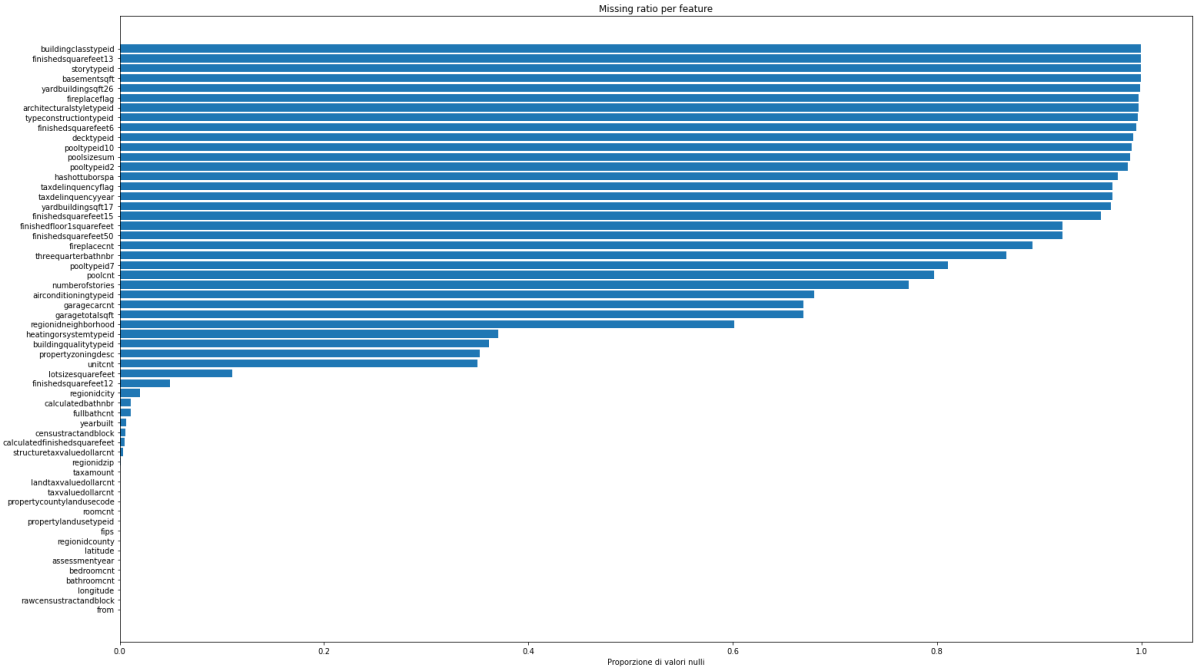
REPORT PROGETTO DATA & WEB MINING

Michele Lotto (875922) e Andrea Chinellato (875422)

Data la mancanza di alcuni valori di test per l'anno 2017, si è scelto di unire i due dataset, dividendoli in train e test campionati casualmente. Tutte le decisioni a venire sono state prese solo sulla base del dataset di train (il test non e' considerato). Solo alla fine e' stato considerato il test per eseguire un'analisi delle istanze più corrette e delle istanze più sbagliate.
Il dataset di validazione non e' stato creato in quanto per ogni modello e' stata effettuata la Cross Validation con `GridsearchCV`.
Il primo problema riscontrato riguarda i dati nulli. È stato disegnato un grafico utile per capire quante features non presentano valori e quindi possono risultare inutili per l'analisi.



DATA PREPARATION & TRANSFORMATION:
Utilizzando il file "zillow_data_dictionary.xlsx" si è provveduto a eseguire un aggiustamento manuale su alcune feature. Come conseguenza della valutazione delle features nulle, sono state suddivise le features quantitative e categoriali in due liste riempiendo rispettivamente i valori nulli con la media e la moda.
Si dichiaria la funzione `resize_range(list)` che prepara il dataset alla fase di "One Hot Encoding", creando una categoria "Altro" in cui ricadono tutte le features presenti nel dataset di test, ma assenti in train. Questa funzione ha la finalità di rendere i due dataset (train e test) compatibili fra loro, al fine di prendere tutte le decisioni guardando solamente il dataset di train.

