Horse Racing Data Analysis

- 22.10.2022

Team Leo!
Ishita PES1UG20CS648, Hita PES1UG20CS645

Line Up

- An Introduction
- EDA
- Clustering
- Modelling
- Results
- Next Up

Ever seen a horse race? Have a look now ⬅️

https://www.youtube.com/watch?v=wIYD42DV3Ro&

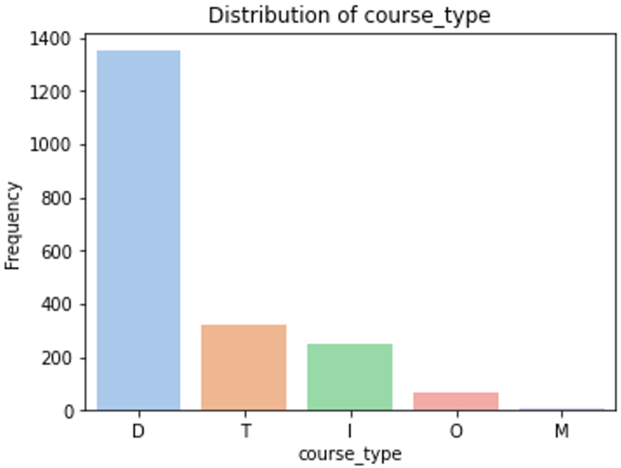
Introduction

- America’s most prestigious and oldest horse races are held annually at Aqueduct, Belmont, Saratoga.
- Each horse is tracked every 0.25 seconds using trakus index (real-time tracking system that determines the exact geographical location of the horse.

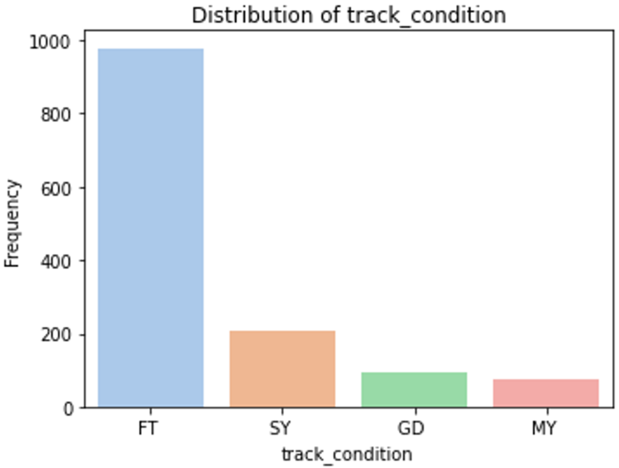
Some nuggets 🍗

1. Purse
2. Drafting Strategy
3. Maiden Races
4. Stakes
5. Allowance Races
6. Claiming Races
7. Odds
8. Jockey
9. Race Course Types - Hurdle Surfaces, Turf and Dirt
10. Track Conditions - Muddy, Sloppy, Firm, Yielding

