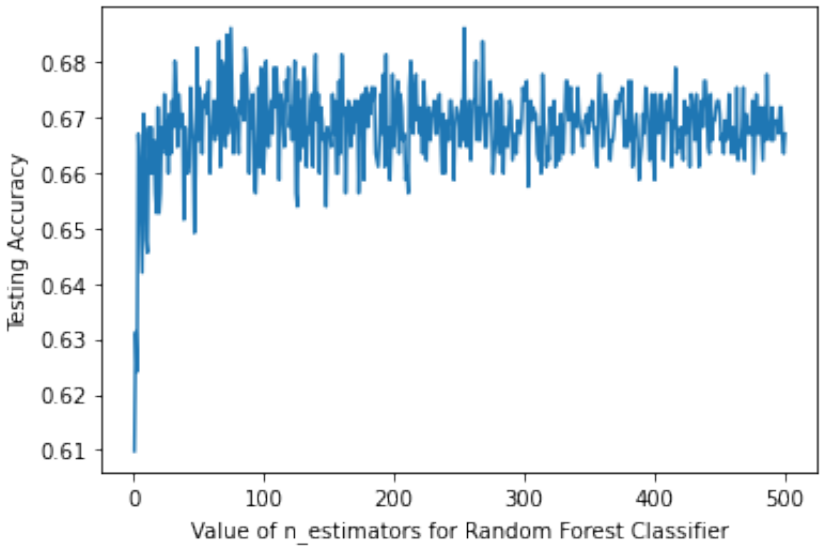
0.1 Random Forest

L'algoritmo *Random Forest* o *Random Decision Forest* è un algoritmo di classificazione e regressione che opera costruendo una moltitudine di alberi di decisione durante la fase di training. Per i task di classificazione l'output sarà la classe più selezionata dagli alberi. Per i task di regressione l'output sarà la media delle predizioni dei singoli task. In questo lavoro, essendo un lavoro di classificazione abbiamo utilizzato il *RandomForestClassifier* di *sckit-learn*. Il lavoro è stato organizzato in questo modo:

1. Utilizzando solo i dati personali degli utenti ed escludendo quelli degli esami superati e dei CFU raggiunti (il dataset bilanciato al momento dell'iscrizione, **??**) sono state provate varie modifiche dei parametri del modello, cambiando il dato relativo al *numero di estimatori.*

2. Successivamente, scelto il parametro che offre risultati migliori sono state provate diverse configurazioni del dataset.

Nella prima fase del lavoro siamo andati quindi a valutare quanto il cambiamento degli iperparametri del modello incidessero sui risultati. Avendo notato come il parametro che incidesse di più sul cambiamento dei risultati sia *n_estimators* e avendo notato come il variare di altri parametri quali *max_depth, min_samples_leaf* incidano poco sul risultato, per motivi di potenza di calcolo, abbiamo preferito effettuare il tuning solo sull'iperparametro n_estimators. Per prima cosa eravamo interessati a vedere di quanto potesse variare l'accuratezza al variare del parametro. Abbiamo, quindi, costruito un grafico che mette in relazione l'accuratezza al numero di estimatori nel modello. I tentativi di calcolare questo rapporto anche fino a valori molto alti fallivano a causa di esecuzioni troppo lunghe per la nostra potenza di calcolo. Per avere un'idea del fenomeno siamo andati quindi a calcolare l'accuracy in base al numero di alberi per tutti i valori da uno a cinquecento. Mostriamo ora il grafico ottenuto:



Come si può notare dal grafico piccole variazioni del parametro possono far variare il risultato di accuratezza ottenuto. Ci siamo chiesti quindi se valori molto piu grandi di questo iperparametro potessero cambiare sensibilmente la situazione. Abbiamo utilizzato in questo caso, il metodo *GridSearchCV*. Tramite questo metodo è possibile definire una griglia di valori di iperparametri sui quali randomicamente verrà fatto train e test di un modello applicando una k-fold cross validation. La ricerca è stata effettuata sui valori da 1 a 500 e su 1000, 10000, 50000, 100000 con un k pari a 3. Alla fine dell'esecuzione abbiamo stampato quindi il valore di *best_params_* che è risultato pari a 207. È stato quindi utilizzato sempre questo come parametro in tutti i modelli creati successivamente. Per garantire, inoltre, una riproducibilità dei risultati è stato impostato un *RandomState* pari a 42.

```
[ ]  print(rf_random.best_params_)

     {'n_estimators': 207}
```