CORRELAZIONE TRA LOGERROR E FEATURES:
Per quanto riguarda le features quantitative, e' stato creato un grafico di correlazione tra ognuna di queste e la variabile risposta. Si nota che tutte le features sono poco correlate: variano di qualche centesimo attorno allo zero.

PCA (Principal Component Analysis):
Si usa PCA per visualizzare i dati di train, in modo da identificare pattern interessanti: sono stati trovati due agglomerati di dati con colore (logerror) omogeneo. Non risulta particolarmente utile la visualizzazione, sembra che la risposta sia sempre intorno allo 0, al variare delle features. È stata eseguita anche una visualizzazione esclusiva per anno, trovando i medesimi risultati.

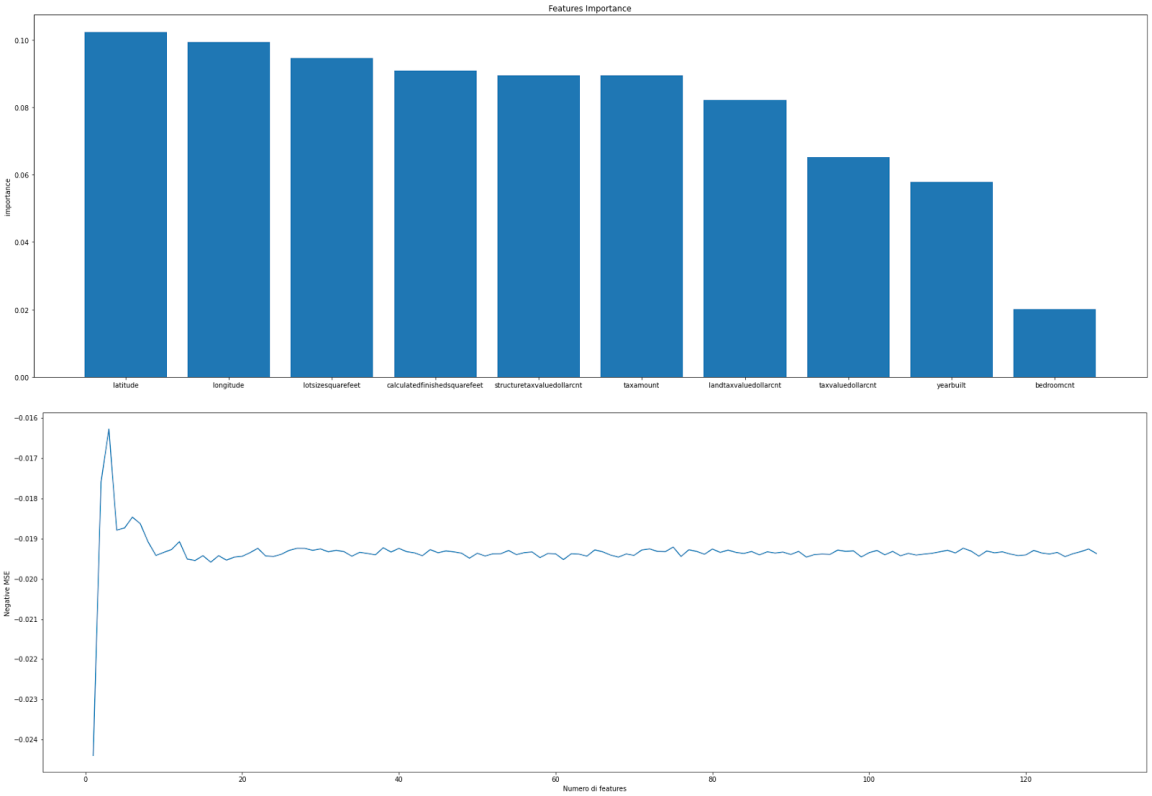
FREQUENCY PLOT LOGERROR:
È stato eseguito un grafico di frequenza per la variabile risposta, la maggior parte dei valori di quest'ultima si distribuisce intorno allo 0. Il 95% dei valori è compreso nell'intervallo (-0.2,0.3).

