό το μοντέλο στόχος μας είναι να δημιουργήσουμε ένα χάρτη εκτίμησης της μέση σοβαρότητας ατυχήματος. Για την διαδικασία αυτή, χωρίζουμε το χάρτη σε ένα πλέγμα 80χ80.

```
# Create a 80x80 grid
map_extend_min = data_df[['Start_Lat', 'Start_Lng']].min()
map_extend_max = data_df[['Start_Lat', 'Start_Lng']].max()

xx = np.linspace(map_extend_min['Start_Lng'], map_extend_max['Start_Lng'], 80)
yy = np.linspace(map_extend_min['Start_Lat'], map_extend_max['Start_Lat'], 80)
```

Στη συνέχεια, για κάθε κελί επιλέγουμε τις εγγραφές που ανήκουν γεωγραφικά σε αυτό και παίρνουμε τη μέση τιμή των χαρακτηριστικών τους. Για τα Boolean χαρακτηριστικά, αν η μέση τιμή είναι μεγαλύτερη η ίση του 0.5, τότε αλλάζουμε την αντίστοιχη τιμή σε 1, διαφορετικά 0.

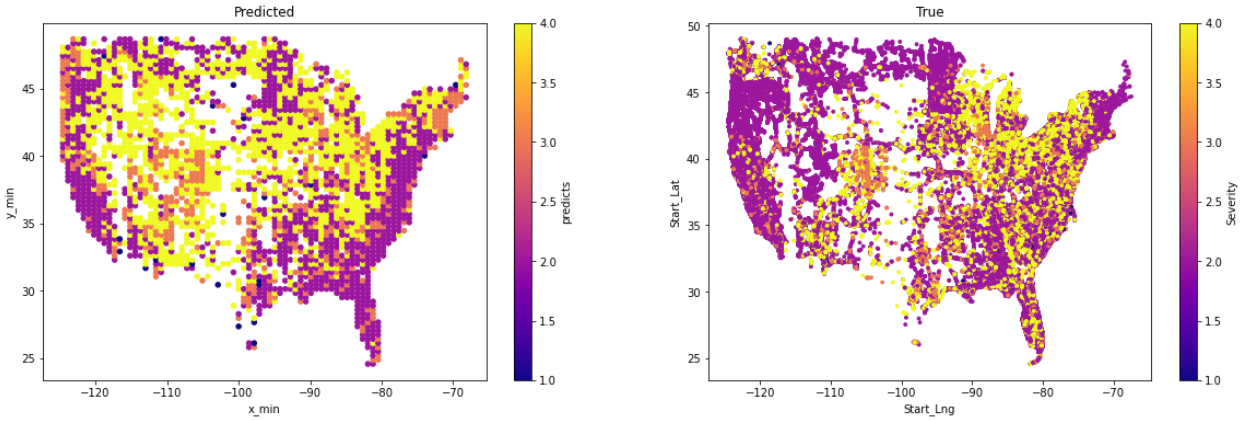
```
map_avg = pd.DataFrame(columns=(['x_min', 'x_max', 'y_min', 'y_max'] + columns))
k = 0
# For each cell, find the records that are within the cell and calculate the average of each attribute
for i in range(len(xx[:-1])):
    for j in range(len(yy[:-1])):
        selected_records = data_df.loc[(data_df['Start_Lng'] > xx[i]) &
                                       (data_df['Start_Lng'] < xx[i+1]) &
                                       (data_df['Start_Lat'] > yy[j]) &
                                       (data_df['Start_Lat'] < yy[j+1])]
        avg_data = selected_records[columns].mean()
        if avg_data[columns].isna().sum() != len(columns):
            avg_data[columns] = 0

        map_avg.loc[k, ['x_min', 'x_max', 'y_min', 'y_max']] = [xx[i], xx[i+1], yy[j], yy[j+1]]
        map_avg.loc[k, columns] = avg_data
        k += 1

# Ignore cells that do not have any record
map_avg_dropped = map_avg.loc[map_avg[columns].sum(axis=1) != 0]

# Set 1 if the mean of a boolean attribute is >= 0.5, otherwise 0
map_avg_dropped.loc[:, bool_lbls] = map_avg_dropped[bool_lbls].applymap(lambda v: 1 if v >= 0.5 else 0)
```

Ύστερα, προετοιμάζουμε τα δεδομένα για πρόβλεψη κάνοντας την κατάλληλη προεπεξεργασία, εκτελούμε την πρόβλεψη και λαμβάνουμε τις εκτιμώμενες τιμές για την σοβαρότητα του ατυχήματος για κάθε κελί. Τέλος, για κάθε κελί εμφανίζουμε μια κουκίδα με χρώμα ανάλογα τις εκτιμώμενης τιμής.



Παρατηρούμε πως η εκτίμησή μας έχεις αρκετή διαφορά από τα δεδομένα μας. Αυτό οφείλεται στην κακή απόδοση του ταξινομητή μας (56%). Με ένα καλύτερο μοντέλο θα ήταν δυνατή η δημιουργία χρήσιμων χαρτών με τις πιθανότητες της σοβαρότητας των ατυχημάτων ώστε οι αρχές να έχουν την αντίστοιχη ετοιμότητα.

Τέλος, χρήσιμη είναι η κατανομή των πιθανοτήτων για κάθε κλάση. Παρατηρούμε ότι στις περισσότερες φορές το μοντέλο μας δεν είναι «σίγουρο» για την εκτίμησή του. Αυτό φαίνεται από τα Q1, Q2 και Q3 τα οποία βρίσκονται σε όλες τις περιπτώσεις κάτω από 50%! Εμείς, όμως, αναθέτουμε μια κλάση με βάσει της υψηλότερης πιθανότητας. Για παράδειγμα, αν το διάνυσμα εξόδου είναι [0.32, 0.26, 0.13, 0.29] τότε θα κατατάξουμε το δείγμα στην πρώτη κλάση με πιθανότητα 32%, ενώ στην πραγματικότητα θα μπορούσε να είναι η τέταρτη κλάση με 29%.