Metodi Predittivi per l'Azienda

Analisi sugli incidenti in America

Introduzione

Il dataset preso in esame è un dataset contenente i dati relativi agli incidenti stradali avvenuti in America (il dataset copre 49 stati degli USA) dal Febbraio del 2016 fino al Dicembre del 2020.

I dati sono stati ottenuti mediante l'utilizzo di API che scaricano le informazioni da diverse entità differenti come il dipartimento dei trasporti americano, le forze dell'ordine, le telecamere del traffico ed i sensori posti sulle strade.

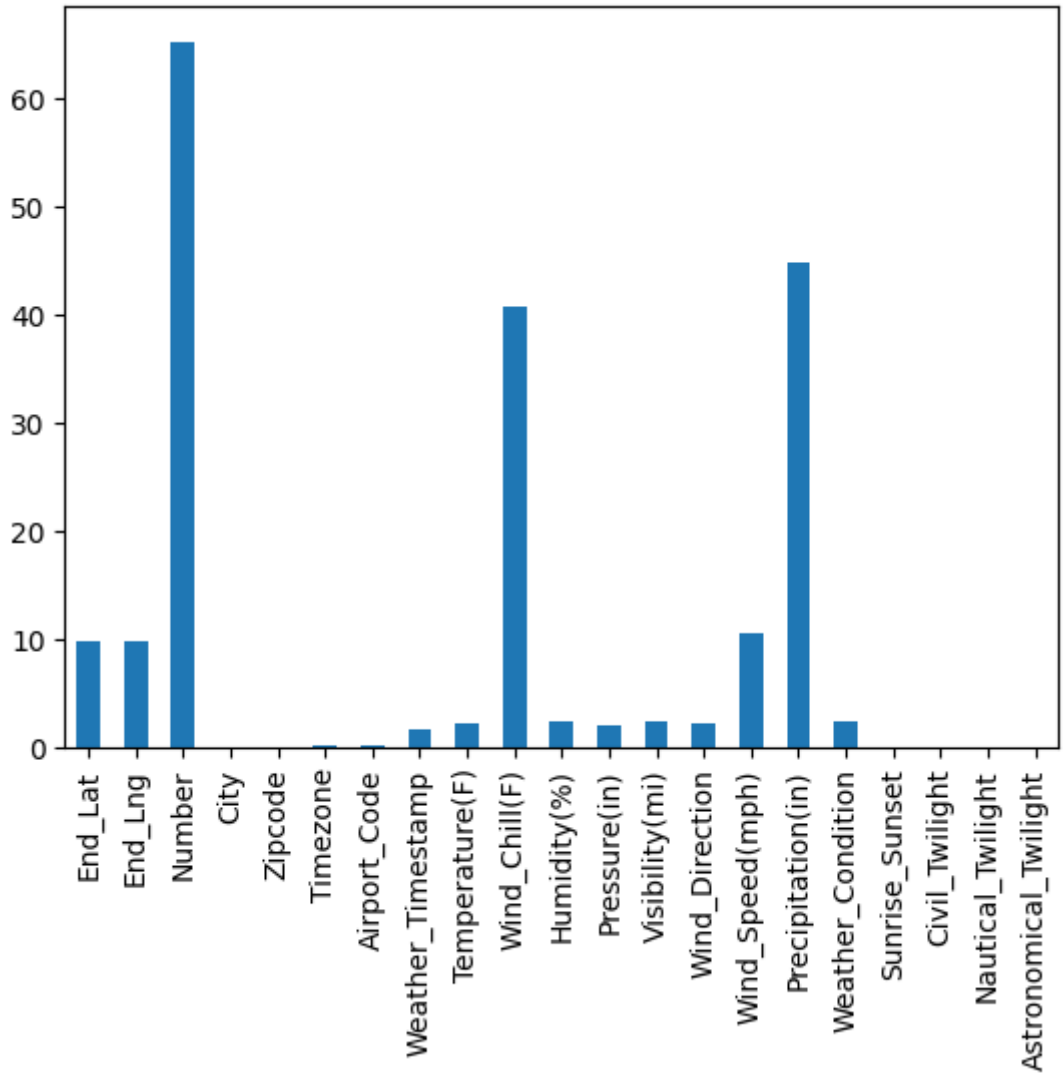
Il dataset fornisce, per ogni incidente, molte informazioni tra cui una classificazione della gravità dell'incidente, i dati relativi ad ora e luogo dell'incidente, la quantità di coda generata dall'incidente e la durata di questa, le condizioni atmosferiche al momento dell'incidente e l'indicazione della presenza di eventuali condizioni particolari della strada (es. incroci, rotonde ecc).

Obiettivi

Vista la grande quantità di informazioni contenute nel dataset, questo si presta per molte applicazioni.
Il mio primo obiettivo è quello di analizzare le informazioni e di riconoscere eventuali fattori che possano condizionare la probabilità che si verifichi un incidente. Il secondo obiettivo è quello di trainare un algoritmo per poter classificare la gravità di un incidente a partire dai fattori che lo hanno scaturato. Queste funzioni potrebbero rivelarsi decisamente importanti per le forze dell'ordine per poter analizzare e comprendere, in base alle strade e alle condizioni atmosferiche della giornata, quali saranno i posti dove porre maggior attenzioni ed effettuare più controlli al fine di diminuire la quantità di incidenti e diminuire il numero di code generate da questi.

Processo

Per prima cosa, dopo aver caricato il dataset, ho deciso di analizzare per ogni elemento del dataset la percentuale di dati nulli al fine di eliminare gli elementi inutili e completare le informazioni mancanti, di seguito il grafico con le percentuali di valori nulli per ogni elemento:

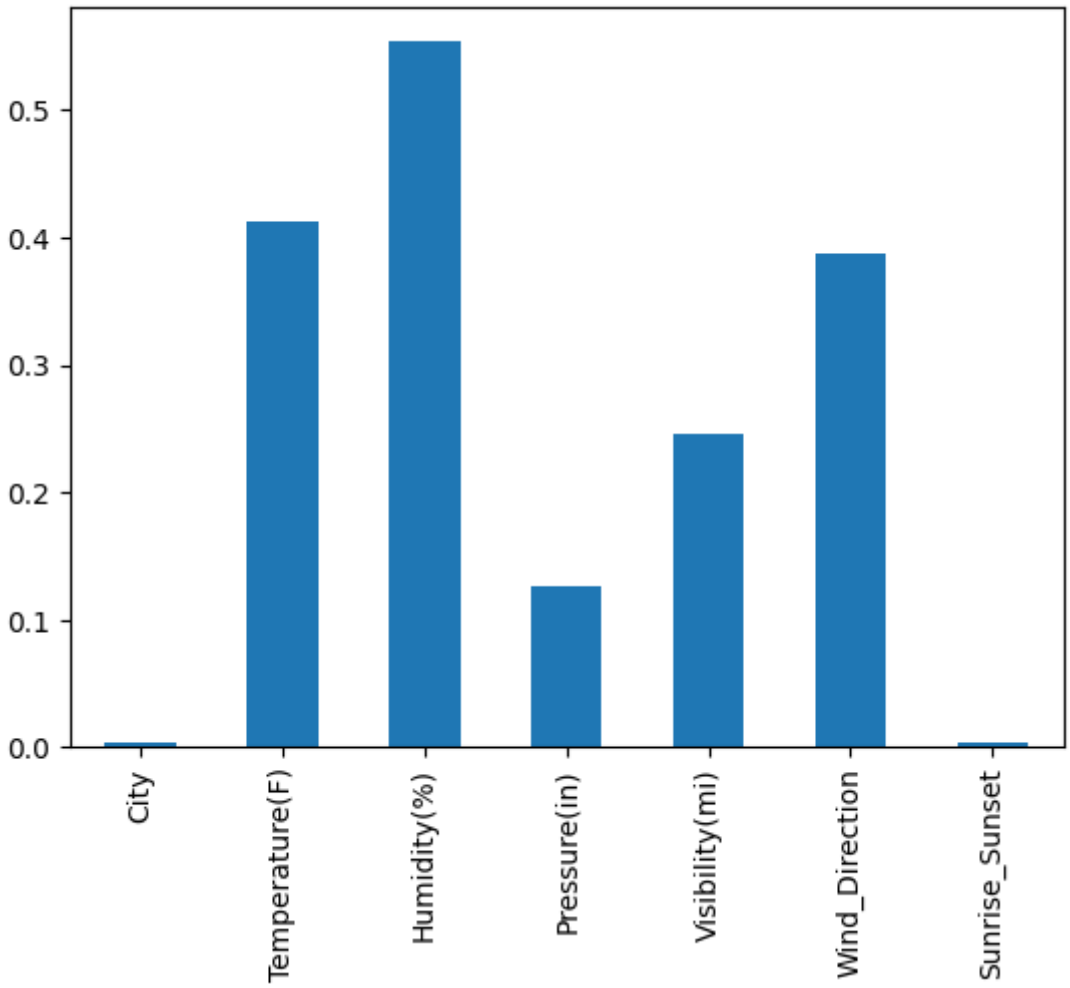


Notiamo come gli elementi che hanno più valori nulli siano: Number, Wind_Chill(F), Precipitation(in), End/Lat/Long, Wind_Speed(mph). Per ognuno di questi, dopo aver analizzato il tipo di dato e l'informazione contenuta ho deciso di:

- Number: numero civico vicino al quale si è verificato l'incidente. Non contenendo informazione utile ho deciso di eliminare il dato.
- Wind_Chill(F): temperatura del vento. Avendo già le informazioni relative alla temperatura atmosferica ed avendo questo dato il 40% di valori nulli ho deciso di eliminarlo.
- Precipitation(in): quantità di precipitazione. Questo dato risulta molto importante ai fini delle successive analisi, ma quasi il 50% dei valori risulta nullo. Essendo presente un dato con la descrizione testuale delle condizioni atmosferiche ho provato ad incrociare i dati per vedere se a valore nullo corrispondesse una giornata senza precipitazioni ma così non è stato. Ho deciso così di eliminare il dato e sostituirlo con un dato booleano, ottenuto mediante la classificazione delle condizioni atmosferiche, che rappresenta se in quella giornata piovesse/nevicasse o meno.
- End/Lat/Long: coordinate di fine incidente. Ho eliminato il dato.
- Wind_Speed(mph): velocità del vento. In questo caso, essendo i dati mancanti ~10% ed essendo un'informazione importante ai fini delle analisi, ho deciso di basarmi nuovamente sulla descrizione delle condizioni atmosferiche. Infatti in questo campo compaiono i valori

"Windy" e "Tornado" che presuppongono una giornata ventosa. Il mio approccio è stato quindi quello di generare una variabile booleana che descrivesse se la giornata fosse ventosa e quindi di calcolare il valore medio della velocità del vento nelle giornate ventose ed il valore medio della velocità del vento nelle giornate non ventose. Di conseguenza ho selezionato gli elementi con valori nulli e sostituiti con le rispettive medie a seconda che fossero giornate ventose o meno (ho notato che tutti i dati mancanti erano relativi a giornate non ventose).

In seguito ho eliminato i valori non necessari per le successive analisi. Di seguito il grafico delle percentuali di valori nulli dopo le modifiche di sopra:



Ho quindi proceduto rimuovendo i dati con valori nulli per gli elementi:

- City
- Wind_Direction
- Sunrise_Sunset

Ed inserendo i valori medi per gli elementi:

- Temperature
- Humidity
- Visibility
- Pressure

In questo modo ho eliminato tutti i valori nulli all'interno del dataset, mantenendo comunque un grosso numero di dati, ovvero poco meno di 3 milioni di incidenti quando il dataset iniziale ne conteneva 3 milioni (ora 2.82 milioni, prima 2.9 milioni).

Dopodiché ho eseguito un pò di trasformazioni sui dati ed ho convertito tutti gli object type in interi.

A questo punto ho effettuato delle analisi sui dati per estrarre informazioni e comprendere se vi sono eventuali fattori che possano condizionare la probabilità che si verifichi un incidente.

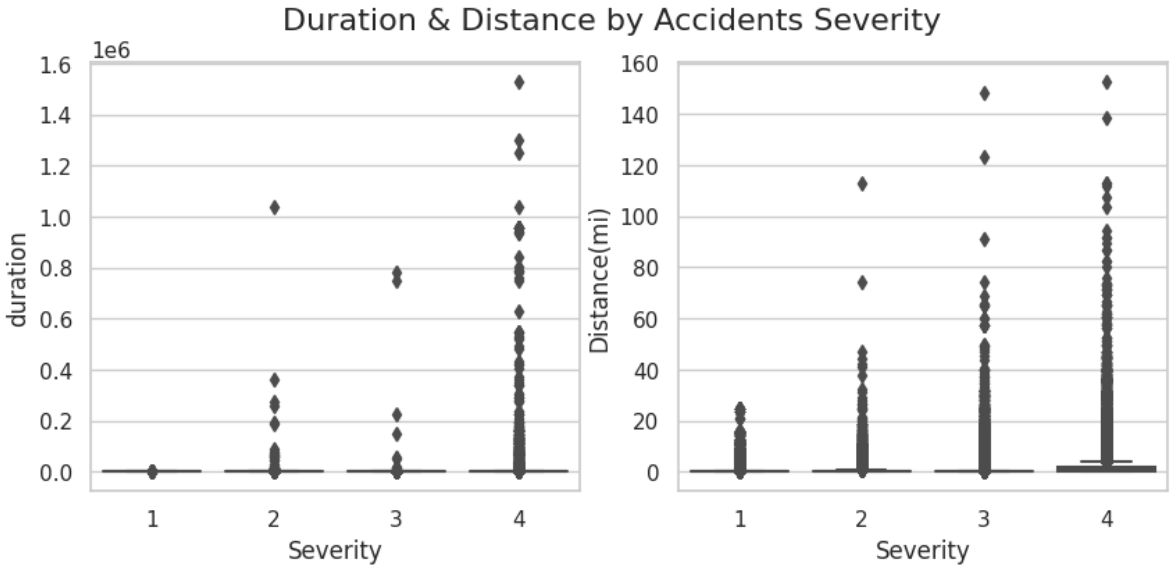
Prima di tutto ho calcolato la durata media degli incidenti: ~74 minuti.

In seguito ho calcolato per ogni grado di gravità quanti incidenti fossero presenti all'interno del dataset:

1° grado: 28182
2° grado: 2068650
3° grado: 611976
4° grado: 114875

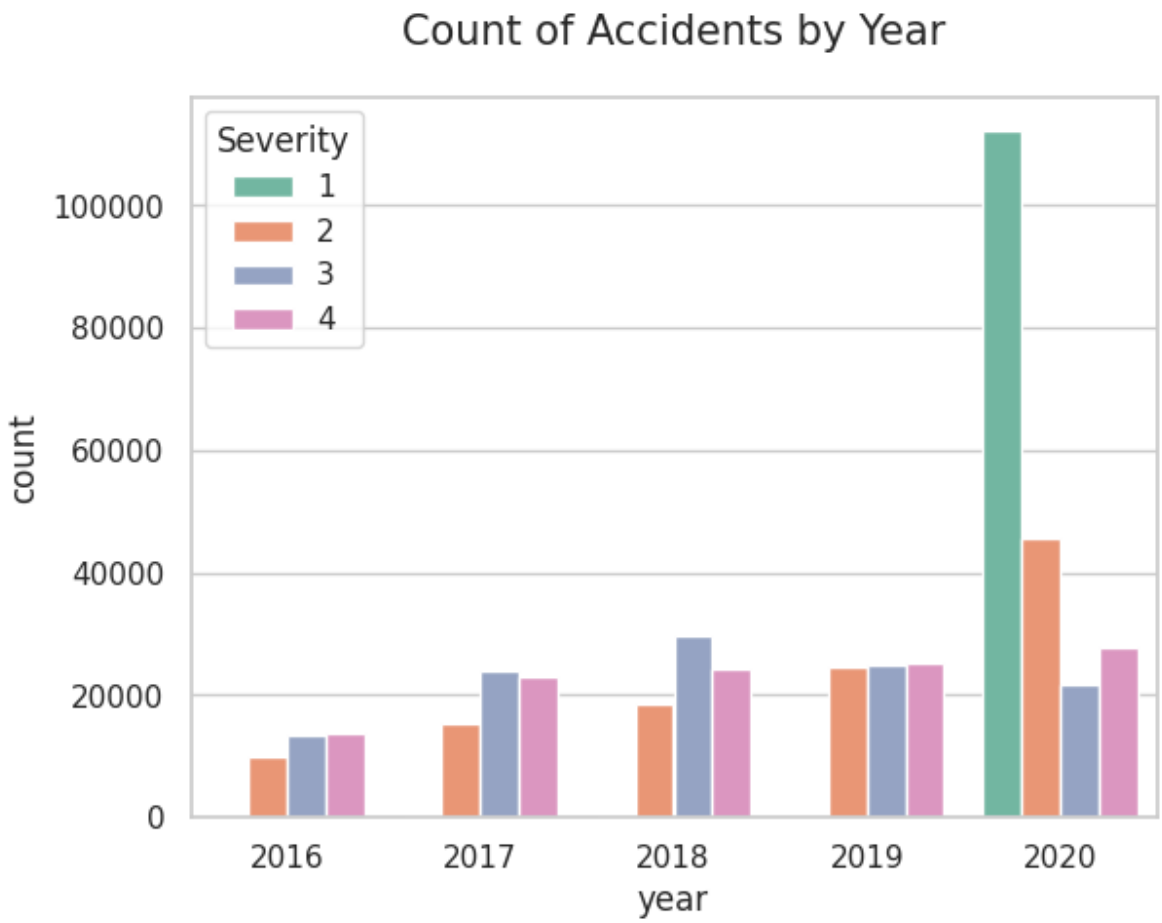
Come possiamo notare gli incidenti di 4° grado sono molto minori rispetto per esempio a quelli di 2° grado. Al fine di effettuare delle analisi che confrontino i dati relativi ai 4 differenti gradi di gravità ho deciso di effettuare una combinazione di oversampling e undersampling per compensare la differenza di numero di incidenti, nello specifico ho selezionato 114.000 incidenti per ogni categoria.
Di seguito le evidenze raccolte.