FEATURE IMPORTANCE e FEATURE SELECTION:
Dato l'elevato numero di features a seguito della fase di DATA PREPARATION & TRANSFORMATION, si cerca di capire (prima della costruzione dei modelli) quali sono le features più importanti per l'analisi. Per fare questo è stata impiegata una Random Forest selezionata tramite `GridSearchCV`. Sono stati testati i seguenti iperparametri (i migliori risultati sono quelli di default):

- min_samples_leaf=1
- min_samples_split=2
- max_features=n_features

- max_leaf_nodes=None (numero illimitato)
- max_samples = X.shape[0] (intero dataset)

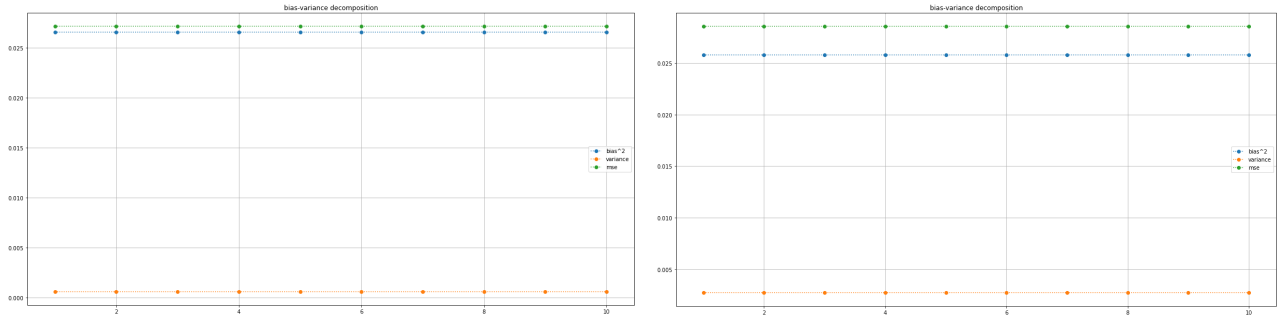
Si plotta ora il grafico di feature importance e si applica la RFECV per diminuire il numero di feature.
Si trovano solamente 3 features significative: ‘longitude’, ‘latitude’ e ‘taxamount’.



MODELLI
Per ogni modello considerato si esegue un tuning dei parametri utilizzando `GridsearchCV`, si plotta la decomposizione del MSE e si esegue una previsione con il dataset di test per un confronto finale dei modelli.
Di seguito sono elencati i modelli con i rispettivi migliori iperparametri:

- Linear Regression:
  - 'fit_intercept': True,
  - 'normalize': True,
  - 'positive': True
- Decision Tree Regressor:
  - 'max_features': 0.95,
  - 'max_leaf_nodes': 340
- Ensemble Methods (AdaBoost Regressor):
  - 'learning_rate': 0.0075,
  - 'n_estimators': 25
- Random Forest Regressor: {parametri di default}

