New York city taxi trip duration

De Rosso Daniel

Sommario

In questo report si presentano il lavoro di analisi e i risultati ottenuti da esso su dati riguardanti i tragitti dei taxi, effettuati a New York, registrati nell'anno 2016. Gli obiettivi perseguiti sono stati quelli di prevedere la durata di un tragitto avendo a disposizione dati come: coordinate spaziali del punto di origine e destinazione, timestamp di inizio viaggio e numero di passeggeri. Per rispondere ai quesiti presentati si è effettuata in principio un'analisi esplorativa del dataset, volta a familiarizzare con i dati in possesso e all'applicazione delle dovute operazioni per renderli adeguati alle successive analisi. In secondo luogo si è andato ad identificare il modello adatto al caso per effettuare le stime, andando a confrontare random forest e reti neurali di diverse architetture. Il risultato finale, ottenuto con metrica RMSLE, è stato un'errore di 0.47498.

Link GitHub codice

Indice

1. Introduzione

I yellow cab di New York sono ormai diventati un simbolo della grande mela. Raffigurati spesso nei film come il mezzo di trasporto di tutti i giorni, garantiscono agli abitanti di New York una solouzione comoda, ma soprattuto veloce, per muoversi in città. Sebbene negli ultimi anni i taxi facciano fatica a contrastare alternative "app-based", come la nota piattaforma Uber, le statistiche riportano più di 13,000 taxi operanti nella città di New York.[1]

1.1 Dati

Il dataset in esame[2] è una raccolta di tutti i tragitti di taxi, avvenuti nel 2016, nella città di New York. Inizialmente i record sono stati raccolti e pubblicati da "NYC Taxi and Limousine Commission (TLC)", per poi essere presentati come competizione su kaggle.com. Il dataset viene già suddiviso in due parti per facilitare le fasi di training e test. In particolare i due dataset contengono rispettivamente 1,458,644 record di train e 625,134 di test. Gli attributi per singolo viaggio sono i seguenti:

- **id** identificativo di ogni viaggio
- **vendor_id** identificativo univoco per ogni provider
- **pickup_datetime** data e ora dell'inizio del viaggio
- **dropoff_datetime** data e ora della fine del viaggio
- **passenger_count** numero di passeggeri
- **pickup_longitude** longitudine di inizio viaggio
- **pickup_latitude** latitudine di inizio viaggio
- **dropoff_longitude** longitudine di fine viaggio
- **dropoff_latitude** latitudine di fine viaggio
- **store_and_fwd_flag** flag che specifica se i dati del viaggio sono stati caricati in ritardo per mancanza di connessione
- **trip_duration** durata del viaggio in secondi

1.2 Domanda di ricerca

Il problema viene posto con l'obiettivo di predire, utilizzando i dati a disposizione, la stima della durata del viaggio. La valutazione viene eseguita con la funzione Root Mean Squared Logaritmic Error come metrica.

$$\varepsilon = \sqrt{1/n \sum (\log(p_i + 1) - \log(\alpha_i + 1))^2} \quad (1)$$

Per la natura stessa del problema si dovrà quindi utilizzare un modello regressivo.

2. Dataset e Preprocessing

Nella fase iniziale si è svolta un'esplorazione dei dati a disposizione per familiarizzare con essi e studiarne le distribuzioni. Successivamente si cercherà di estrarre features supplementari e rimuovere outliers che potrebbero influire negativamente sui modelli nelle fasi successive.

2.1 Esplorazione dati

Come anticipato, il dataset originario è già suddiviso in test e in train per facilitare la valutazione su kaggle.com. Assumento quindi una distribuzione uniforme in entrambi i subset, le analisi seguenti faranno riferimento al solo dataset di training.