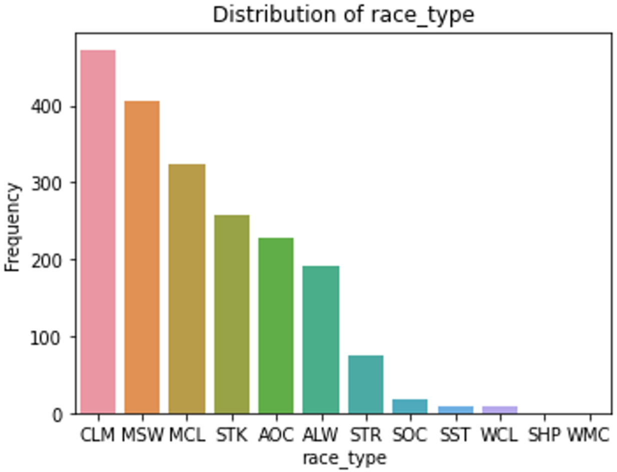
EDA



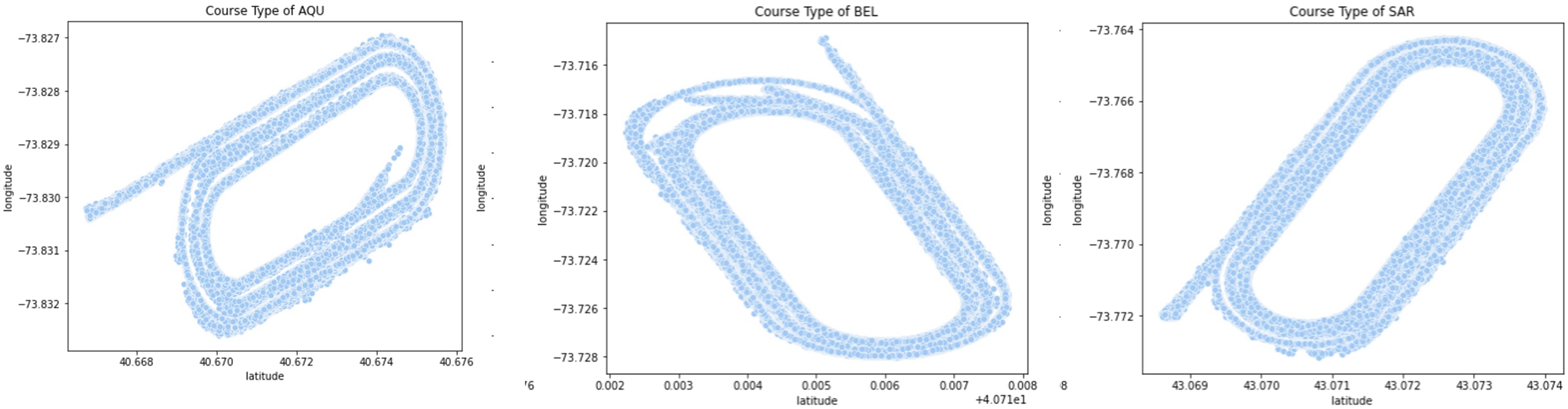
D - Dirt
T - Turf
I - Inner Turf
O - Outer Turf



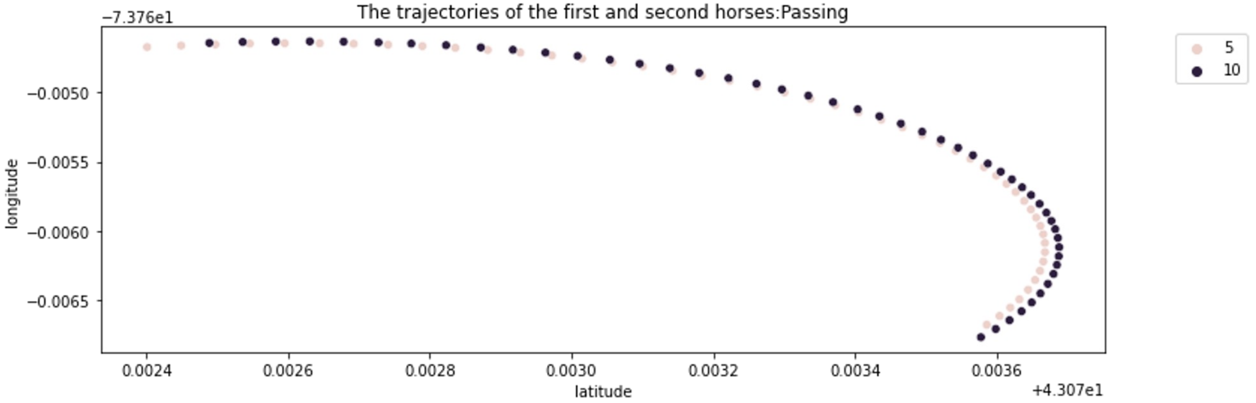
FT - Fast Track
SY - Sloppy
M - Muddy
GD - Good



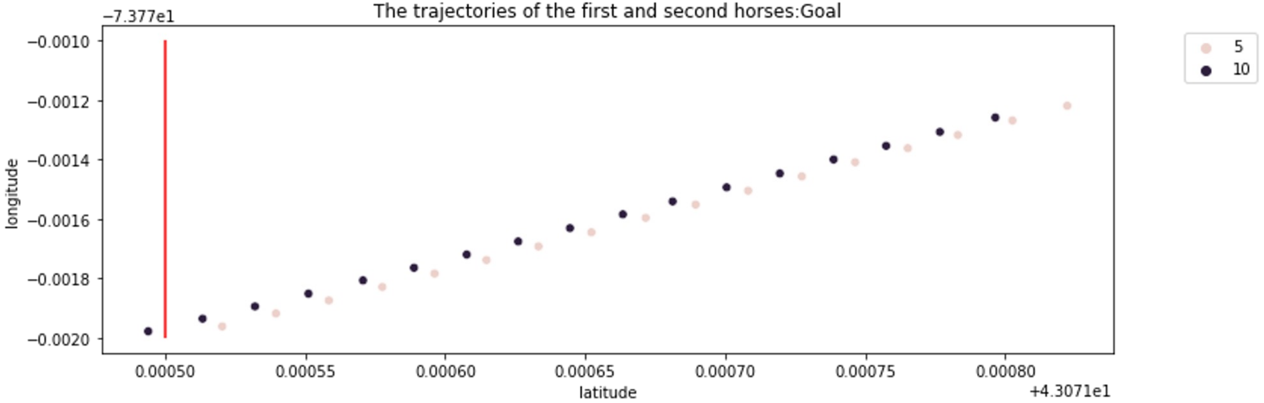
CLM - Claiming
MSW - Maiden Special Weight
STK - Stakes



