FINAL REPORT
of the penetration testing:
Reconnaissance attack only

Penetration Testing
Target:
DATAVISION

Company:
DataVision s.r.o.

Supervisor:
Martin Podhola

Performed by:
Gabriele Tassinari

Date: June 2024

Indice

1 Introduzione

2 No idea

3 No idea

4 Modello di Predizione

4.1 Introduzione

L'obiettivo principale di questo progetto è sviluppare un modello di machine learning capace di predire l'esito delle partite NBA. Questo rappresenta una sfida ambiziosa che richiede un'analisi approfondita delle statistiche delle squadre e l'applicazione di avanzate tecniche di machine learning. Il dataset utilizzato comprende un array di statistiche per ciascuna squadra, sia per la squadra di casa che per quella in trasferta.

Sono stati testati sette differenti modelli sul set di training: Multi-Layer Perceptron (MLP), Random Forest, Support Vector Machine (SVM), XGBoost, Regressione Lineare, Modello Bayesiano e K-Nearest Neighbors (KNN).

Per sviluppare un modello competitivo rispetto alle soluzioni attualmente disponibili sul mercato, è stato inizialmente affrontato il problema come una questione di classificazione, utilizzando un MLP. Dato che i risultati ottenuti erano promettenti, sono state poi esplorate ulteriori possibilità per migliorare le prestazioni del modello. Di conseguenza, è stato convertito il problema di classificazione in un problema di regressione, con l'obiettivo di prevedere non solo l'esito (vittoria o sconfitta) ma anche il margine di punti con cui una squadra vince o perde.

L'approccio di regressione offre un livello di dettaglio superiore, consentendo non solo di determinare il vincitore della partita, ma anche di fornire una stima più precisa delle prestazioni delle squadre. Questa dualità di approccio, classificazione e regressione, ci permette di ottenere un modello robusto e versatile, in grado di adattarsi a diverse esigenze di predizione nel contesto delle partite NBA.

In sintesi, il progetto si articola in due fasi principali:

1. Creazione e valutazione di un modello di classificazione tramite MLP.
2. Espansione del modello in un contesto di regressione per migliorare la precisione delle predizioni.

I risultati ottenuti dalle varie sperimentazioni con i diversi modelli saranno discussi nelle sezioni successive, evidenziando i vantaggi e le limitazioni di ciascun approccio.

4.2 Il Modello di Classificazione

4.2.1 MLP

Introduzione al MLP Il Multi-Layer Perceptron (MLP) è una classe di reti neurali artificiali ampiamente utilizzata nei problemi di classificazione e regressione. Un MLP è composto da uno strato di input, uno o più strati nascosti e uno strato di output. Ogni nodo (o neurone) in uno strato è connesso a ciascun nodo nel successivo strato, rendendolo un tipo di rete completamente connessa. I MLP utilizzano la retropropagazione per allenare il modello, aggiornando i pesi dei neuroni in base all'errore commesso nelle predizioni.

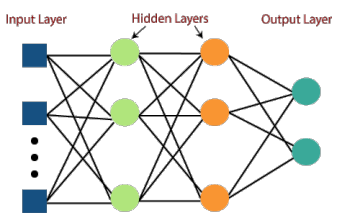

Figura 1: Multi-Layer Perceptron

Sviluppo del Modello Il primo approccio analizzato è stato quello di creare un MLP per risolvere il problema di classificazione. Il primo passo è stato quello di effettuare il preprocessing dei dati, eliminando tutte le informazioni non necessarie per l'allenamento della rete o eliminando informazione che rendevano il modello meno accurato e dunque meno performante. In particolare, sono stati rimossi i seguenti campi:
`['pts_H', 'pts_A', 'referee_id', 'winner', 'home_team', 'away_team', 'referee_name', 'season', 'date']`

Inoltre prima di poter utilizzare il vettore finale, composto dunque dalle statistiche delle due squadre, per allenare il nostro modello, è stato necessario preprocessarlo. È stato quindi usato il `MinMaxScaler`, ovvero un algoritmo che garantisce che i valori fossero compresi tra -1 e 1, questo ha portato notevoli miglioramenti sia in termini di accuracy del modello sia in termini di performance della rete.