In primo luogo non si riscontrano valori mancanti in nessuna colonna. Con una semplice funzione riassuntiva delle variabili quantitative, è già possibile osservare (figura 1) la presenza di valori errati nel

numero di passeggeri (min passenger_count=0) e nella durata del viaggio (min trip_duration=1, max trip_duration=3.5^6)

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration
count	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06
mean	1.534950e+00	1.664530e+00	-7.397349e+01	4.075092e+01	-7.397342e+01	4.075180e+01	9.594923e+02
std	4.987772e-01	1.314242e+00	7.090186e-02	3.288119e-02	7.064327e-02	3.589056e-02	5.237432e+03
min	1.000000e+00	0.000000e+00	-1.219333e+02	3.435970e+01	-1.219333e+02	3.218114e+01	1.000000e+00
25%	1.000000e+00	1.000000e+00	-7.399187e+01	4.073735e+01	-7.399133e+01	4.073588e+01	3.970000e+02
50%	2.000000e+00	1.000000e+00	-7.398174e+01	4.075410e+01	-7.397975e+01	4.075452e+01	6.620000e+02
75%	2.000000e+00	2.000000e+00	-7.396733e+01	4.076836e+01	-7.396301e+01	4.076981e+01	1.075000e+03
max	2.000000e+00	9.000000e+00	-6.133553e+01	5.188108e+01	-6.133553e+01	4.392103e+01	3.526282e+06

Figura 1. Statistiche descrittive delle variabili quantitative

Effettuando quindi un'analisi più precisa sulla colonna trip_duration, come mostrato in figura 2, si può osservare come la distribuzione dei valori presenti diversi outliers. Vista la durata del viaggio, in alcuni casi superiore alle 10 ore, questi outliers rappresentato sicuramente degli errori nelle entry dei record.

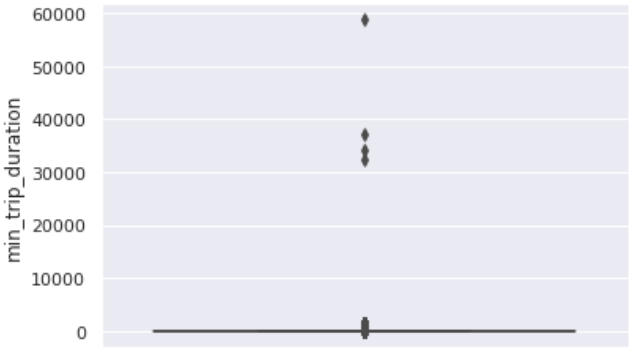

Figura 2. Box plot della durata del viaggio in minuti

Essendo il dataset una raccolta di dati dei taxi di New York, è importante controllare anche il dominio delle coordinate spaziali di questi per evidenziare punti anomali. Come si può intuire dalla figura 3, si presentano una serie di valori irregolari, al di fuori dei dati ammessi nella descrizione del dataset.

Si procederà quindi alla rimozione di essi nelle fasi successive.

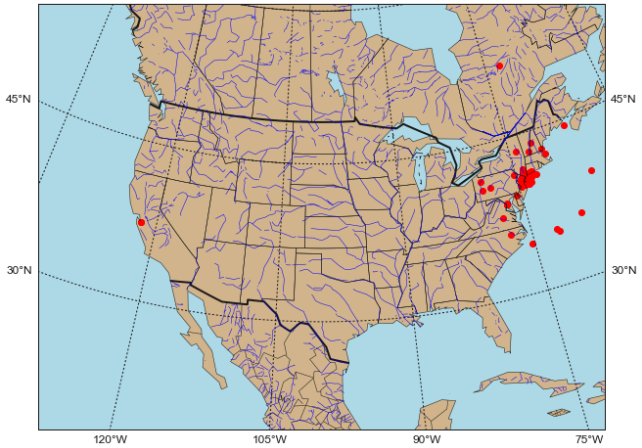

Figura 3. Coordinate dei pickup points nel territorio nord americano

Per concludere l'analisi esplorativa si è voluto studiare i dati temporali, per ricercare variabili che potessero spiegare varianze significative rispetto alla durata del viaggio. Come facilmente intuibile, la maggior parte dei viaggi viene svolta nelle ore diurne, con un calo significativo nella fascia oraria compresa tra le 1 e le 6 del mattino.

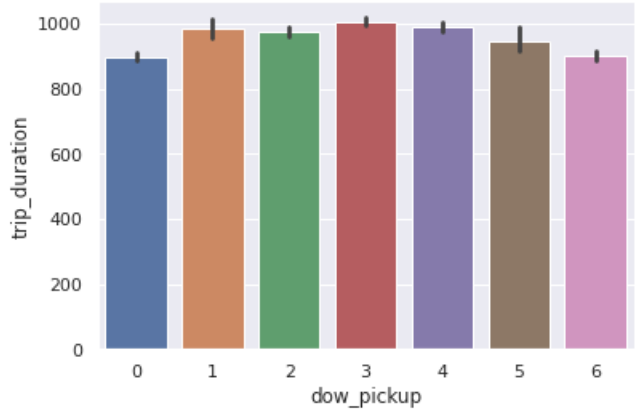

Figura 4. Durata media del viaggio rispetto al giorno della settimana

La figura 4 mostra come il giorno della settimana in cui si effettua il viaggio, corrisponda in media ad una durata minore del tragitto nel fine settimana e nel lunedì.

2.2 Feature engineering

2.2.1 Variabili temporali

Essendo il valore delle variabili temporali di tipo timestamp, si è proceduto con un estrazione dei singoli dati per permettere una maggiore interpretabilità al modello.

Il primo dato interessante che si è voluto estrarre è quello del giorno della settimana, si rappresentano quindi i giorni dal lunedì alla domenica con valori numerici 0-6 nella colonna **dow_pickup**.

Successivamente la data completa viene suddivisa in mese, giorno ed ora (colonne **month_pickup**, **day_pickup** e **hour_pickup**).

2.2.2 Distanza

Utilizzando le coordinate spaziali di pickup e dropoff, viene calcolata la colonna **dist_km**, che rappresenta la distanza in chilometri dal punto di inizio e fine del viaggio. Per questo calcolo si è utilizzata la funzione haversine[3], la quale calcola la distanza di due punti sulla superficie di una sfera date le coordinate longitudinali e latitudinali di essi.

2.2.3 Velocità media

Utile per evidenziare la presenza di outliers, il dato sulla velocità media in km/h del viaggio è stato ottenuto combinando il valore della distanza, precedentemente estrato, e la durata del viaggio. La colonna **average_speed** è stata quindi aggiunta al dataset.

2.3 Data integration

Utilizzando la libreria GeoPandas[4], derivata da pandas e ottimizzata per la gestione di dati geospaziali, è stato possibile integrare il dataset originale con il dataset dei quartieri della città di New York. Nel dettaglio, dopo aver caricato lo shapefile[5] di New York, contenete i poligoni dei rispettivi quartieri in coordinate spaziali, si è realizzata una join incrociata con il dataset originale, per arricchiere ogni record con il relativo quartiere (se il punto rientra effettivamente in un poligono della lista precedentemente caricata). Successivamente questa operazione si sono aggiunte le seguenti colonne al dataset: **BoroCode, BoroName, Geometry**.

Questa punto sarà utile anche nella fase di pulizia degli outliers per discriminare i record esterni al territorio di New York.

2.4 Outliers

Utilizzando i dati estratti e arricchiti, e tenendo in considerazione le osservazioni fatte in fase di esplorazione del dataset, si è applicato un filtro per rimuovere tutti i dati con queste caratteristiche:

- durate del viaggio superiore alle 3 ore. La legge americana non permette, per motivi lavorativi, turni di viaggio superiori alle 13 ore. Inoltre la distribuzione dei dati sulle durate dei viaggi riportava una frequenza inferiore allo 0.1% per tragitti durati più di 3 ore, per questo motivo si è proceduto con la semplice eliminazione.

- numero di passeggeri uguale a 0 o superiore a 6. Essendo i taxi predisposti maggiormente per 4 passeggeri, con alcune eccezioni che permettono di arrivare fino a 6, si è proceduto con l'eliminazione dei viaggi che non rispettassero queste caratteristiche.

- coordinate spaziali dei pickup points al di fuori di New York.

- distanza del viaggio uguale a 0.

- velocità media del viaggio superiore ai 100km/h. Entro i limiti cittadini, la velocità massima consentita dalla legge americana è di 72km/h,

con autostrade che arrivano a massimo 105km/h nei pressi di New York.

Successivamente alla rimozione di tutti questi records, la variazione della grandezza del dataset originale è dello 0.6%. Si ritiene per questo motivo che la distribuzione dei dati iniziali non sia stata influenzata significativamente.

2.5 Data summary

Dopo aver pulito i valori anomali e dopo aver arricchito il dataset con colonne aggiuntive, si riportano alcune visualizzazioni per riassumere al meglio i dati.

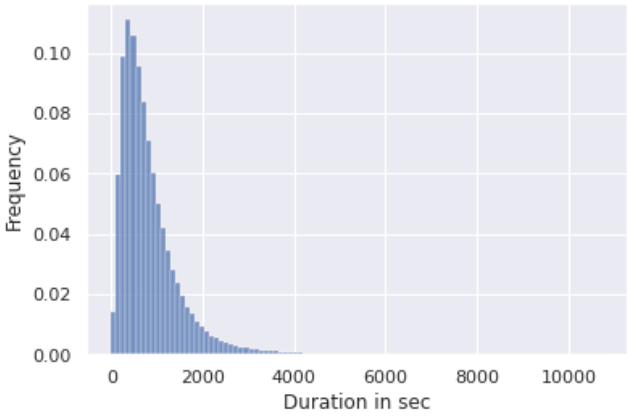

Figura 5. Distribuzione di frequenza dei viaggi in base alla durata

La media della durata del viaggio è di 838 secondi (circa 14 minuti), con il terzo quartile che cade nei 1074 secondi (circa 18 minuti).

La figura 6 mostra i pickup points all'interno di New York dopo la rimozione di tutti i valori anomali.

Infine, come si può evincere dalla figura 7, i viaggi che partono dal quartiere del Queens, sono in media più lunghi di tutti gli altri. Questa varianza si può spiegare dalla presenza dell'aeroporto in questo quartiere, il quale richiede viaggi in media più lunghi per raggiungere quartieri più centrali come Manhattan e Brooklyn.

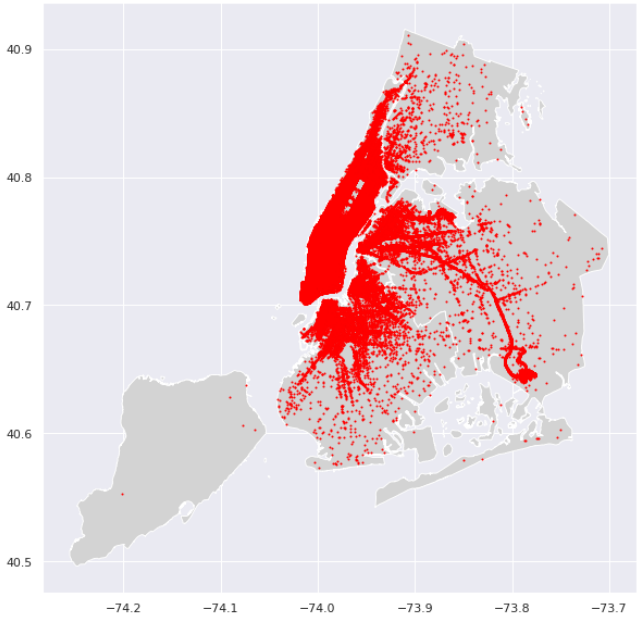

Figura 6. Pickup points in New York

2.6 Feature selection

Utilizzando parsimonia e senso critico si è deciso di rimuovere gli attributi superfli e ridondanti prima di passare alla fase di allenamento del modello. Questo per sia per migliorare le performance sai per rendere il tutto più interpretabile. Dopo un'attenta analisi quindi, le colonne rimenenti ritenute valide per la fase successiva sono le seguenti:

- **passenger_count**, conteggio dei passeggeri
- **trip_duration**, durata del viaggio in secondi. Variabile dipenente oggetto della previsione.
- **month_pickup**, mese in cui si è effettuato il viaggio
- **hour_pickup**, ora del giorno in cui si è effettuato il viaggio
- **dow_pickup**, giorno della settimana in cui si è effettuato il viaggio
- **BoroCode**, codice identificativo del quartiere di New York relativo alla partenza del viaggio (0-4)

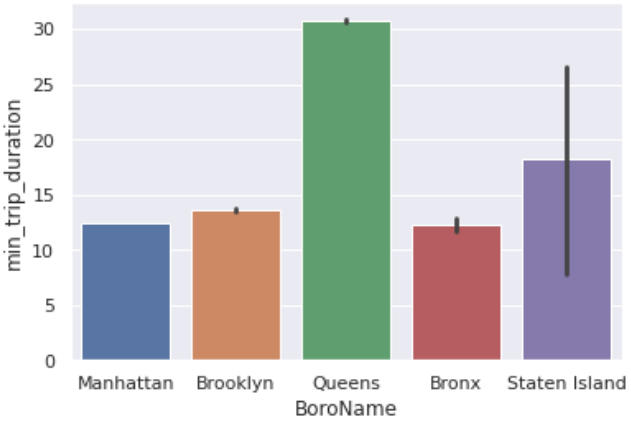

Figura 7. Durata media del viaggio in minuti rispetto al quartiere di partenza

- **dist_km**, distanza in chilometri del tragitto calcolato con funzione haversine

3. Note metodologiche

In questa sezione si produrranno in sequenza dei modelli, con l'obiettivo di migliorarli e ottimizzarli, per ottenere il modello con le migliori performance da utilizzare in fase finale di testing.

3.1 Random forest

Si è deciso di utilizzare una random forest standard per sfruttare le performance restituite come benchmark per le fasi successive. Inoltre le random forest permettono di estrarre l'importanza delle features che sono state usate, per predirre il valore atteso. Per allenare la random forest con i migliori hyperparameters è stata utilizzata una cross-validation, facendo variare il parametro della profondità, individuando così la miglior configurazione con max_depth=10. Tale modello ha restituito in fase di validazione un valore MSLE=0.1891.

3.1.1 Features importance

Come già anticipato, per spiegare al meglio il modello si è ritenuto utile utilizzare una funzione di feature importance che mostrasse quanto ogni singola variabile abbia influito sulla predizione della regressione.

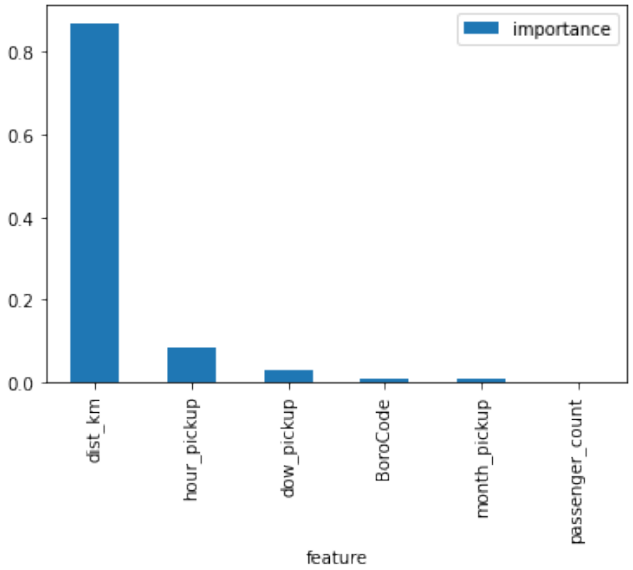

Figura 8. Features importance nella previsione della random forest

Come si può capire dalla figura 8, e come si poteva intuire a priori, la distanza del viaggio è significativamente più influente di tutte le altre per il calcolo della variabile dipendente. Si cercherà quindi nelle fasi successive di migliorare la partecipazione, per quanto possibile, anche degli altri attributi.

3.2 Neural network

Si passa ora ad un utilizzo di reti neurali feed forward, con l'obiettivo di migliorare le performance precedenti senza incorrere in overfitting. Come punto di partenza si è deciso di utilizzare una rete neurale con 4 hidden layer, un valore di 256 come spazio di output per ogni livello e ReLu come funzione di attivazione. Al layer di output invece è stata applicata una funzione lineare per restituire il risultato della regressione.

Il risultato di tale modello, con 200 epoche e batch size di 1024, è stato di un valore MSLE=0.2200

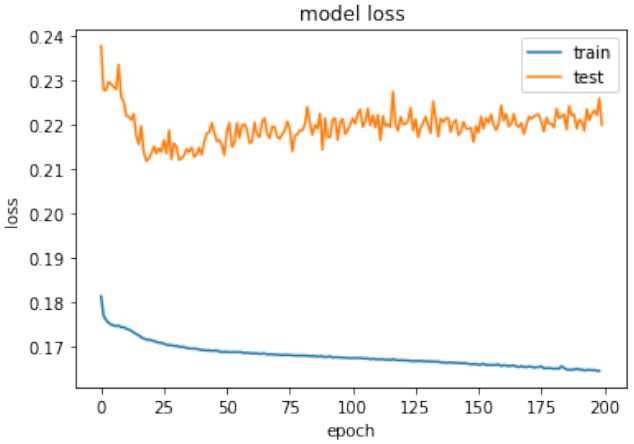

Figura 9. MSLE train e validation con rete neurale base

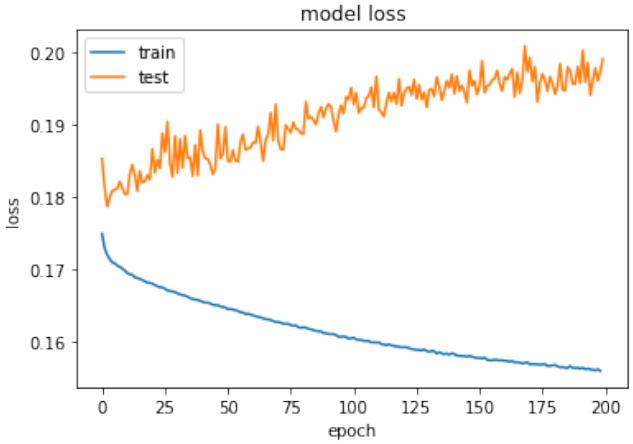

Figura 10. MSLE train e validation applicando one hot encoding

per il validation set. Misurazione che non fa preferire questo modello alla random forest. Come si può osservare anche dalla figura 9, al passare delle epoche, il modello entra in uno stato di overfitting. Infatti, sebbene le performance di training migliorino leggermente, quelle del validation set peggiorano con l'esecuzione, fino a stabilizzarsi su un valore dei MSLE di circa 0.22.

3.2.1 One hot encoding

Finora le variabili derivate da timestamp (**month pickup, hour pickup, dow pickup**), e la colonna relativa al quartiere (**BoroCode**), sono state considerate come variabili quantitative. Sebbene siano variabili numeriche però, rientrano nelle varibili categoriche. Per questo motivo in questa sezione si è voluto applicare una funzione one hot encoding, assegnando quindi un flag booleano ad ogni possibile valore che esse possono avere. La rete neurale che viene allenata in questo punto mantiene l'architettura precedente, a meno dei nodi in ingresso, che aumentano in proporzione al numero della cardinalità dei valori univoci che hanno le colonne sottoposte alla funzione one hot encoding.

Dalla figura 10 si può osservare come per alcune epoche, si ottenga un valore di MSLE inferiore allo 0.18, prima di andare in overfitting. Ad ogni modo, il valore MSLE ottenuto alla fine della fase di training è di 0.1990. Ciò significa che il modello in fasi successive necessiterà ancora la risoluzione del problema di overfitting, ma le performance sono tuttavia migliorate.

3.2.2 Circular variables

Un'alternativa per trattare dati ciclici è quella di utilizzare trasformazioni trigonometriche[6]. Applicare queste funzioni permette di risolvere la discontinuità con questo tipo di variabili, che per loro natura dovrebbero essere circolari. Vengono quindi applicate le seguenti funzioni

$$\sin \pi * x/max(x) \quad \cos \pi * x/max(x) \qquad (2)$$

alle colonne: **hour_pickup, month_pickup, dow_pickup**. Gli attributi precedenti vengono quindi divisi in 2 colonne ciascuno.

Per quanto riguarda la colonna **BoroCode**, in questo caso viene comunque trasformata con la funzione one hot encoding.

Come si può osservare dalla figura 11, le prestazioni di quest'ultimo modello non migliorano rispetto al precedente. Per la prossima ed ultima

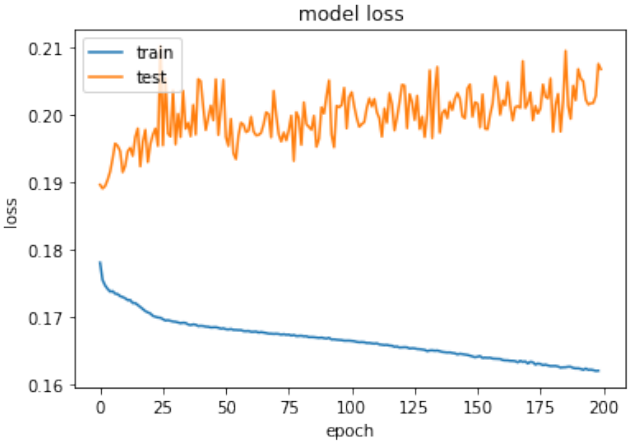

Figura 11. MSLE train e validation applicando circular values

analisi si è scelto quindi di procedere con il modello trasformato con sole funzioni one hot encoding del paragrafo 3.2.1.

3.2.3 Regularization, dropout e early stopping

Come ultimo passo per migliorare l'outcome si cerca di risolvere i problemi di overfitting precedentemente riportati. In particolare si applicheranno le tecniche di regolarizzazione, dropout ed earlystopping. Per quanto riguarda la regolarizzazione, essa verrà applicata ad ogni layer della rete neurale con una funzione L2 e factor=0.0001. Ad ogni layer viene inoltre applicata una funzione di dropout per eliminare casualmente nodi dall'architettura e ridurre l'overfitting, in particolare il dropout rate è stato impostato ad un valore di 0.2. Come ultima configurazione si arresta l'allenamento della rete neurale dopo l'esecuzione di 50 epoche che non hanno portato miglioramento alla loss, salvando lo stato del modello che ha riportato risultati migliori in termini di MSLE durante tutta la fase di training. A differenza degli allenamenti precedenti, considerando che il training viene fermato in automatico dall'early stopping, si è voluto aumentare il numero di epoche di alenamento a 1000 come tetto massimo.

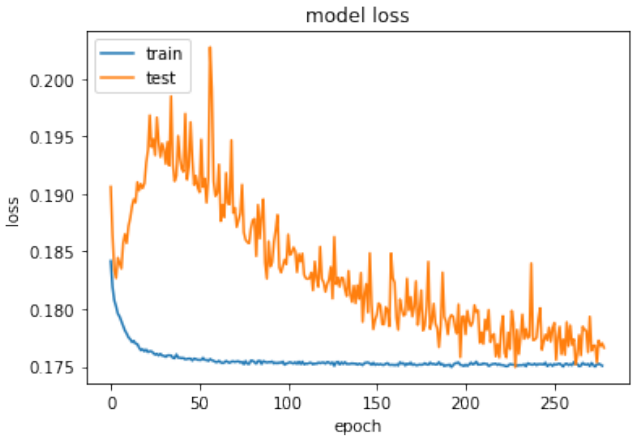

Figura 12. MSLE train e validation risolvendo overfitting

Come si puo vedere in figura 12, il modello migliora significativamente la metrica di loss. A differenza dei casi precedenti infatti, con l'aggiunta delle tecniche di regolarizzazione, il modello trova la configurazione ideale dei parametri più lentamente e senza entrare in uno stato di overfitting. Come risultato finale si ha che, il modello salvato in epoca 229, riporta un valore di MSLE=0.17494.

3.2.4 Model explanation con lime

Per concludere la sezione di training, si è voluto utilizzare la libreria LIME (Local Interpretable Model-Agnostic Explanations)[7] per aumentare l'interpretabilità del modello. LIME è indipendente dal modello, ciò significa che può essere applicato ad ad ogni tecnica di machine learning. Questo viene fatto modificando un singolo campione, ritoccando le features e osservando come ciò impatta il risultato in output. Per questo motivo LIME mostra in risposta una lista di valori, che indicano quanto ogni feature abbia influenzato la predizione. Viene definito local interpretable perchè dipende dal sample che riceve, per questo motivo può essere instabile.

La figura 13 mostra il risultato di LIME, delle 7 feature più influenti, applicato ad un random

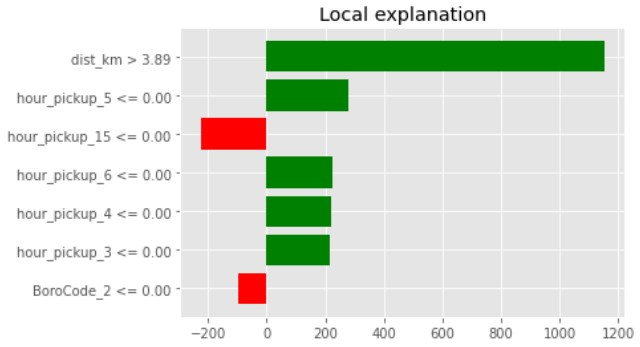

Figura 13. Output LIME con random sampling (predicted value=3294)

sampling. Come osservato anche in precedenza, la feature di maggior influenza rimane la distanza del tragitto. Aumentano però di importanza anche gli attributi relativi all'orario di inizio tragitto.

4. Risultati e discussioni

Il modello ottenuto eseguendo one hot encoding e regolarizzazione viene salvato con pickle per mantenerne lo stato. Viene poi operata la previsione della variabile **trip_duration**, anche per il dataset di test. Una volta fatto ciò, l'output è stato formattato secondo le specifiche di kaggle e caricato sul sito per ricevere una valutazione. Utilizzando quindi come metrica la misura RMSLE, presentata nella sezione 1.2, viene calcolato automaticamente uno score finale di 0.47498. Risultato che differisce leggermente con i valori ottenuti in fase di training, e ciò può essere spiegato in parte dalla presenza di outilers nel dataset di test (che per il calcolo del modello sono invece stati rimossi) e in parte dalla metrica di valutazione RMSLE che tende a penalizzare gli errori bassi.

Ulteriori miglioramenti che potrebbero essere apportati a questo modello riguardano: un maggior arricchimento dei dati (time series su condizioni traffico e condizioni meteo) e un calcolo più efficace della distanza percorsa. In questo problema infatti viene considerata la distanza tra il punto di destinazione e di origine in linea d'aria, sarebbe però più utile poter conoscere il percorso stradale più breve che congiunge i due punti.

5. Conclusione

Attraverso questo elaborato è stato possibile mostrare come, partendo da dati raccolti quotidianamente dai taxi, sia possibile estrarre un modello di previsione utile per anticipare il valore della durata attesa del viaggio. Come precedentemente espresso, il punto centrale per la corretta risoluzione di questo problema risiede in una attenta estrazione dei dati dal dataset originario e un successivo arricchimento di essi. Nonostante le ulteriori migliorie che potrebbero essere apportate in fase di preprocessing, i risultati ottenuti sono da ritenersi soddisfacenti, sia comparati al modello di random forest utilizzato come benchmark, sia confontando gli esiti ottenuti dagli altri partecipanti alla competizione.

Riferimenti bibliografici

[1] nyc.gov, "Nyc, taxi & limousine commission," sito di statistiche sui yellow cab. [Online]. Available: https://www1.nyc.gov/site/tlc/businesses/yellow-cab.page

[2] kaggle.com, "New york city taxi trip duration," competizione kaggle. [Online]. Available: https://www.kaggle.com/competitions/nyc-taxi-trip-duration/data

[3] "Haversine formula," spiegazione funzione haversine. [Online]. Available: https://en.wikipedia.org/wiki/Haversine_formula

[4] "Geopandas," sito ufficiale geopandas. [Online]. Available: https://geopandas.org/en/stable/

[5] "Shapefile," pagina Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Shapefile

[6] "Directional statistics," pagina Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Directional_statistics

[7] "Lime," pagina GitHub della libreria. [Online]. Available: https://github.com/marcotcr/lime