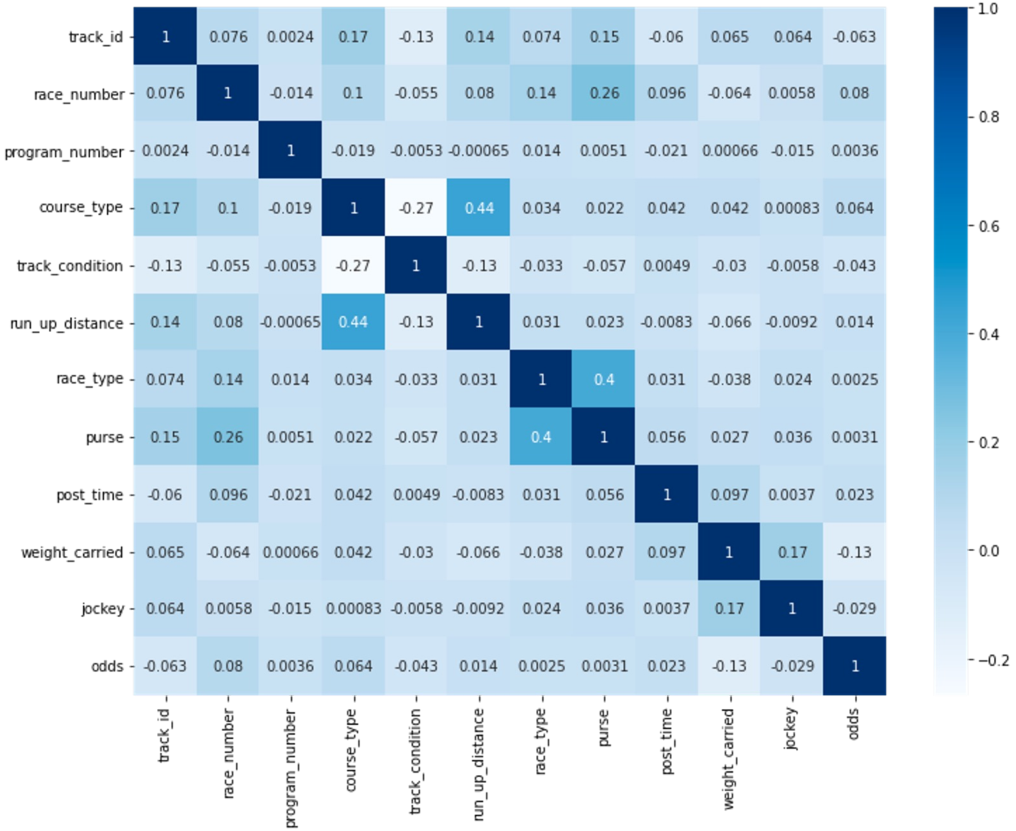
Scatterplots of the racing track at the 3 locations, Aqueduct(AQU), Belmont(BEL) and Saratoga(SAR)



Scatterplot of the trajectory followed by the winner and runner-up horse at the finish line.



Close up of the first horse overtaking the second.