Struttura del Modello Il modello di machine learning è stato quindi costruito utilizzando tre strati `Dense`, ovvero strati fortemente connessi in grado di apprendere dai dati in input per ottenre un preciso output. La rete neurale ottenuta è stata quindi addestrata, come consuetudine, utilizzando l'80% del Dataset suddiviso il Train Set e Validation Set. Per migliorare ulteriormente il modello è stato eseguito un processo di fine-tuning manuale per massimizzare l'accuratezza.

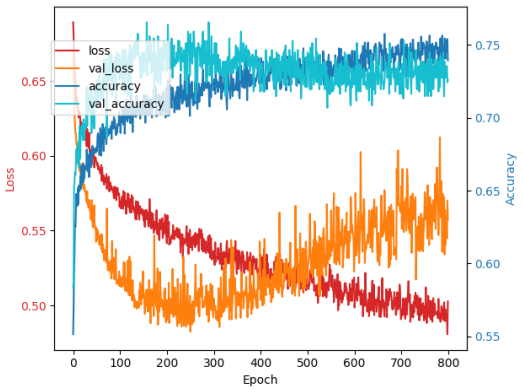

Figura 2: Pre Fine-Tuning e Overfitting

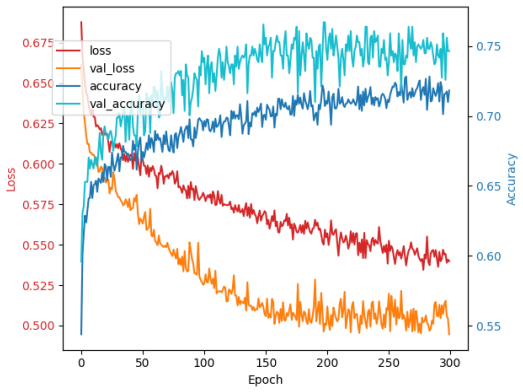

Figura 3: Post Fine-Tuning

Inizialmente, è stato utilizzato un *batch size* di 16 e sono state impostate 800 epoche per monitorare la funzione di *loss* e identificare il punto di *overfitting* del modello. Attraverso questo processo, è stato determinato che i parametri ideali per l'allenamento erano 300 epoche e un *batch size* di 32.

Risultati Prima di poter analizzare questo risultato occorre precisare che l'accuracy fornita dai modelli citati online risulta essere tra il 62% ed il 66%. In particolare è presente un interessante articolo, redatto dalla Bryant University, nel quale presentano il loro modello che possiede una accuracy del 65%.

In quest'ottica è possibile dire che il nostro modello finale ha ottenuto un'accuratezza estremamente competitiva, ottenendo una accuracy del 69,92%, dimostrando così la validità dell'approccio MLP nel contesto della classificazione delle partite NBA. Questi risultati indicano che il nostro modello è in grado di fornire predizioni accurate (7 volte su 10) sulla vittoria o sconfitta delle squadre, basandosi sulle statistiche pre-partita.

In sintesi, l'implementazione del MLP come modello di classificazione ha fornito una solida base per ulteriori miglioramenti e ottimizzazioni, consentendo di esplorare ulteriori tecniche e modelli per migliorare le capacità predittive.

Results MLP:

```
Loss: 0.5629295706748962
Accuracy: 0.6992385983467102
```

4.3 I Modelli di Regressione

Il problema di classificazione è stato poi convertito in un problema di regressione, in modo da testare nuovi approcci e verificare se il modello ottenuto potesse essere ulteriormente migliorato. Tutti i modelli analizzati in questa sezione sono stati prima ottimizzati utilizzando una GridSearch per ottenere la combinazione ideale di iperparametri per ciascun modello. Verranno quindi analizzati i vari modelli e i risultati ottenuti, in ordine crescente di accuracy fornita.