Durata e lunghezza della coda

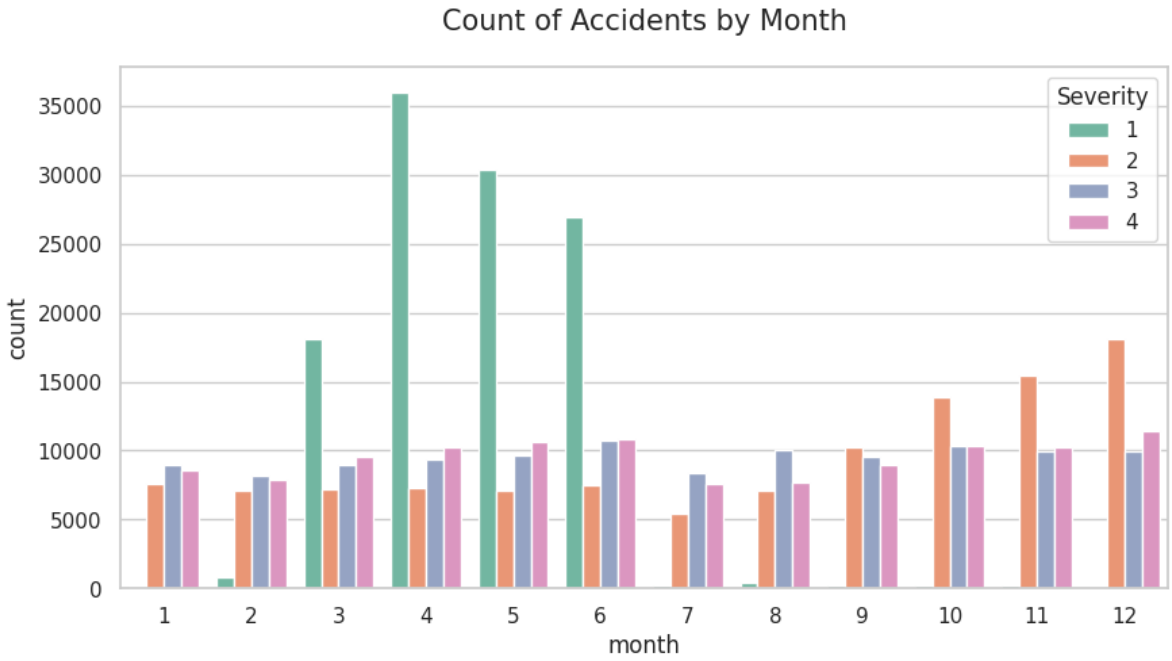


Possiamo notare un andamento crescente rispetto alla gravità dell'incidente per quanto riguarda la distanza delle code generate dagli incidenti e la durata.

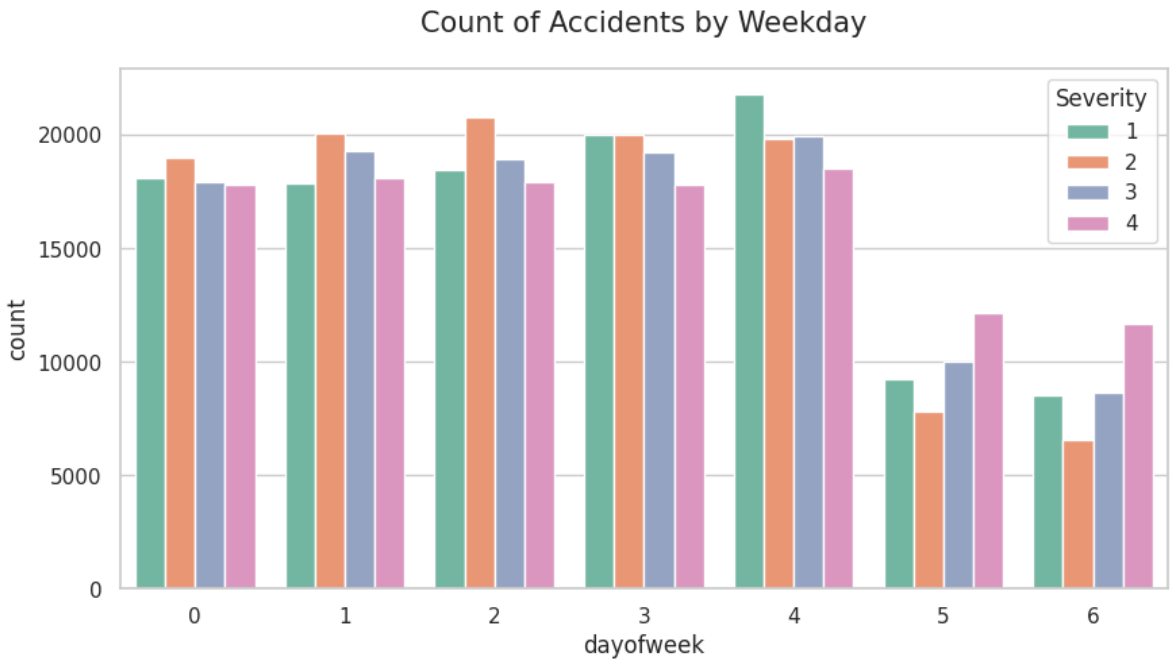
Incidenti in relazione a anno, mese, giorno e ora



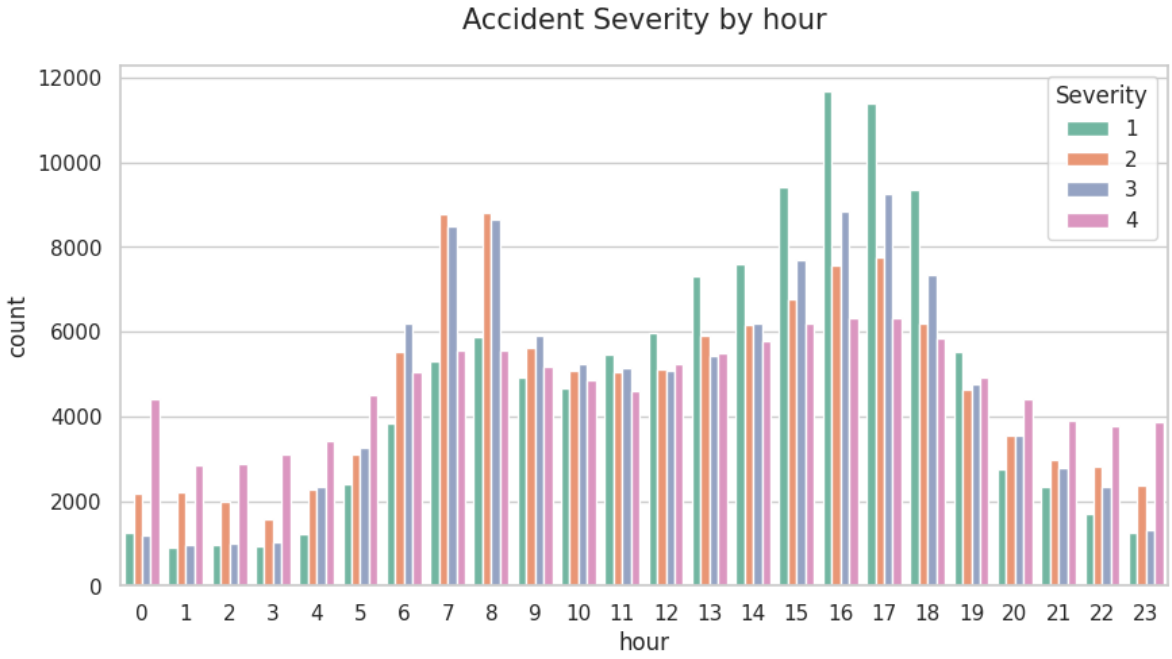
Possiamo notare che nell'anno 2020 sono stati inseriti nel dataset gli incidenti di grado 1 e notiamo inoltre una tendenza in aumento dei casi di incidenti di grado 4.



Notiamo che Dicembre è il mese in cui si verificano più incidenti gravi nell'arco di un anno.



Notiamo che in settimana il numero di incidenti è maggiore rispetto al weekend, ma nel weekend è più probabile che si verifichino incidenti gravi.



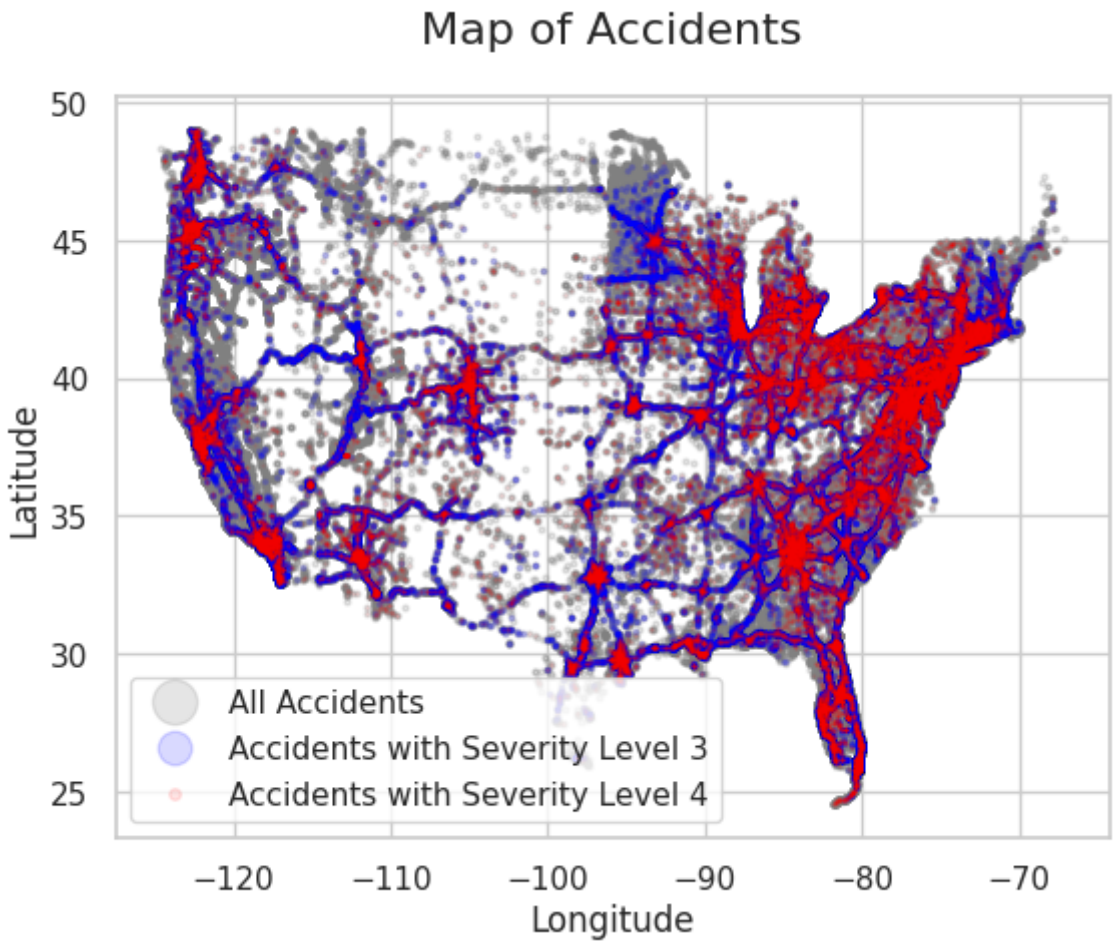
Anche qui notiamo che durante il giorno si verificano più incidenti, ma durante la notte è più probabile che si verifichino incidenti gravi.

Relazione della gravità degli incidenti con il tipo di strada

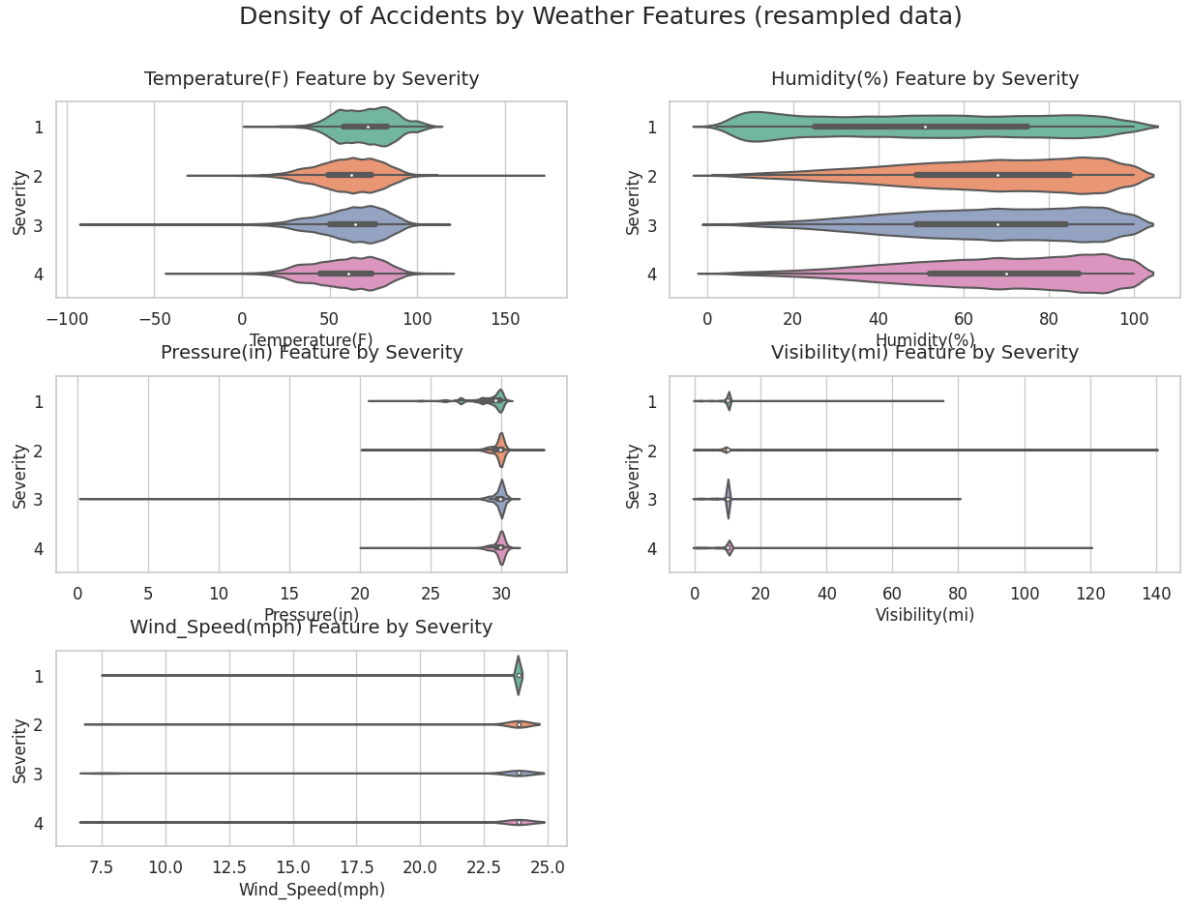
Ho estratto dal campo street le varie tipologie di strade creando una variabile booleana che descrivesse la tipologia di strada dell'incidente, e ho creato una matrice di correlazione tra il tipo di strada e la gravità dell'incidente.

Possiamo notare come le I- (ovvero le autostrade interstatali) sembrano essere le strade più pericolose, seguite dalle Highways (autostrade) e dalle Routes (autostrade federali), il motivo sembrerebbe essere la maggior velocità media che si percorre su queste strade rispetto alle strade normali.

Mappa degli incidenti

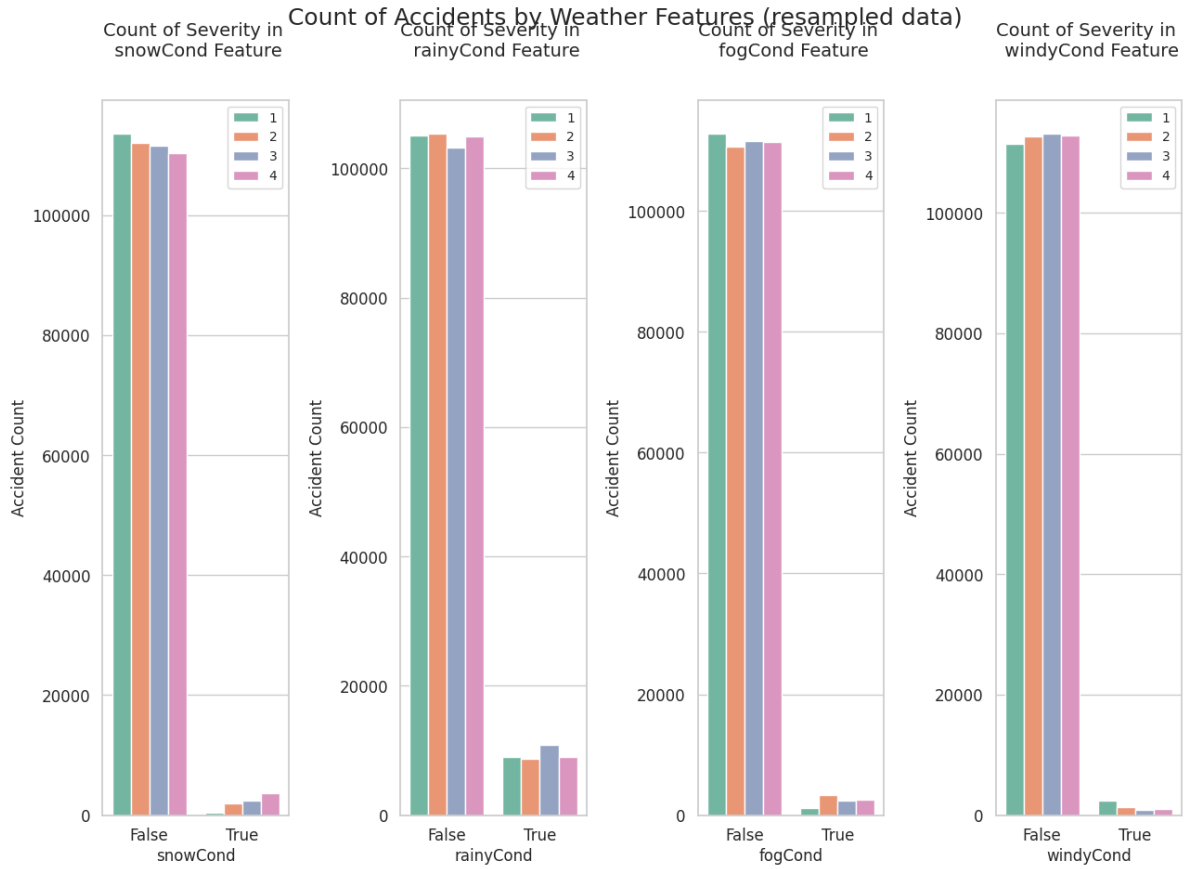


Relazione incidenti e condizioni atmosferiche



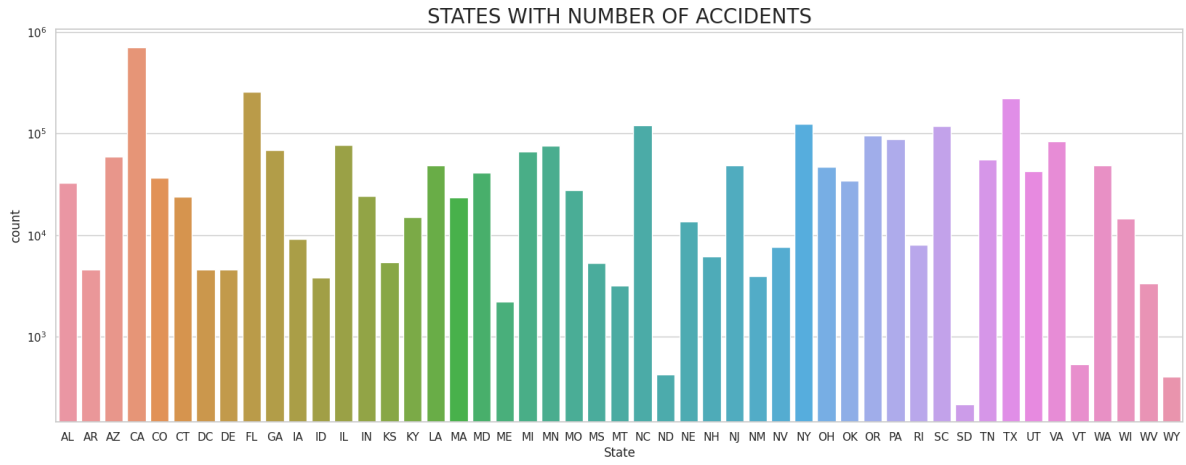
Possiamo notare come la distribuzione degli incidenti rispetto alla temperatura sia mediamente uguale ma, se un incidente avviene con temperature più basse, è più probabile che sia un incidente grave. Non si notano sostanziali differenze per quanto riguarda l'umidità, gli incidenti dal grado 2 in poi mediamente si verificano con percentuali di umidità più alte.

Relazione con le condizioni meteo

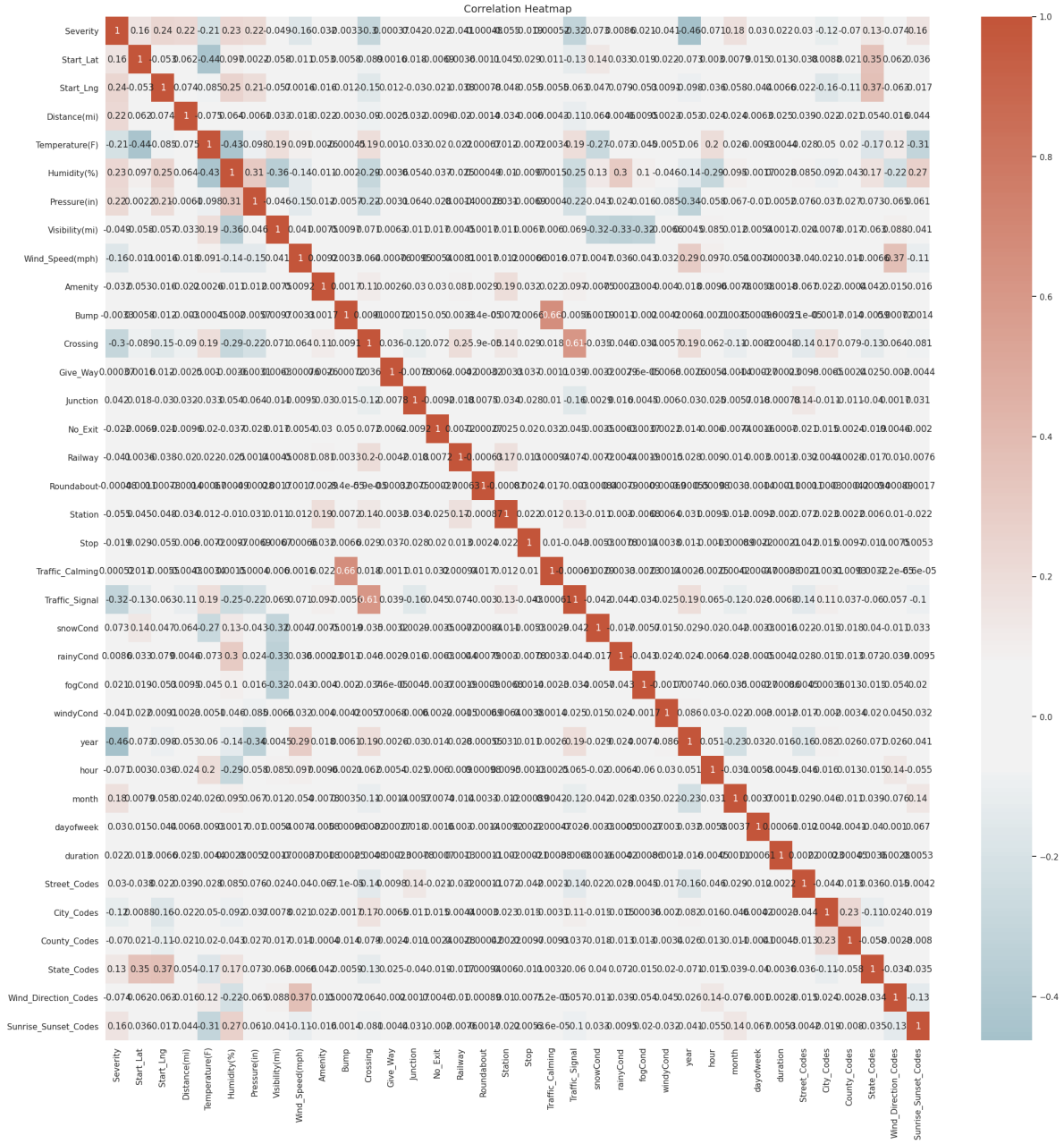


In questi grafici notiamo come in condizioni di neve è più probabile che se si verifichi un incidente questo sia di grado 4, in condizioni di pioggia è più probabile che sia di grado 3, in condizioni di nebbia di grado 2 ed in condizioni di vento di grado 1.

Numero di incidenti per stato



Matrice di correlazione



Classificazione

Come già detto l'obiettivo è quello di classificare la gravità di un eventuale incidente a partire dai fattori che l'hanno scaturito. Essendo l'obiettivo quello di poter analizzare a priori, a partire dalle informazioni della strada e atmosferiche, la gravità di un eventuale incidente, ho rimosso le informazioni presenti nel dataset riguardanti le conseguenze dell'incidente vero e proprio come la durata e la lunghezza della coda generata.

L'idea infatti è che l'algoritmo possa essere utilizzato dalle forze dell'ordine per prevenire incidenti stradali che abbiano gravi conseguenze sulla congestione del traffico.

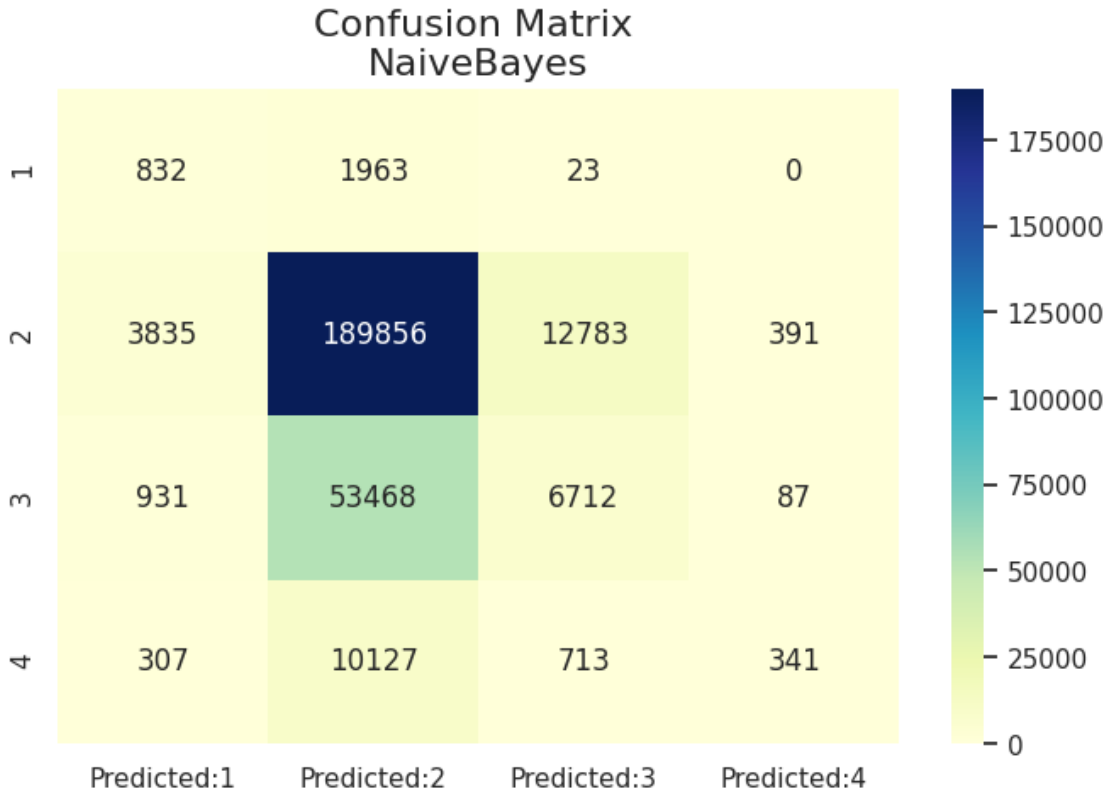
Per effettuare la parte di data mining ho preso in considerazione 5 algoritmi:

- NaiveBayes
- Random Forest
- Gradient Boost
- Knn
- Adaboost

Come primo test ho scelto di generare i dataset per il train ed i test utilizzando l'opzione stratify offerta dal metodo train_test_split che genera le variabili mantenendo le proporzioni tra le classi di gravità. Ho utilizzato ~850.000 dati per il train e ~300.000 per il test.

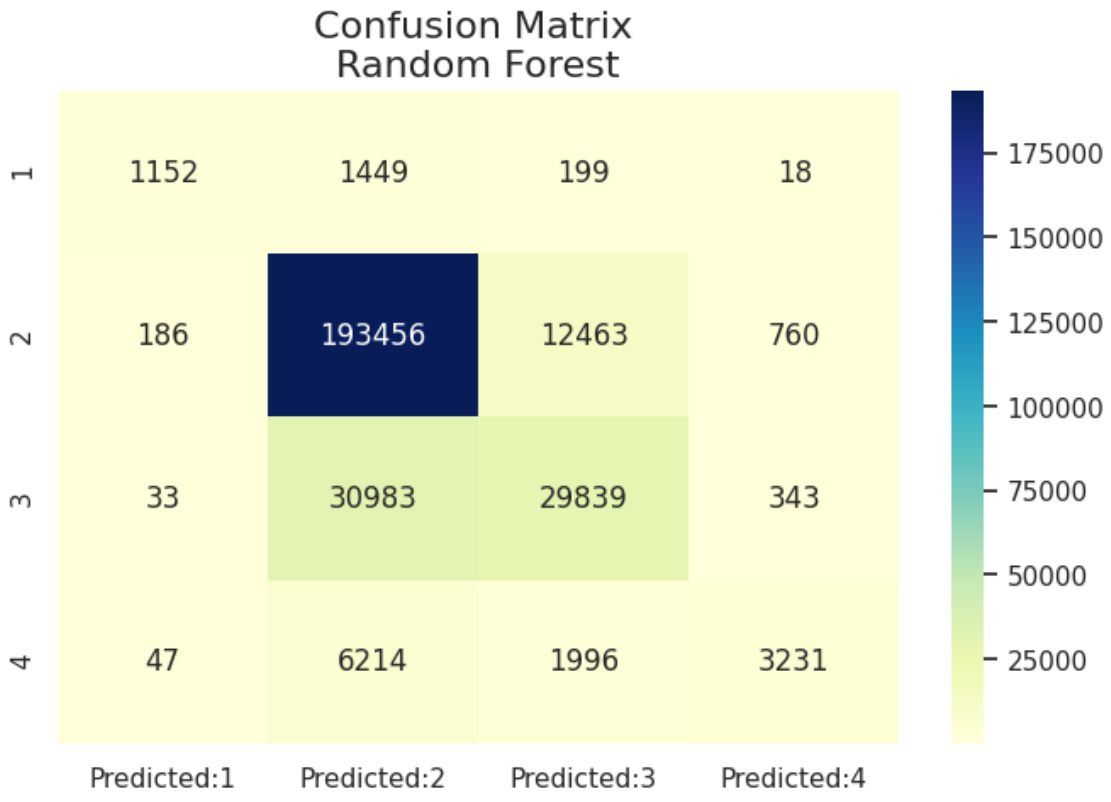
Di seguito i risultati dei train e dei test con i vari algoritmi:

- **Naive Bayes**: i risultati non sono soddisfacenti, l'algoritmo identifica bene gli incidenti di classe 2 (82% f1-score) ma è decisamente scarso per quanto riguarda gli altri gradi.



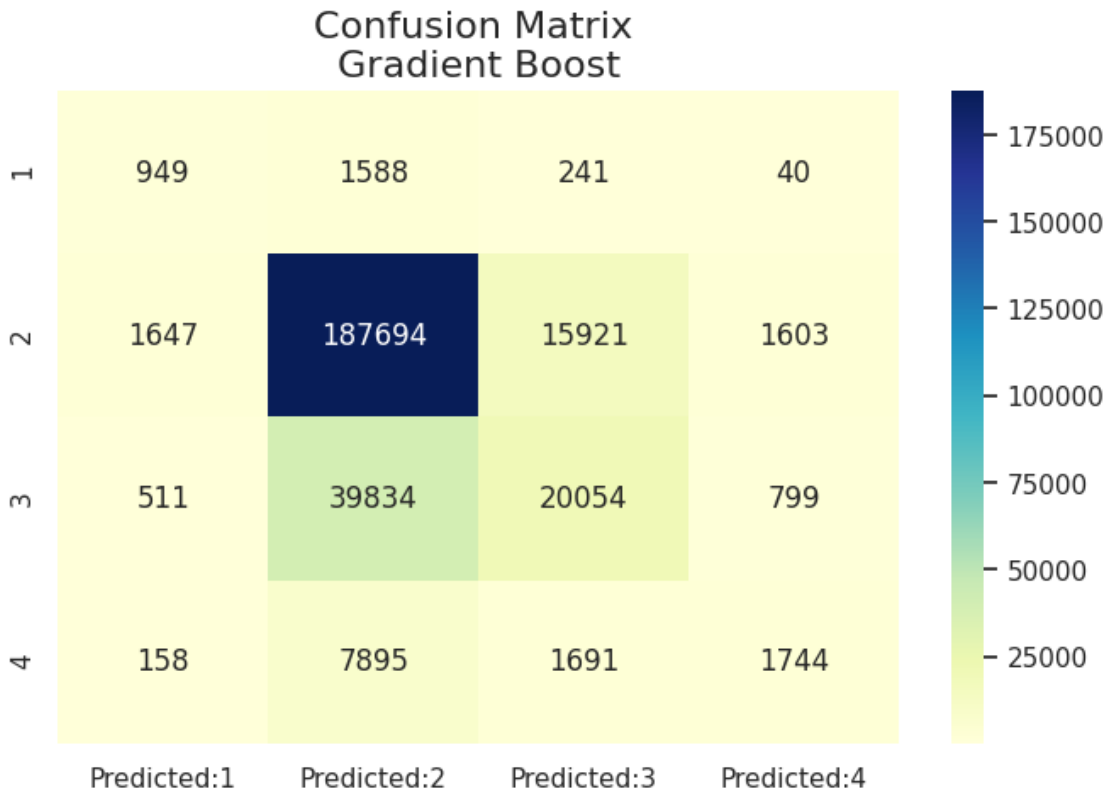
	precision	recall	f1-score	support
1	0.14	0.30	0.19	2818
2	0.74	0.92	0.82	206865
3	0.33	0.11	0.16	61198
4	0.42	0.03	0.06	11488
accuracy			0.70	282369
macro avg	0.41	0.34	0.31	282369
weighted avg	0.63	0.70	0.64	282369

- **Random Forest**: in questo caso l'algoritmo si è comportato leggermente meglio, ma anche qui è lampante la tendenza a classificare tutti gli incidenti come classe 2.



	precision	recall	f1-score	support
1	0.81	0.41	0.54	2818
2	0.83	0.94	0.88	206865
3	0.67	0.49	0.56	61198
4	0.74	0.28	0.41	11488
accuracy			0.81	282369
macro avg	0.76	0.53	0.60	282369
weighted avg	0.79	0.81	0.79	282369

- **Gradient Boost**: anche il gradient boost risulta solo accettabile solo per riconoscere gli incidenti di grado 2, con risultati peggiori rispetto al random forest per quanto riguarda gli altri gradi.

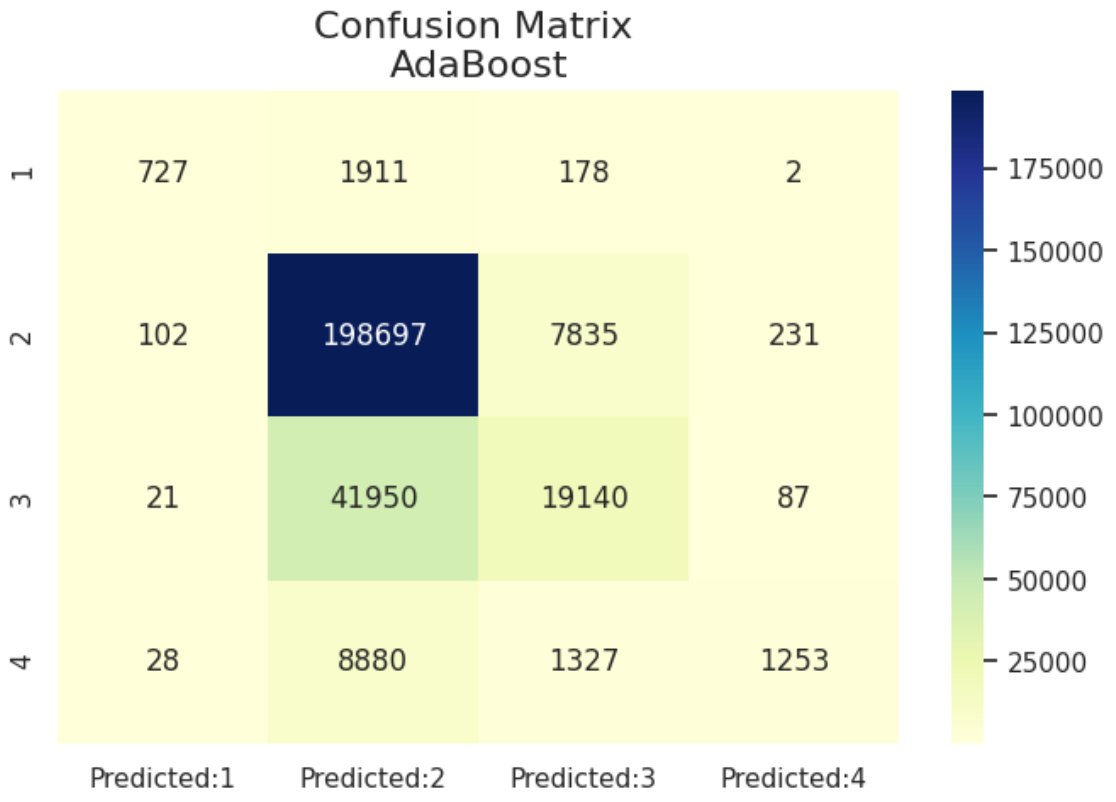


	precision	recall	f1-score	support
1	0.29	0.34	0.31	2818
2	0.79	0.91	0.85	206865
3	0.53	0.33	0.40	61198
4	0.42	0.15	0.22	11488
accuracy			0.75	282369
macro avg	0.51	0.43	0.45	282369
weighted avg	0.71	0.75	0.72	282369

- **KNN**: anche per knn non abbiamo ottenuto dei risultati accettabili.

	precision	recall	f1-score	support
1	0.22	0.09	0.13	282
2	0.78	0.86	0.82	20686
3	0.44	0.36	0.39	6120
4	0.17	0.06	0.09	1149
accuracy			0.71	28237
macro avg	0.40	0.34	0.36	28237
weighted avg	0.68	0.71	0.69	28237

- **Ada Boost**: Anche in questo caso notiamo che i risultati sono sbilanciati verso il grado 2.



	precision	recall	f1-score	support
1	0.83	0.26	0.39	2818
2	0.79	0.96	0.87	206865
3	0.67	0.31	0.43	61198
4	0.80	0.11	0.19	11488
accuracy			0.78	282369
macro avg	0.77	0.41	0.47	282369
weighted avg	0.77	0.78	0.74	282369

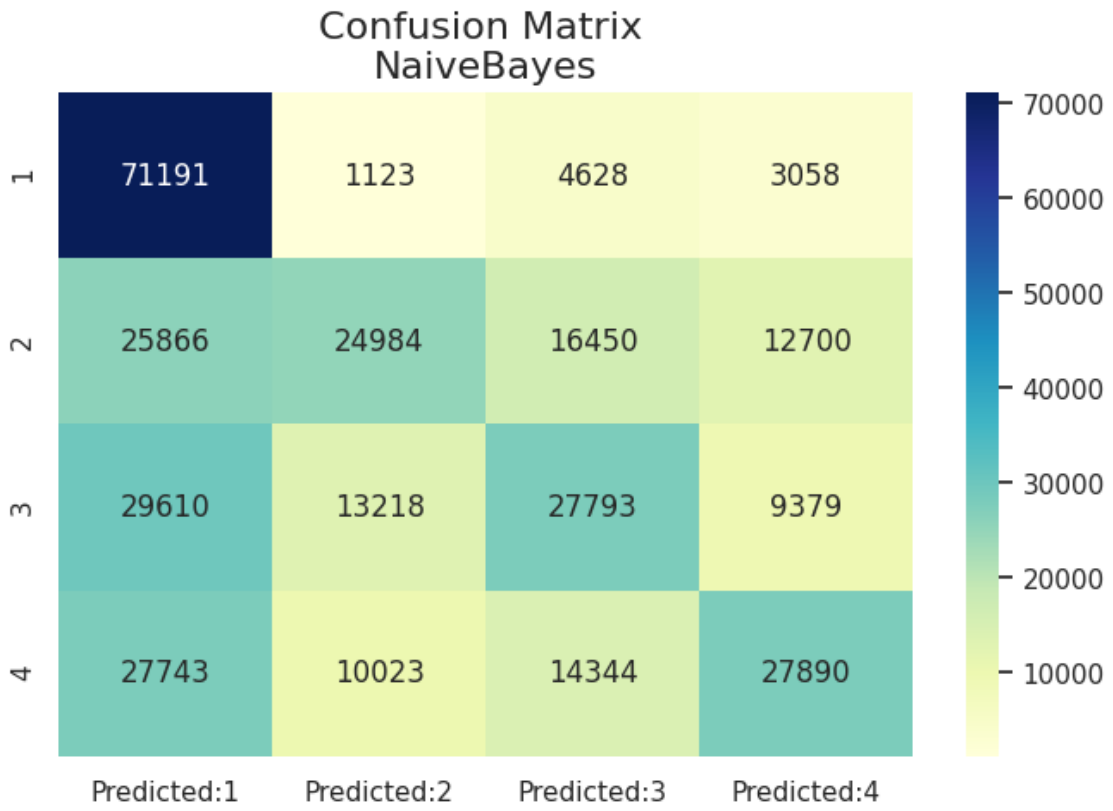
Analisi: tutti gli algoritmi di classificazione non hanno dato risultati accettabili, alcuni identificando molto bene gli incidenti di gravità 2. In generale abbiamo notato uno squilibrio verso questa categoria, probabilmente dovuto al fatto che il dataset è molto squilibrato verso questa categoria.

Il prossimo step è stato quello di effettuare un resampling e quindi allineare il numero di casi, in questo modo i dataset di train e test conterranno un numero uguale di incidenti per ogni categoria, al fine di eliminare lo squilibrio verso la classificazione a grado 2. Nello specifico ho utilizzato 400.000 casi suddivisi equamente tra le 4 categorie.

Al fine di effettuare un tuning più preciso dei parametri degli algoritmi ho utilizzato il metodo gridSearch che permette di effettuare i train degli algoritmi con i vari parametri e di selezionare l'esecuzione che fornisce i risultati migliori.

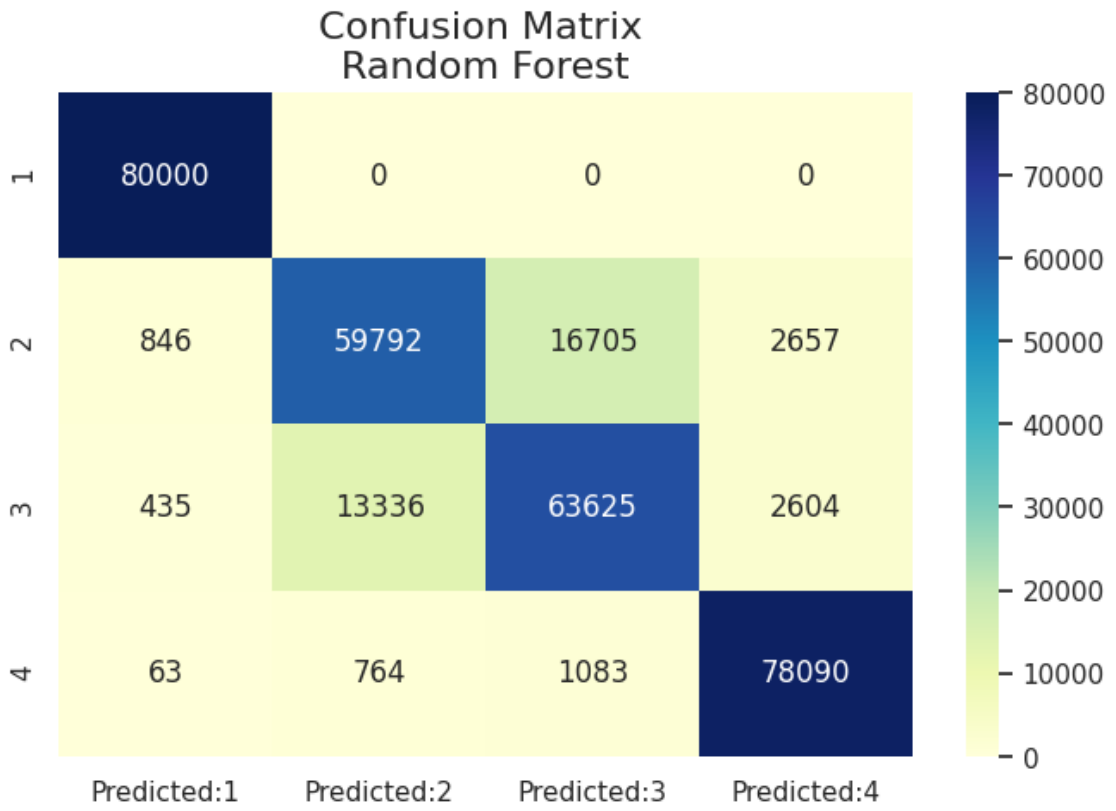
Di seguito i risultati:

- **Naive Bayes**: I risultati ottenuti sono più incoraggianti rispetto ai precenti, otteniamo infatti un f1-score complessivamente migliore rispetto al run precedente, restano risultati non ancora utilizzabili però.



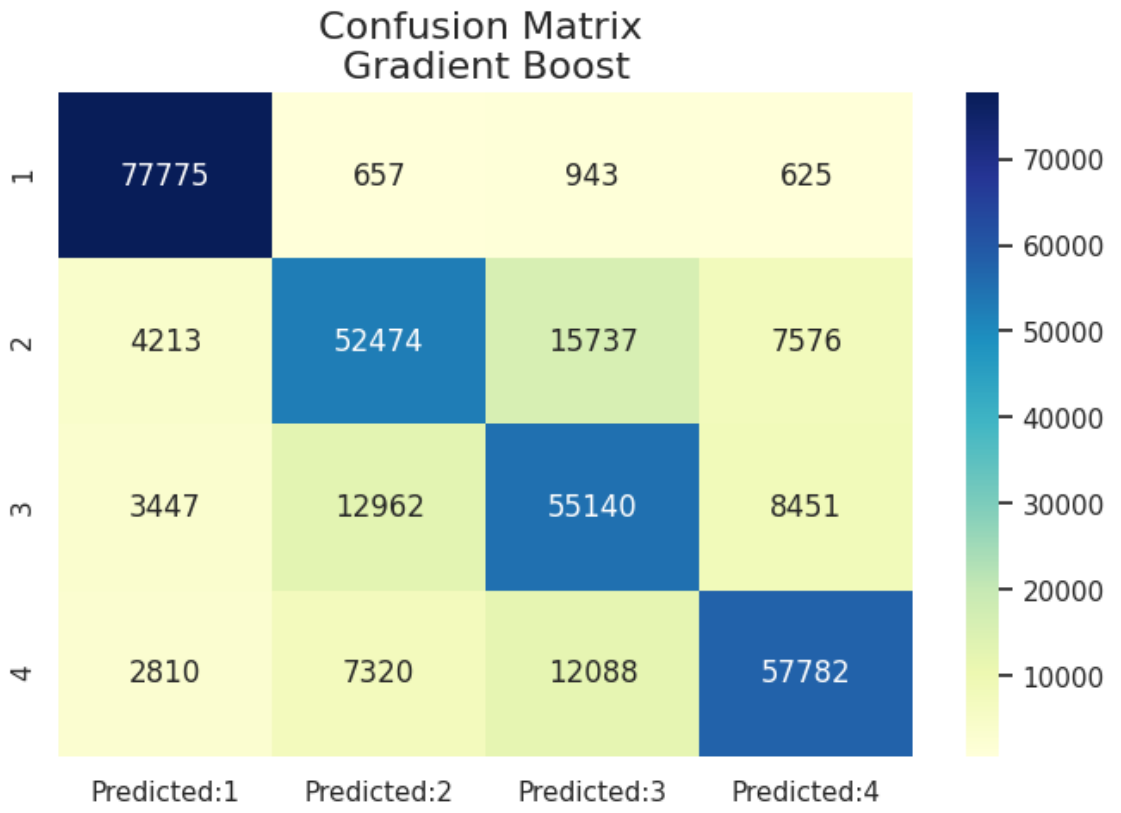
	precision	recall	f1-score	support
1	0.46	0.89	0.61	80000
2	0.51	0.31	0.39	80000
3	0.44	0.35	0.39	80000
4	0.53	0.35	0.42	80000
accuracy			0.47	320000
macro avg	0.48	0.47	0.45	320000
weighted avg	0.48	0.47	0.45	320000

- **Random Forest**: Il random forest migliora notevolmente fornendo dei buoni risultati, nello specifico categorizza molto bene gli incidenti di grado 1 e soprattutto quelli di grado 4 (che sono i dati più importanti):



	precision	recall	f1-score	support
1	0.98	1.00	0.99	80000
2	0.81	0.75	0.78	80000
3	0.78	0.80	0.79	80000
4	0.94	0.98	0.96	80000
accuracy			0.88	320000
macro avg	0.88	0.88	0.88	320000
weighted avg	0.88	0.88	0.88	320000

- **Gradient Boost**: i risultati nel gradient boost sono peggiori rispetto a quelli del random forest, ma poiché ottenuti da un numero di dati minori rispetto a quelli utilizzati per il train del random forest (a causa dei lunghi tempi di esecuzione)



	precision	recall	f1-score	support
1	0.88	0.97	0.92	80000
2	0.71	0.66	0.68	80000
3	0.66	0.69	0.67	80000
4	0.78	0.72	0.75	80000
accuracy			0.76	320000
macro avg	0.76	0.76	0.76	320000
weighted avg	0.76	0.76	0.76	320000

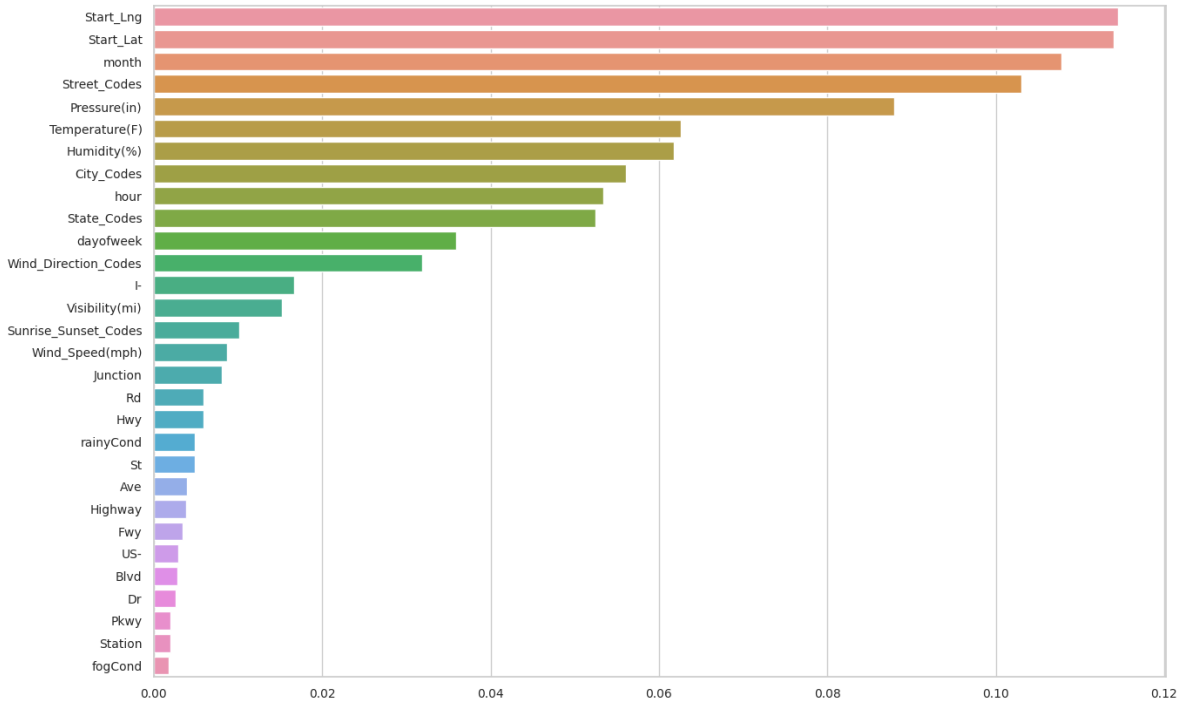
- **KNN:** fornisce risultati migliori rispetto al run precedente ma non buoni quanto il random forest ed il gradient boost.

	precision	recall	f1-score	support
1	0.66	0.80	0.72	11400
2	0.48	0.40	0.44	11400
3	0.50	0.53	0.52	11400
4	0.55	0.49	0.52	11400
accuracy			0.56	45600
macro avg	0.55	0.56	0.55	45600
weighted avg	0.55	0.56	0.55	45600

Analisi: eliminando lo squilibrio del dataset i risultati sono complessivamente migliori, con gli algoritmi di classificazione composti da alberi decisionali che forniscono i risultati migliori.

Conclusione

Possiamo quindi dire che l'algoritmo Random Forest ha fornito i migliori risultati.
Di seguito il peso delle feature per il random forest:



Notiamo come i dati coincidano con le nostre analisi precedenti, l'algoritmo sui seguenti dati:

- Posizione
- Mese
- Numero della strada
- Pressione
- Temperatura
- Ora e giorno della settimana

Abbiamo perciò raggiunto l'obiettivo di classificare correttamente la gravità di un incidente a partire dalle condizioni atmosferiche e analizzando le strade su cui si possono verificare. Lo scopo dell'algoritmo è quello di poter aiutare le forze dell'ordine a mantenere più sicure le strade e minimizzare le code generate dagli incidenti. L'utilizzo potrebbe essere quello di eseguire l'algoritmo a inizio settimana, utilizzando le previsioni atmosferiche della settimana ed analizzando una serie di strade (magari le più importanti a livello di traffico) per capire su quali si potrebbero verificare incidenti gravi, permettendo così di aumentare i controlli o modificare i limiti di velocità.

References

https://seaborn.pydata.org/index.html

https://vita.had.co.nz/papers/letter-value-plot.html

https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

https://scikit-learn.org/stable/index.html