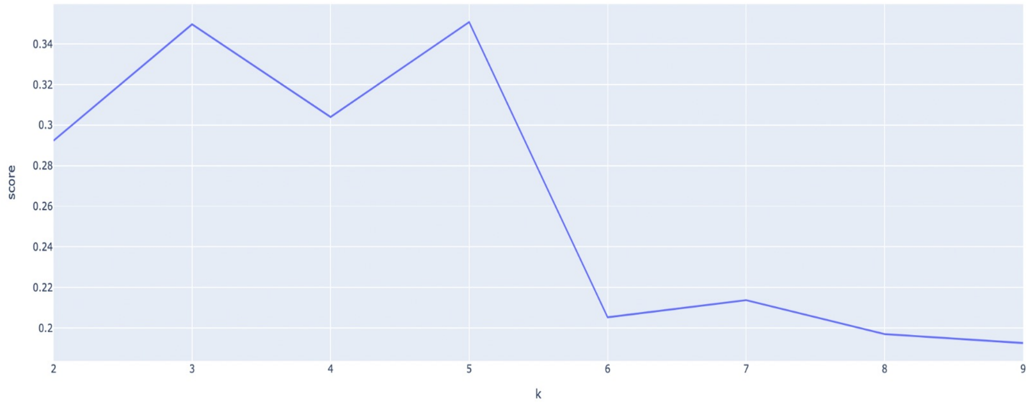
	track_id	race_number	program_number	course_type	track_condition	run_up_distance	race_type	purse	post_time	weight_carried	jockey	odds
track_id	1	0.076	0.0024	0.17	-0.13	0.14	0.074	0.15	-0.06	0.065	0.064	-0.063
race_number	0.076	1	-0.014	0.1	-0.055	0.08	0.14	0.26	0.096	-0.064	0.0058	0.08
program_number	0.0024	-0.014	1	-0.019	-0.0053	-0.00065	0.014	0.0051	-0.021	0.00066	-0.015	0.0036
course_type	0.17	0.1	-0.019	1	-0.27	0.44	0.034	0.022	0.042	0.042	0.00083	0.064
track_condition	-0.13	-0.055	-0.0053	-0.27	1	-0.13	-0.033	-0.057	0.0049	-0.03	-0.0058	-0.043
run_up_distance	0.14	0.08	-0.00065	0.44	-0.13	1	0.031	0.023	-0.0083	-0.066	-0.0092	0.014
race_type	0.074	0.14	0.014	0.034	-0.033	0.031	1	0.4	0.031	-0.038	0.024	0.0025
purse	0.15	0.26	0.0051	0.022	-0.057	0.023	0.4	1	0.056	0.027	0.036	0.0031
post_time	-0.06	0.096	-0.021	0.042	0.0049	-0.0083	0.031	0.056	1	0.097	0.0037	0.023
weight_carried	0.065	-0.064	0.00066	0.042	-0.03	-0.066	-0.038	0.027	0.097	1	0.17	-0.13
jockey	0.064	0.0058	-0.015	0.00083	-0.0058	-0.0092	0.024	0.036	0.0037	0.17	1	-0.029
odds	-0.063	0.08	0.0036	0.064	-0.043	0.014	0.0025	0.0031	0.023	-0.13	-0.029	1

Correlation heat map of chosen relevant attributes.

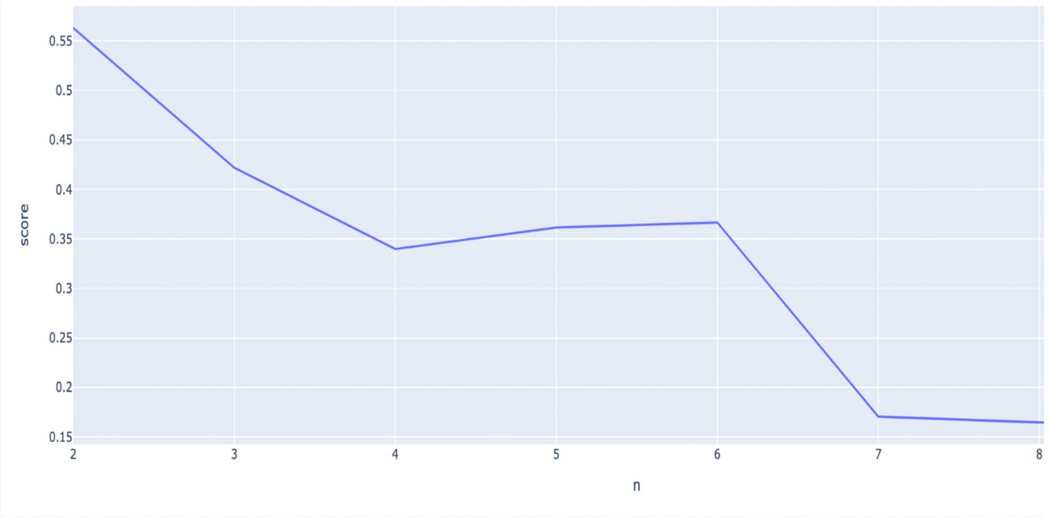
Attributes that show high degree of coreelation:
course_type and run_up_distance = 0.43
purse and race_type = 0.42

Unsupervised Learning - Clustering

The aim is to find any patterns/understand the racing data more.
K means clustering followed by agglomerative clustering(hierarchical clustering) is implemented.

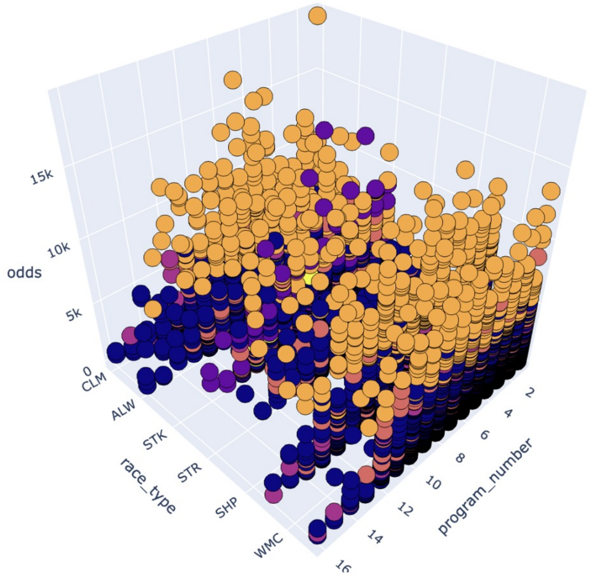


Elbow plot for K means Clustering.
Optimum value, K=6
The graph is a oscillating a bit until 5 clusters.



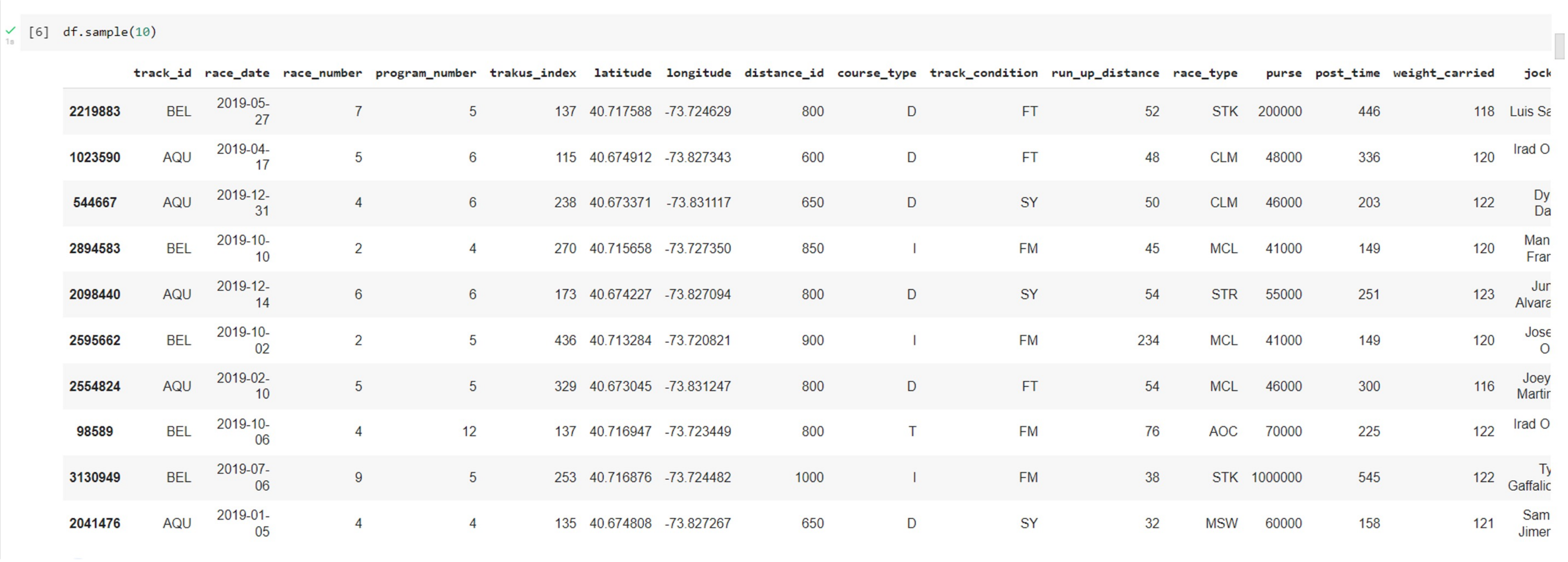
Elbow plot for Agglomerative Clustering.
Optimum value, K=6.
We proceed with this as it has an appropriate knee shape.

3D Overview of clusters formed of data points.
(with respect to race_type and program_number(unique identifier of a horse))



Preprocessing

Before Preprocessing

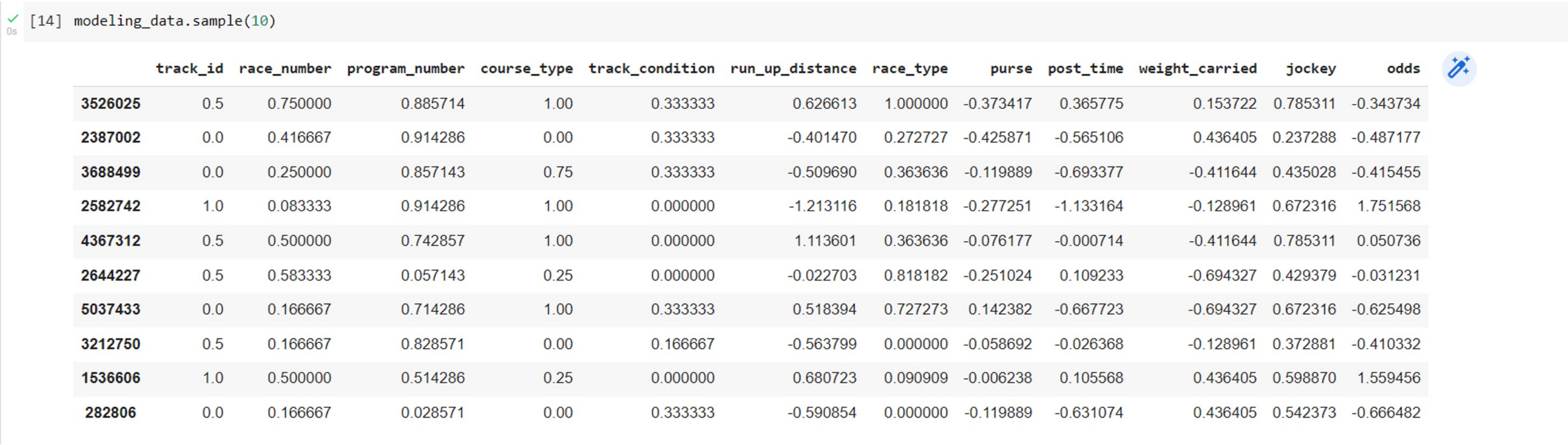


```
[6] df.sample(10)
```

	track_id	race_date	race_number	program_number	trakus_index	latitude	longitude	distance_id	course_type	track_condition	run_up_distance	race_type	purse	post_time	weight_carried	jock
2219883	BEL	2019-05-27	7	5	137	40.717588	-73.724629	800	D	FT	52	STK	200000	446	118	Luis Sa
1023590	AQU	2019-04-17	5	6	115	40.674912	-73.827343	600	D	FT	48	CLM	48000	336	120	Irad O
544667	AQU	2019-12-31	4	6	238	40.673371	-73.831117	650	D	SY	50	CLM	46000	203	122	Dy Da
2894583	BEL	2019-10-10	2	4	270	40.715658	-73.727350	850	I	FM	45	MCL	41000	149	120	Man Frar
2098440	AQU	2019-12-14	6	6	173	40.674227	-73.827094	800	D	SY	54	STR	55000	251	123	Jur Alvara
2595662	BEL	2019-10-02	2	5	436	40.713284	-73.720821	900	I	FM	234	MCL	41000	149	120	Jose O
2554824	AQU	2019-02-10	5	5	329	40.673045	-73.831247	800	D	FT	54	MCL	46000	300	116	Joey Martir
98589	BEL	2019-10-06	4	12	137	40.716947	-73.723449	800	T	FM	76	AOC	70000	225	122	Irad O
3130949	BEL	2019-07-06	9	5	253	40.716876	-73.724482	1000	I	FM	38	STK	1000000	545	122	Ty Gaffalic
2041476	AQU	2019-01-05	4	4	135	40.674808	-73.827267	650	D	SY	32	MSW	60000	158	121	Sam Jimer

Preprocessing

After Preprocessing -
Removed unnecessary columns such as race_id, trackus_index, latitude, longitude, distance_id and duplicate rows. Performed data scaling and standardisation - MinMaxScaler for categorical data and StandardScaler for continuous data.



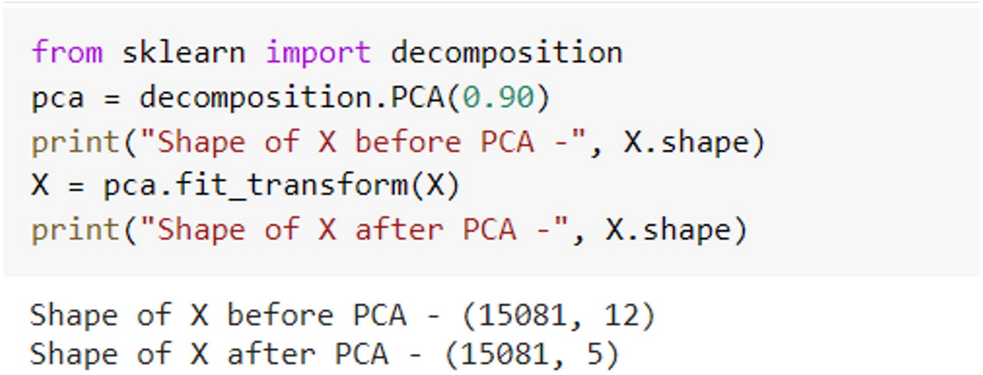
```
[14] modeling_data.sample(10)
```

	track_id	race_number	program_number	course_type	track_condition	run_up_distance	race_type	purse	post_time	weight_carried	jockey	odds
3526025	0.5	0.750000	0.885714	1.00	0.333333	0.626613	1.000000	-0.373417	0.365775	0.153722	0.785311	-0.343734
2387002	0.0	0.416667	0.914286	0.00	0.333333	-0.401470	0.272727	-0.425871	-0.565106	0.436405	0.237288	-0.487177
3688499	0.0	0.250000	0.857143	0.75	0.333333	-0.509690	0.363636	-0.119889	-0.693377	-0.411644	0.435028	-0.415455
2582742	1.0	0.083333	0.914286	1.00	0.000000	-1.213116	0.181818	-0.277251	-1.133164	-0.128961	0.672316	1.751568
4367312	0.5	0.500000	0.742857	1.00	0.000000	1.113601	0.363636	-0.076177	-0.000714	-0.411644	0.785311	0.050736
2644227	0.5	0.583333	0.057143	0.25	0.000000	-0.022703	0.818182	-0.251024	0.109233	-0.694327	0.429379	-0.031231
5037433	0.0	0.166667	0.714286	1.00	0.333333	0.518394	0.727273	0.142382	-0.667723	-0.694327	0.672316	-0.625498
3212750	0.5	0.166667	0.828571	0.00	0.166667	-0.563799	0.000000	-0.058692	-0.026368	-0.128961	0.372881	-0.410332
1536606	1.0	0.500000	0.514286	0.25	0.000000	0.680723	0.090909	-0.006238	0.105568	0.436405	0.598870	1.559456
282806	0.0	0.166667	0.028571	0.00	0.333333	-0.590854	0.000000	-0.119889	-0.631074	0.436405	0.542373	-0.666482

PCA

Performed PCA on the modeling_dataset -

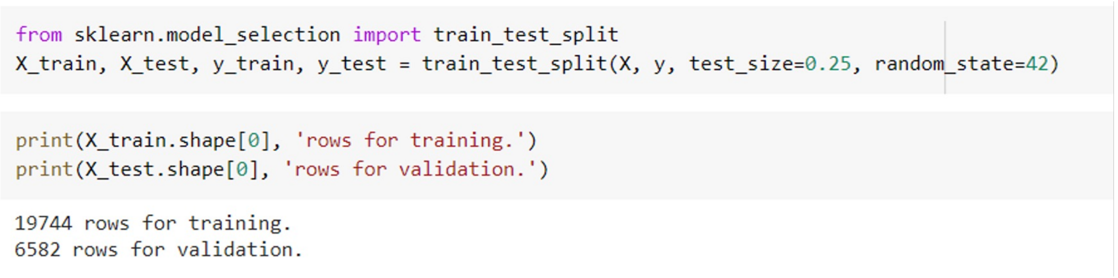
- Principal component analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.
- Smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.



```python
from sklearn import decomposition
pca = decomposition.PCA(0.90)
print("Shape of X before PCA -", X.shape)
X = pca.fit_transform(X)
print("Shape of X after PCA -", X.shape)
```

```
Shape of X before PCA - (15081, 12)
Shape of X after PCA - (15081, 5)
```

Modeling and Inference



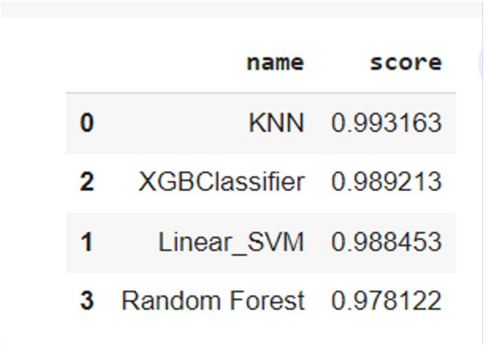
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

```
print(X_train.shape[0], 'rows for training.')
print(X_test.shape[0], 'rows for validation.')
```

```
19744 rows for training.
6582 rows for validation.
```

Split the modeling dataset into training (19744 rows) and test(6582 rows)