4.3.1 Analisi delle metriche

Nel valutare le prestazioni del modello di regressione per la predizione dei risultati delle partite NBA, sono stati utilizzati diversi indicatori chiave di errore e precisione.

- **Mean Absolute Error (MAE):** Il MAE misura la deviazione media tra le predizioni del modello e i valori reali. Nel nostro caso, il MAE indica dunque l'errore medio tra la differenza di punti predetti dal modello e i risultati reali delle partite.
- **Mean Squared Error (MSE):** Il MSE misura la media dei quadrati delle differenze tra predizioni e valori reali. Il MSE evidenzia dunque una variabilità nelle differenze tra predizioni e valori reali rispetto al MAE.
- **Root Mean Squared Error (RMSE):** Il RMSE è la radice quadrata del MSE ed è espresso nelle stesse unità della variabile target. Nel nostro caso, il RMSE indica che in media le predizioni del modello hanno un errore di una certa quantità di punti rispetto ai risultati reali.
- **Sign Accuracy:** La Sign Accuracy rappresenta la precisione del modello nel predire correttamente la direzione (positiva o negativa) delle differenze tra predizioni e valori reali. Questa metrica misura se il segno tra la differenza dei punti predetti e quella reale è concorde, in modo da prevedere quindi il vincitore di una partita indipendentemente dalla differenza dei punti.

Queste metriche forniscono una valutazione completa delle performance del modello di regressione, evidenziando sia l'errore medio delle predizioni che la sua capacità di predire correttamente l'esito delle partite NBA. Il confronto tra MAE, MSE e RMSE fornisce una panoramica dell'accuratezza delle predizioni a diversi livelli di dettaglio, mentre la Sign Accuracy fornisce una misura della precisione nella classificazione della direzione delle differenze.

4.3.2 KNN

Il K-Nearest Neighbors (KNN) è un algoritmo di machine learning utilizzato per problemi di classificazione e regressione. Nel contesto della regressione, KNN prevede il valore di una variabile target basandosi sui valori medi dei k vicini più prossimi nel dataset di addestramento. La vicinanza tra i punti dati è solitamente calcolata tramite una metrica di distanza, come la distanza euclidea. KNN è semplice da implementare e interpretare, ma può essere computazionalmente costoso per dataset di grandi dimensioni.

Il modello KNN è stato valutato utilizzando diverse metriche di errore, ottenendo i seguenti risultati:

```
Mean Absolute Error (MAE): 11.620278833967047
Mean Squared Error (MSE): 215.66732572877058
Root Mean Squared Error (RMSE): 14.6856162869922
Sign Accuracy: 0.6172370088719898
```

Il modello ha raggiunto un'accuratezza del 61,72%.

Sono stati estratti 10 match casuali dal test set per valutare le prestazioni del modello, ottenendo i seguenti risultati:

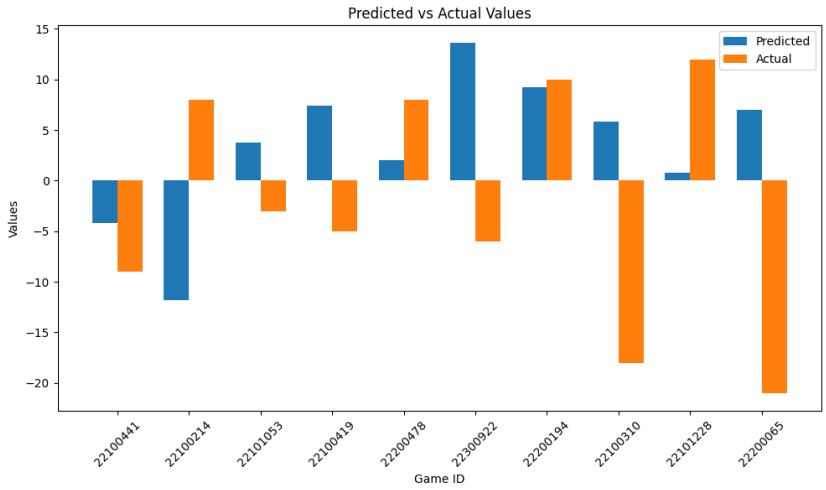

Figura 4: Comparazione differenza punti predetti rispetto a reali in KNN

4.3.3 SVM

Il modello SVM ha mostrato un'accuratezza migliore rispetto ad altri approcci. SVM (Support Vector Machine) è un algoritmo di apprendimento supervisionato utilizzato per la classificazione e la regressione. Esso cerca di trovare il miglior iperpiano o separatore tra i punti dei dati delle diverse classi. Nella figura sottostante è rappresentato un esempio di SVM con due classi.

Questo modello ha ottenuto i seguenti risultati nei test:

```
Mean Absolute Error (MAE): 10.360566704859455
Mean Squared Error (MSE): 173.73226432418772
Root Mean Squared Error (RMSE): 13.18075355676555
Sign Accuracy: 0.6806083650190115
```

Il modello ha raggiunto un'accuratezza del 68.06%. Inoltre, sono stati estratti 10 match casuali dal set di test per valutare ulteriormente le prestazioni del modello:

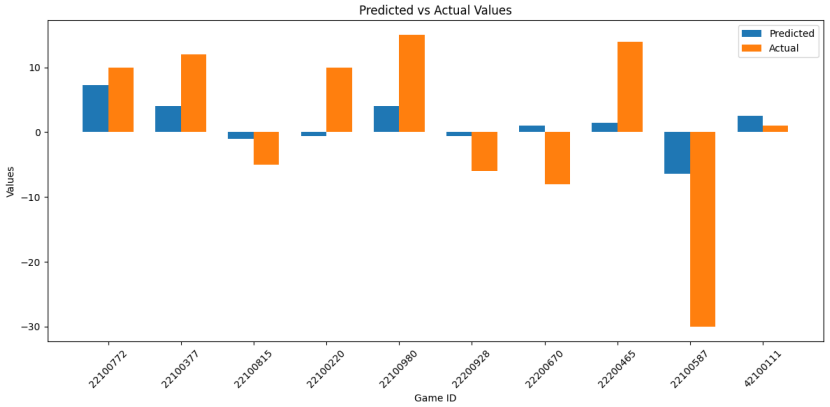

Figura 5: Comparazione differenza punti predetti rispetto a reali in SVM

Questi risultati dimostrano che il modello SVM ha mostrato una buona capacità predittiva per l'esito delle partite NBA, con una precisione significativa e una buona gestione della variazione nei dati di test.

4.3.4 Random Forest

Il Random Forest è un metodo di apprendimento ensemble utilizzato per la classificazione e la regressione. Esso costruisce diversi alberi decisionali e li combina per ottenere previsioni più accurate. In un random forest, solo un sottoinsieme delle caratteristiche è considerato dall'algoritmo per dividere un nodo. Il modello classificherà anche l'importanza di ciascuna caratteristica nel prendere la decisione finale. Nella figura sottostante c'è un esempio di random forest con due alberi.

Questo modello ha ottenuto risultati soddisfacenti nei test, come mostrato di seguito:

```
Mean Absolute Error (MAE): 9.944223032615545
Mean Squared Error (MSE): 159.938928930258
Root Mean Squared Error (RMSE): 12.646696364278617
Sign Accuracy: 0.6958174904942965
```

Il modello ha raggiunto un'accuratezza del 69.58%. Inoltre, sono stati estratti 10 match casuali dal set di test per valutare ulteriormente le prestazioni del modello:

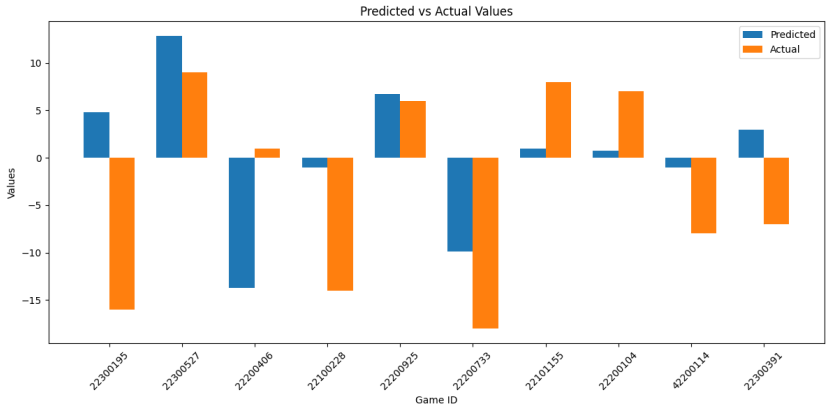

Figura 6: Comparazione differenza punti predetti rispetto a reali in Random Forest

Questi risultati indicano che il modello Random Forest ha dimostrato di essere efficace nel predire l'esito delle partite NBA, con un'accuratezza significativa e una buona capacità di generalizzazione su nuovi dati di test.

4.3.5 XGBoost

XGBoost è un algoritmo di boosting estremamente popolare e efficace utilizzato per la classificazione e la regressione. Utilizza un insieme di alberi decisionali deboli, chiamati "weak learners", e li combina per migliorare progressivamente le prestazioni del modello.

Questo modello ha ottenuto i seguenti risultati nei test:

```
Mean Absolute Error (MAE): 9.6111730557072
Mean Squared Error (MSE): 149.89162676035912
Root Mean Squared Error (RMSE): 12.243023595515902
Sign Accuracy: 0.7046894803548795
```

Il modello ha raggiunto un'accuratezza del 70.47%. Inoltre, sono stati estratti 10 match casuali dal set di test per valutare ulteriormente le prestazioni del modello:

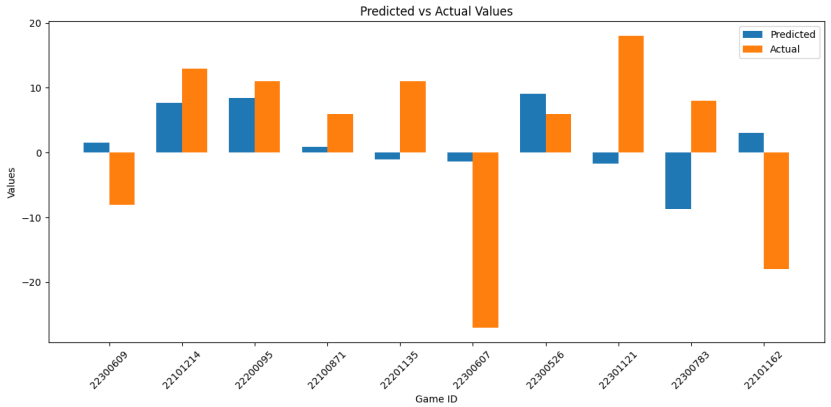

Figura 7: Comparazione differenza punti predetti rispetto a reali in XGBoost

Questi risultati indicano che il modello XGBoost ha dimostrato di essere altamente efficace nel predire l'esito delle partite NBA, con un'accuratezza significativa e una buona capacità di generalizzazione su nuovi dati di test. L'utilizzo di boosting ha permesso al modello di migliorare costantemente la precisione delle predizioni, rendendolo una scelta potente per problemi complessi come quello delle previsioni sportive.

4.3.6 Bayesian

Il modello Bayesian utilizza l'inferenza Bayesiana per stimare parametri incogniti. È particolarmente utile quando si hanno dati limitati e si desidera incorporare conoscenze pregresse o informative nel modello.

Questo modello ha ottenuto i seguenti risultati nei test:

```
Mean Absolute Error (MAE): 9.684169695449112
Mean Squared Error (MSE): 151.8468062429373
Root Mean Squared Error (RMSE): 12.322613612498662
Sign Accuracy: 0.7072243346007605
```

Il modello ha raggiunto un'accuratezza del 70.72%. Inoltre, sono stati estratti 10 match casuali dal set di test per valutare ulteriormente le prestazioni del modello:

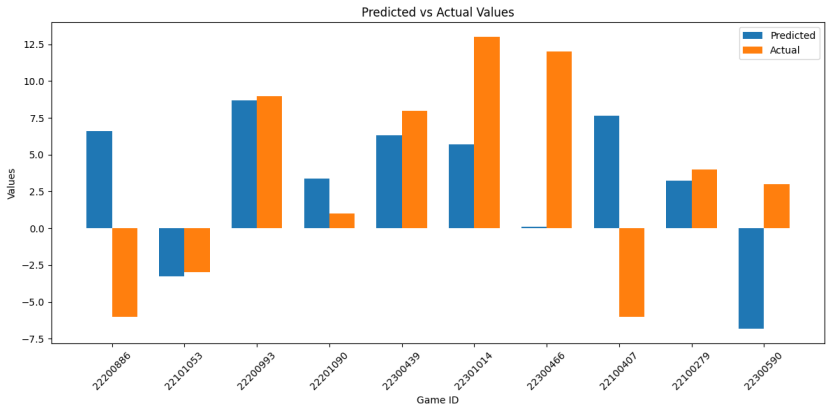

Figura 8: Comparazione differenza punti predetti rispetto a reali in Bayesian

Questi risultati indicano che il modello Bayesian ha mostrato una buona capacità predittiva per l'esito delle partite NBA, con un'accuratezza significativa e una robusta gestione dell'incertezza nei dati

di test. L'approccio basato sull'inferenza Bayesiana ha permesso al modello di integrare informazioni a priori con i dati osservati, migliorando così le performance complessive delle previsioni.

4.3.7 Linear Regression

La regressione lineare è un modello di machine learning che cerca di trovare la relazione lineare migliore tra una variabile dipendente (target) e una o più variabili indipendenti (features). È un metodo semplice ma potente per la predizione numerica.

Questo modello ha ottenuto i seguenti risultati nei test:

```
Mean Absolute Error (MAE): 9.673524553632301
Mean Squared Error (MSE): 151.58868540348476
Root Mean Squared Error (RMSE): 12.312135696274824
Sign Accuracy: 0.7135614702154626
```

Il modello ha raggiunto un'accuratezza del 71.36%. Inoltre, sono stati estratti 10 match casuali dal set di test per valutare ulteriormente le prestazioni del modello:

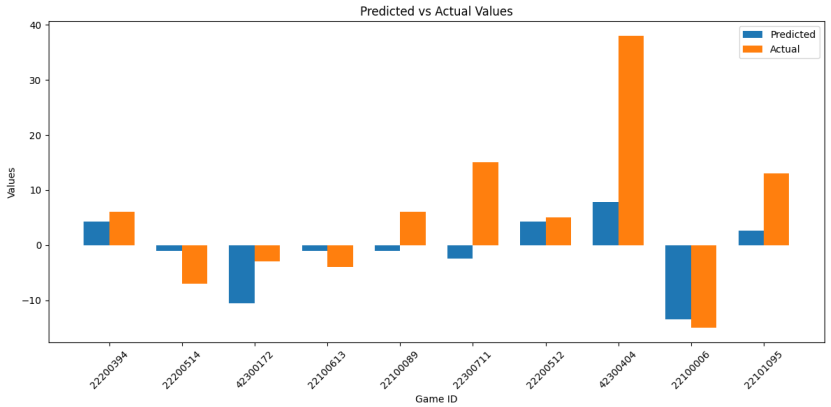

Figura 9: Comparazione differenza punti predetti rispetto a reali in Linear Regression

Questi risultati indicano che il modello di Regressione Lineare ha mostrato una buona capacità predittiva per l'esito delle partite NBA, con un'accuratezza significativa e un'efficace modellazione della relazione lineare tra le variabili di input e il target. La semplicità e la trasparenza della regressione lineare lo rendono un buon punto di partenza per esplorare i modelli di previsione numerica nelle analisi sportive.

4.4 I Risultati

È dunque possibile ora analizzare i dati in maniera unitaria e completa. Per effettuare ciò occorre suddifidere l'analisi dei risultati in due distente sezioni, una per analizzare i risultati inerenti al problema di classificazione: Vittoria o Sconfitta, ed un'altra per analizzare i risultati del problema di regressione.

Inerentemente al problema di classificazione, per gli algoritmi di regressione è stata creata una nuova metrica fittizia denominata sign accuracy che ci ha permesso di convertire il problema di regressione in classificazione. Comparando così le varie sign accuracy ottenute con i vari algoritmi, otteniamo il seguente risultato.

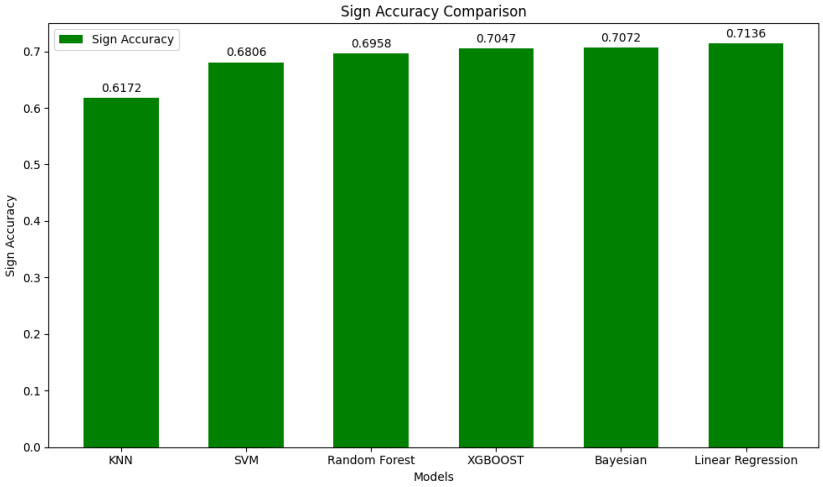

Figura 10: Sign accuracy a confronto

È possibile notare che l'algoritmo che ci fornisce una sign accuracy migliore è il Linear Regression che totalizza una accuracy del 71,36%. Questo è poi seguito dal Bayesian Regressor con 70,72% e dal XGBoost Regressor con 70,46%. Questi risultati algoritmi sono quindi riusciti ad ottenere, a parità di input, un risultato migliore anche del modello MLP creato. In ultima posizione vediamo come il K-Nearest Neighbor abbia le prestazioni peggiori.

Concentrandoci ora sull'analisi del problema di regressione e dunque sull'analisi della differenza tra i punti predetti ed i punti reali otteniamo risultati leggermente diversi. A tal proposito è stato confrontata la metrica: Mean Absolute Error, ovvero la metrica in grado di dare informazioni riguardanti l'errore medio dei vari modelli creati. Confrontando così questa metrica sono stati ottenuti questi risultati.

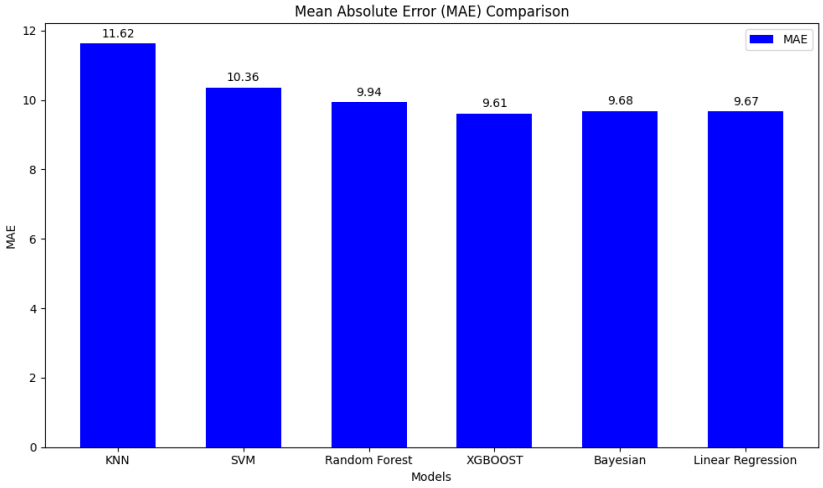

Figura 11: Caption

L'algoritmo in termini di regressione più performante risulta essere XGBoost con 9,61 punti di scarto medio, seguito poi dalla Linear Regression con 9,67 punti e dal Bayesian con 9,68. Il K-Nearest Neighbor al contrario continua a confermarsi il meno performante con 11,62 punti di scarto medio.

4.5 Importanza Metriche

I modelli di regressione sono stati poi confrontati per comprendere quali fossero le features che ciascun modello usava ed analizzava maggiormente per prevedere la differenza di punteggio delle partite. Nela figura sottostante sono messi a confronto i 3 modelli migliori (ovvero bayesian, xgboost e linear regression) ed il modello peggiore ovvero KNN.

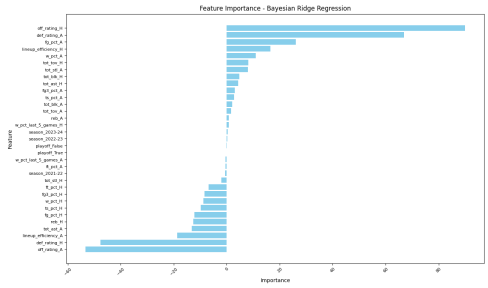

Figura 12: Bayesian

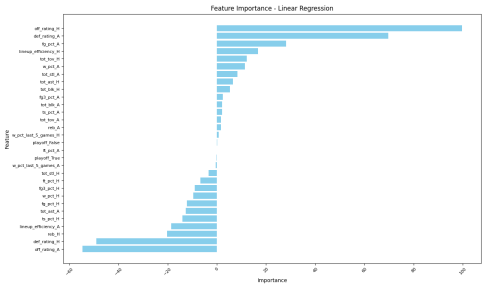

Figura 13: Linear Regression

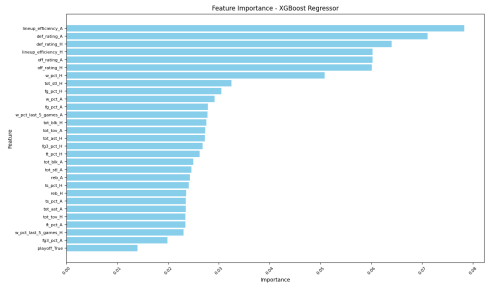

Figura 14: XGBoost

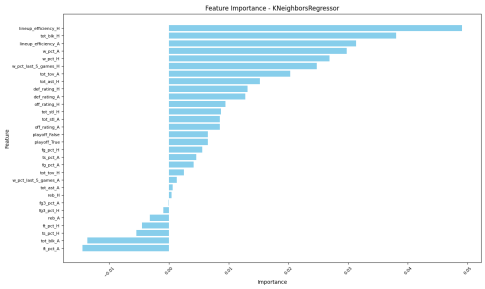

Figura 15: K-Nearest Neighbors

Si può vedere come i 3 modelli migliori (ovvero bayesian, xgboost e linear regression) valutano maggiormente e dunque prediligono le seguenti metriche:

- lineup_efficiency_A
- lineup_efficiency_H
- off_rating_A
- off_rating_H
- def_rating_H
- def_rating_A

Al contrario del modello di regressione K-Nearest Neighbors che mantiene solo le lineup_efficiency ed altre feature a scapito di: off_rating_A off_rating_H def_rating_A def_rating_H

5 Conclusione

6 Appendice