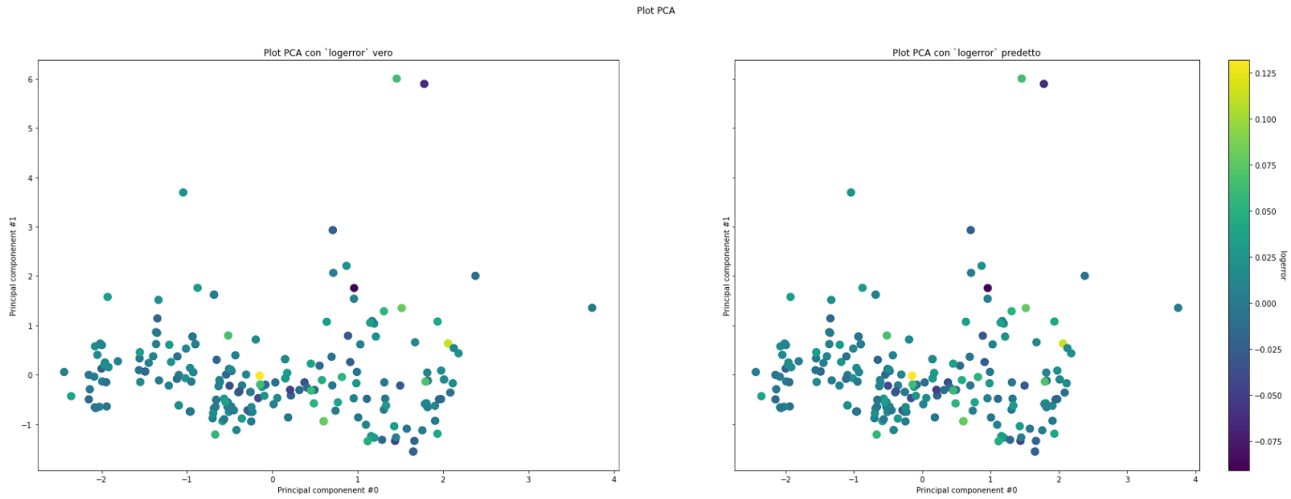
Per ogni modello vengono estratti i valori di bias, varianza e MSE. Tramite l’analisi di bias-variance si individua in tutti i modelli una propensione ad un bias alto con varianza bassa. Si decide quindi di applicare il boosting sul Decision Tree Regressor per ridurre il bias, non ottenendo risultati significativi.
Sulla sinistra `AdaBoost Regressor`, sulla destra `Decision Tree Regressor`:



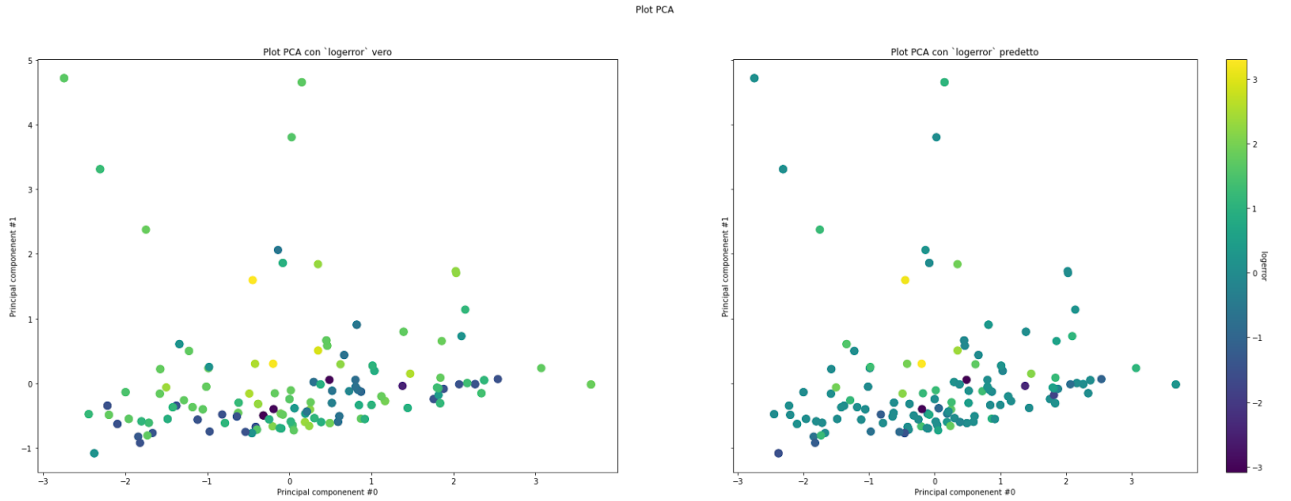
CONCLUSIONI FINALI

Il modello migliore risulta Random Forest regressor con un errore di un centesimo inferiore rispetto agli altri modelli. Tramite la funzione `best_worst_istances(sel)`, si selezionano le prime 200 istanze migliori e le prime 200 istanze peggiori. Tramite la funzione `pca_prop_param(X,Y_true,Y_pred)`, si plottano le istanze migliori, quelle peggiori e le rimanenti. Si evidenzia che il colore (logerror) tra i due grafici si discosta di molto solo nelle istanze peggiori.