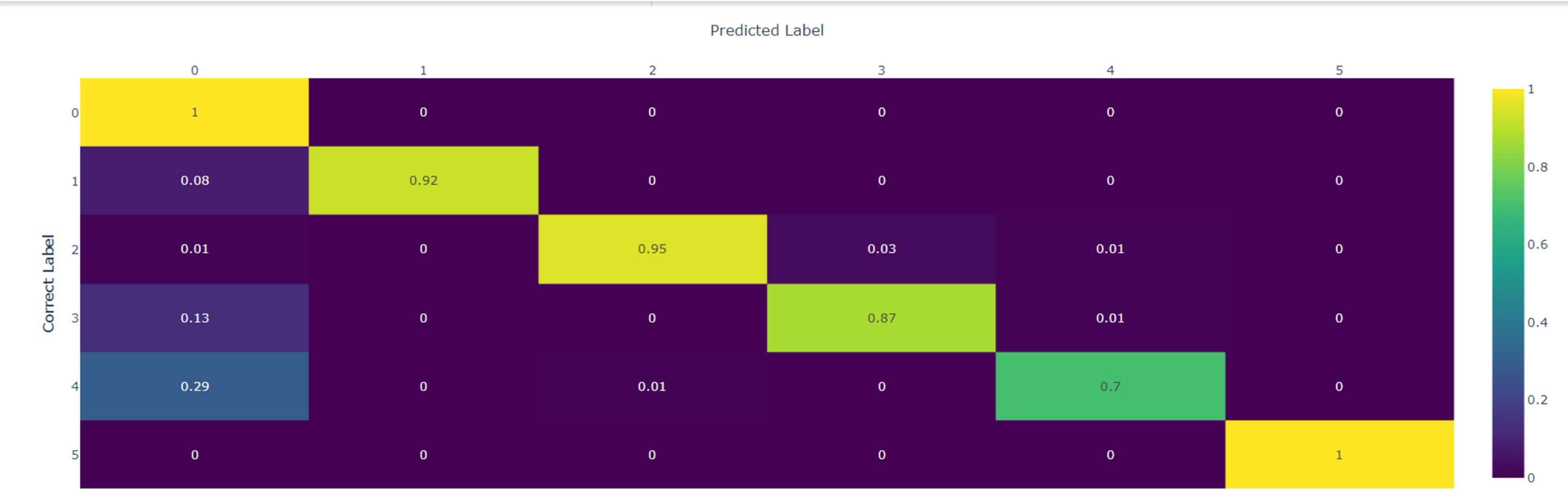
Trained models such as KNN, XGBClassifier, Linear_SVM and Random Forest on the training dataset. RandomForestClassifier, etc. computes the accuracy of the model on the testing data (accuracy is #correct_preds / #all_preds) after being trained on the training data. From this table, it is evident that all performed well and KNN performed best amongst the given models.



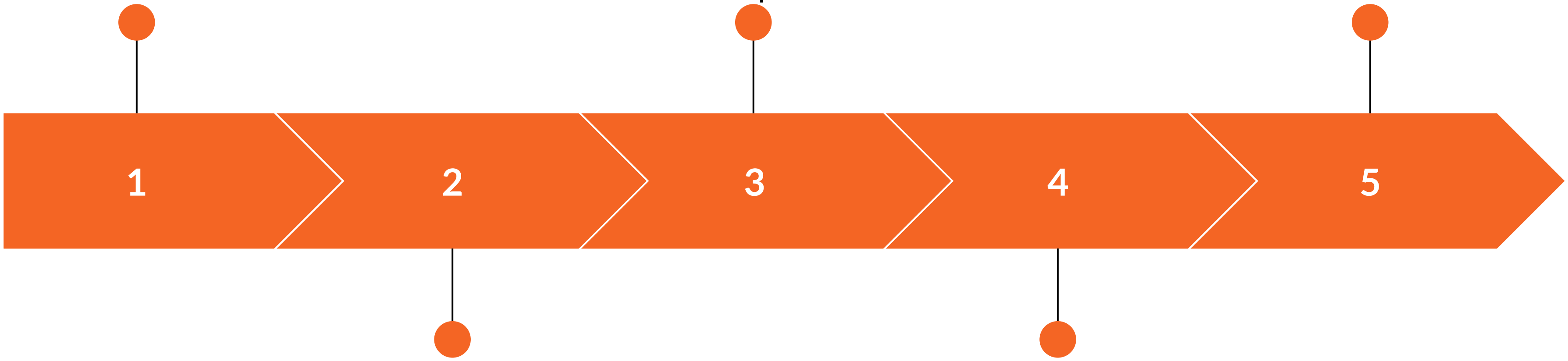
	name	score
0	KNN	0.993163
2	XGBClassifier	0.989213
1	Linear_SVM	0.988453
3	Random Forest	0.978122

Confusion Matrix

Given below is the confusion matrix between the actual and predicted odds by our classifier.



Correct Label \ Predicted Label	0	1	2	3	4	5
0	1	0	0	0	0	0
1	0.08	0.92	0	0	0	0
2	0.01	0	0.95	0.03	0.01	0
3	0.13	0	0	0.87	0.01	0
4	0.29	0	0.01	0	0.7	0
5	0	0	0	0	0	1



1

EDA

2

Unsupervised Learning(k means & agglomerative clustering)

3

Preprocessing

(label encoding, scaling, feature selection,dimensionality reduction train-test split)

4

PCA and Modelling

5

Inference

Next Steps

1. Analysis Of Models
2. Conclusion

Thank You!