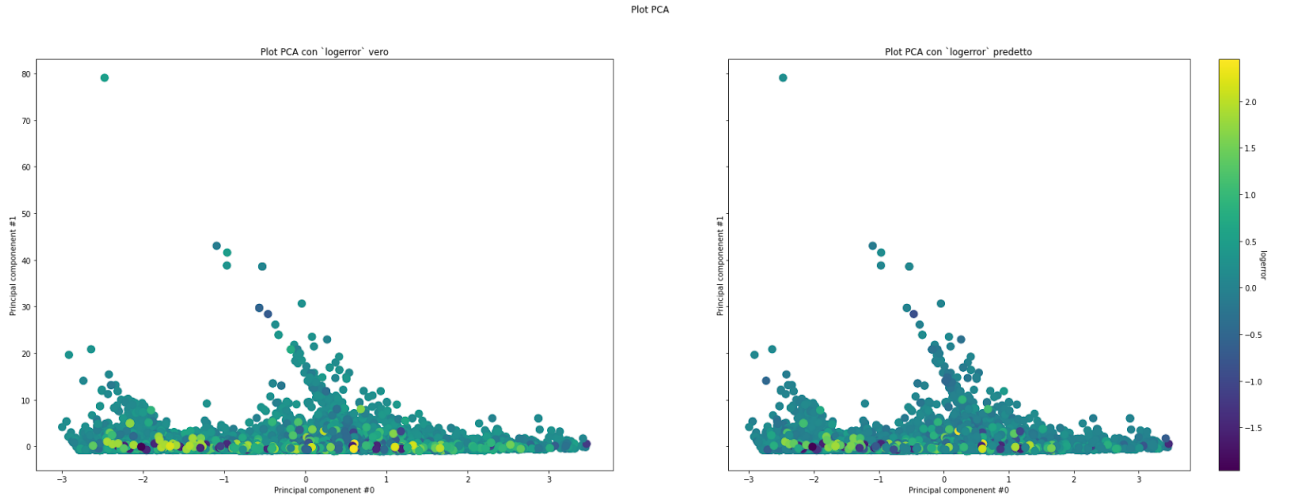
best:



worst:

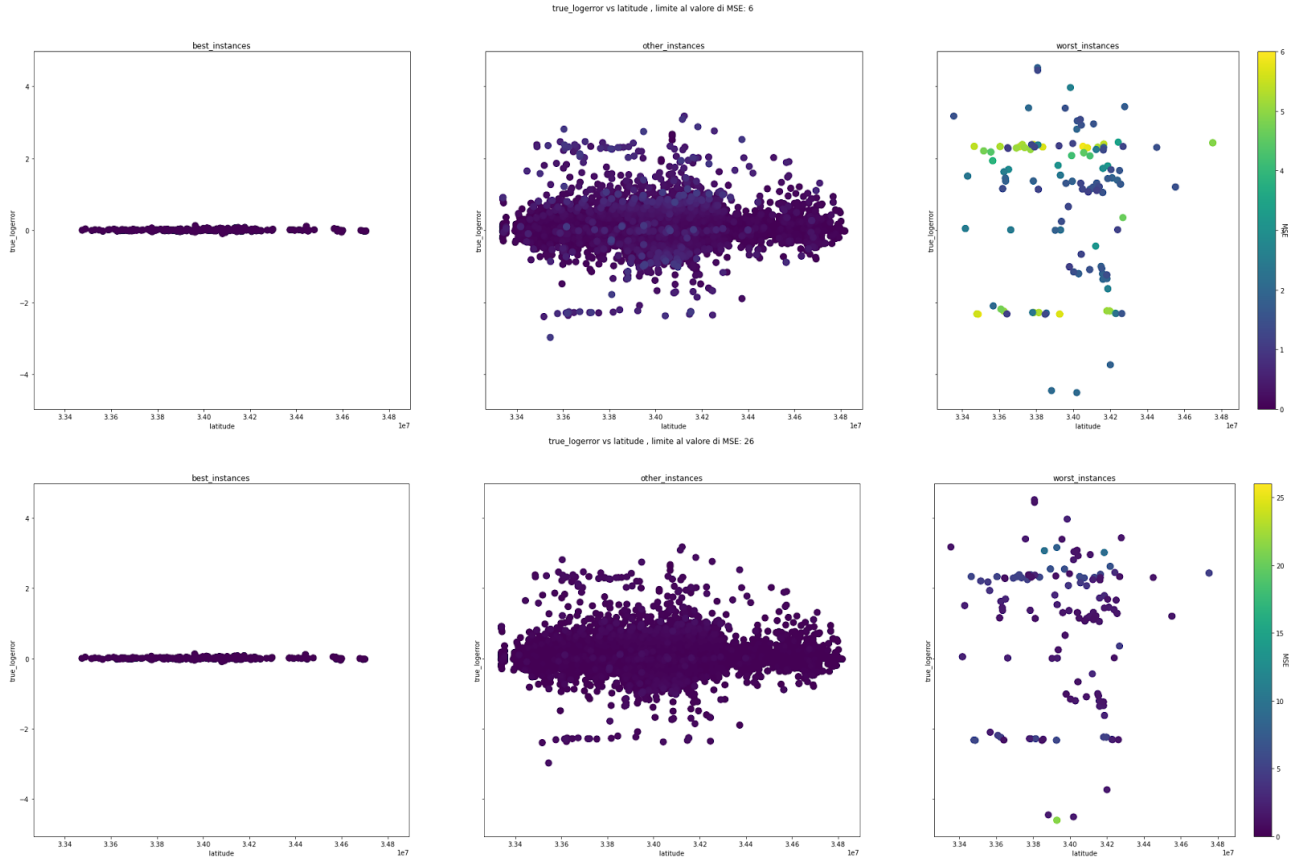


other:

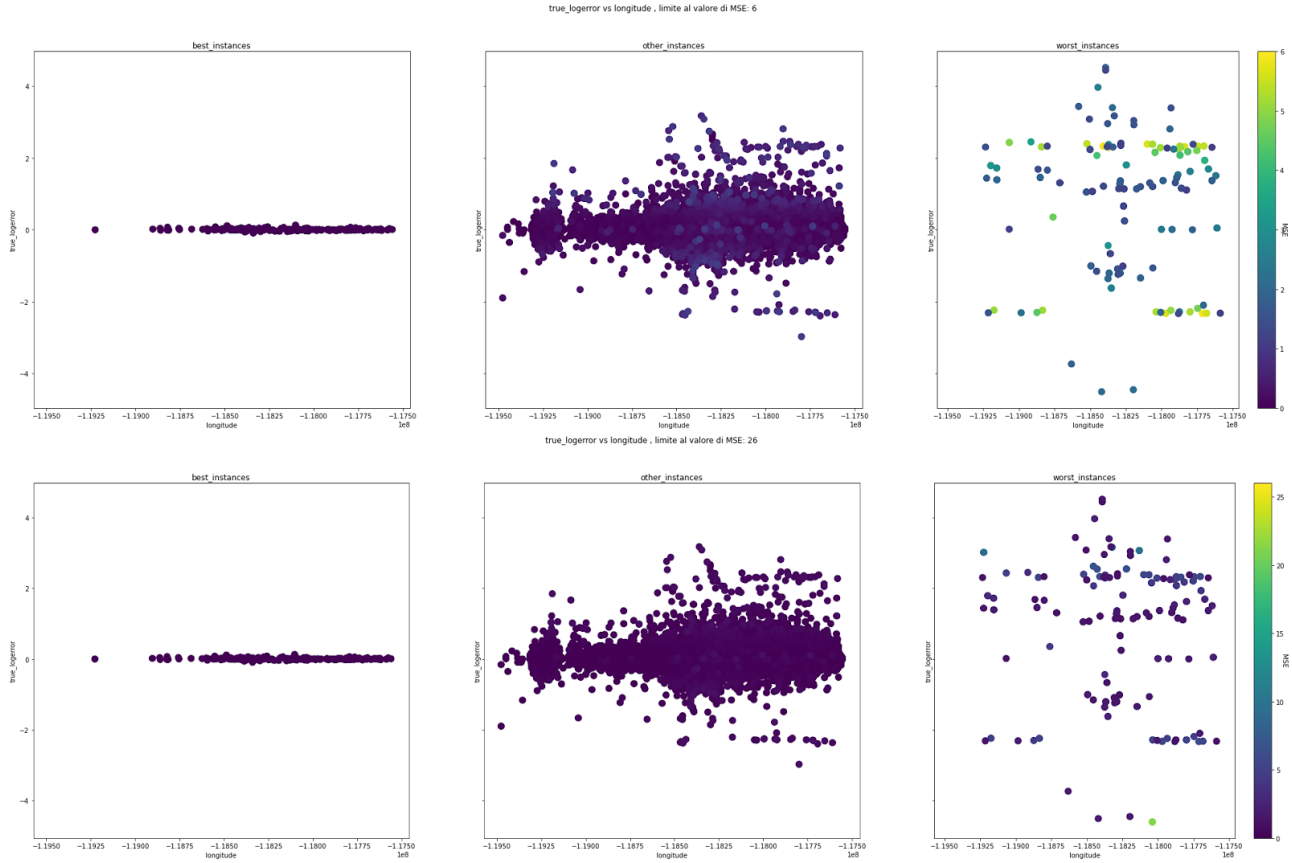


Successivamente con la funzione `plot_var(var,limit)` si plottano `true_logerror vs ogni feature`. Il colore rappresenta il MSE, si limita il suo valore massimo per una migliore visualizzazione, in caso contrario le variazioni di colore si noterebbero solo nel grafico delle istanze peggiori, data l'importante differenza nel MSE.

Latitude:

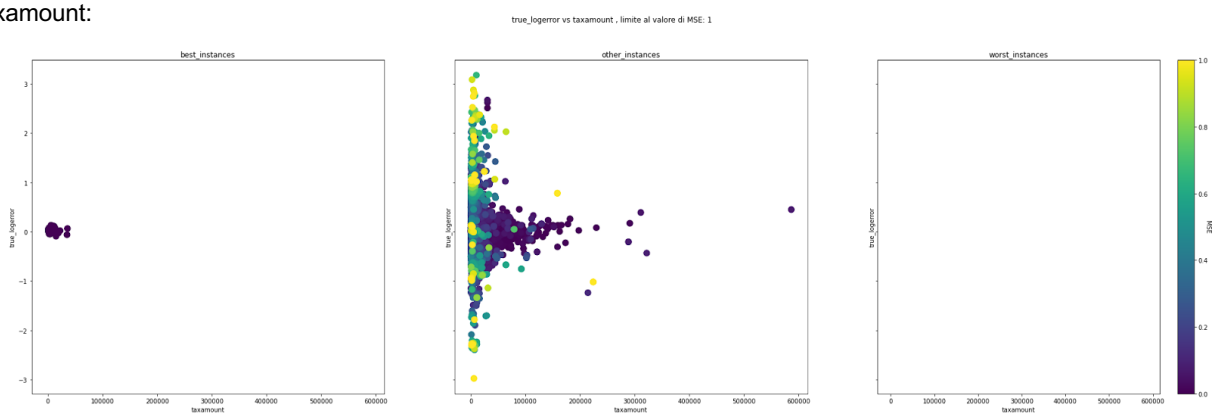


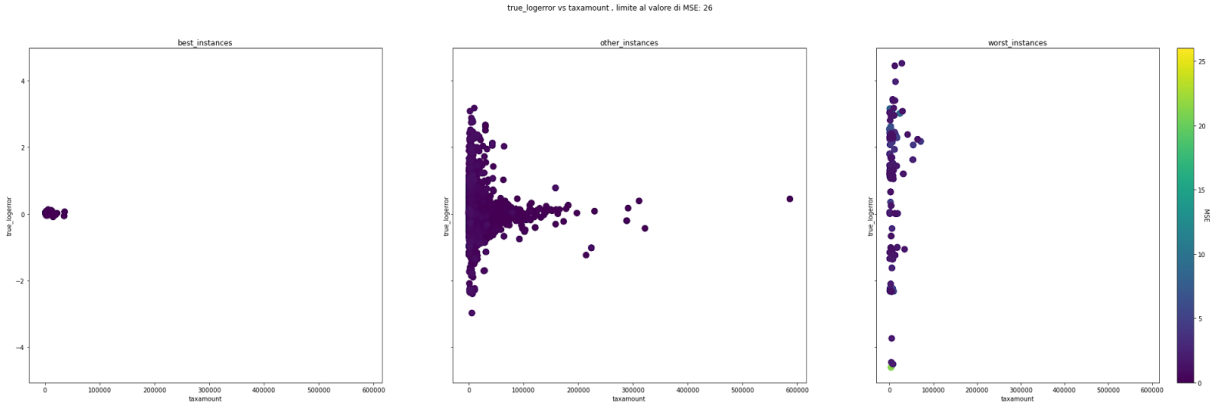
Longitude:



Si nota che le istanze migliori hanno, in media, logerror molto vicino allo zero, diversamente le istanze peggiori presentano, in media, logerror molto distante dallo zero. Il valore della risposta non sembra essere correlato con un aumento del MSE.

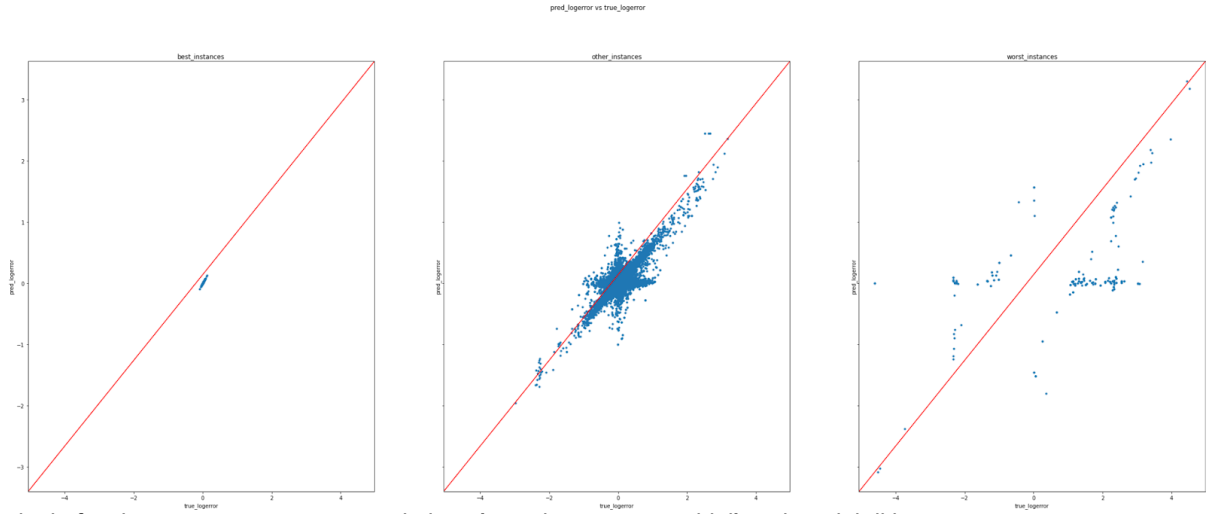
Taxamount:





Infine, è stato notato che `taxamount` si distribuisce come una gaussiana. Inoltre, le istanze migliori sono al centro della gaussiana (valore logerror vicino allo zero), mentre le istanze peggiori sono sulle code (valore logerror vicino allo zero). Il valore di `taxamount` sembra essere sempre < 100000, sia per le istanze migliori che per le peggiori.

Tramite la funzione `plot_pred()`, si plotta `pred_logerror vs true_logerror` per i tre tipi di istanze:



Tramite la funzione `plot_res(limit)`, si plotta `true_logerror vs residui`per i tre tipi di istanze.
Non si nota una forte relazione tra logerror e residui: