Regression Project

Doron T.test

Guy K.nn

The Data

The used data is Natality records from 1969 - 2008

- 116774 rows × 29 columns
- Grain = Natality record
- Some of the columns contain data only on specified period.
- Some of data hadn’t a clear definitions so we made assumptions

source_year	INTEGER	REQUIRED	Four-digit year of the birth. Example: 1975.
year	INTEGER	NULLABLE	Four-digit year of the birth. Example: 1975.
month	INTEGER	NULLABLE	Month index of the date of birth, where 1=January.
day	INTEGER	NULLABLE	Day of birth, starting from 1.
wday	INTEGER	NULLABLE	Day of the week, where 1 is Sunday and 7 is Saturday.
state	STRING	NULLABLE	The two character postal code for the state. Entries after 2004 do not include this value.
is_male	BOOLEAN	REQUIRED	TRUE if the child is male, FALSE if female.
child_race	INTEGER	NULLABLE	The race of the child. One of the following numbers: 1 - White 2 - Black 3 - American Indian 4 - Chinese 5 - Japanese 6 - Hawaiian 7 - Filipino 9 - Unknown/Other 18 - Asian Indian 28 - Korean 39 - Samoan 48 - Vietnamese
weight_pounds	FLOAT	NULLABLE	Weight of the child, in pounds.
plurality	INTEGER	NULLABLE	How many children were born as a result of this pregnancy. twins=2, triplets=3, and so on.
apgar_1min	INTEGER	NULLABLE	Apgar scores measure the health of a newborn child on a scale from 0-10. Value after 1 minute. Available from 1978-2002.
apgar_5min	INTEGER	NULLABLE	Apgar scores measure the health of a newborn child on a scale from 0-10. Value after 5 minutes. Available from 1978-2002.
mother_residence_state	STRING	NULLABLE	The two-letter postal code of the mother's state of residence when the child was born.
mother_race	INTEGER	NULLABLE	Race of the mother. Same values as child_race.
mother_age	INTEGER	NULLABLE	Reported age of the mother when giving birth.
gestation_weeks	INTEGER	NULLABLE	The number of weeks of the pregnancy.
lmp	STRING	NULLABLE	Date of the last menstrual period in the format MMDDYYYY. Unknown values are recorded as "99" or "9999".
mother_married	BOOLEAN	NULLABLE	True if the mother was married when she gave birth.
mother_birth_state	STRING	NULLABLE	The two-letter postal code of the mother's birth state.
cigarette_use	BOOLEAN	NULLABLE	True if the mother smoked cigarettes. Available starting 2003.
cigarettes_per_day	INTEGER	NULLABLE	Number of cigarettes smoked by the mother per day. Available starting 2003.
alcohol_use	BOOLEAN	NULLABLE	True if the mother used alcohol. Available starting 1989.
drinks_per_week	INTEGER	NULLABLE	Number of drinks per week consumed by the mother. Available starting 1989.
weight_gain_pounds	INTEGER	NULLABLE	Number of pounds gained by the mother during pregnancy.
born_alive_alive	INTEGER	NULLABLE	Number of children previously born to the mother who are now living.
born_alive_dead	INTEGER	NULLABLE	Number of children previously born to the mother who are now dead.
born_dead	INTEGER	NULLABLE	Number of children who were born dead (i.e. miscarriages)
ever_born	INTEGER	NULLABLE	Total number of children to whom the woman has ever given birth (includes the current birth).
father_race	INTEGER	NULLABLE	Race of the father. Same values as child_race.
father_age	INTEGER	NULLABLE	Age of the father when the child was born.
record_weight	INTEGER	NULLABLE	1 or 2, where 1 is a row from a full-reporting area, and 2 is a row from a 50% sample area.

EDA & Data cleaning

Remove dup. columns & Rows , and unneeded columns:

```
# data cleanning
df[df['source_year']!=df['year']]
# Year & Source_Year are redundant and one of them can be reduced
df.drop('source_year', axis='columns' ,inplace=True)
#Not required
df.drop(columns = ['wday'],axis='columns', inplace=True)
```

```
In [11]: #Check for Duplicated
         df[df.duplicated()]
```

Out[11]:

year	month	wday	is_male	weight_pounds	plurality	apg

0 rows × 28 columns

Create derived columns

```
df['weight_Kg']=df['weight_pounds']*0.45359237
df['weight_gain_kg']=df['weight_gain_pounds']*0.45359237
```

EDA & Data cleaning

Data Cleaning

```
# df.drop(['Unnamed: 0_x','Unnamed: 0_y'], axis='columns' ,inplace=True)
df.describe()
```

ce	apgar_1min	mother_race	born_alive_alive	born_alive_dead	born_dead	father_race	cigarettes_per_day	drinks_per_week	weight_gain_pounds
00	79631.000000	108328.000000	105375.000000	105274.000000	105159.000000	108328.000000	1136.000000	1567.000000	67736.000000
91	52.522648	1.689692	1.247848	0.566921	0.773086	4.289814	16.464789	13.722399	45.019015
39	45.474914	4.503685	3.921974	5.892360	5.580808	13.836230	25.067778	33.150497	29.741982
00	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000
00	8.000000	1.000000	0.000000	0.000000	0.000000	1.000000	5.000000	0.000000	25.000000
00	9.000000	1.000000	1.000000	0.000000	0.000000	1.000000	10.000000	0.000000	34.000000
00	99.000000	1.000000	2.000000	0.000000	0.000000	2.000000	15.000000	2.000000	52.000000
00	99.000000	78.000000	77.000000	77.000000	77.000000	99.000000	99.000000	99.000000	99.000000

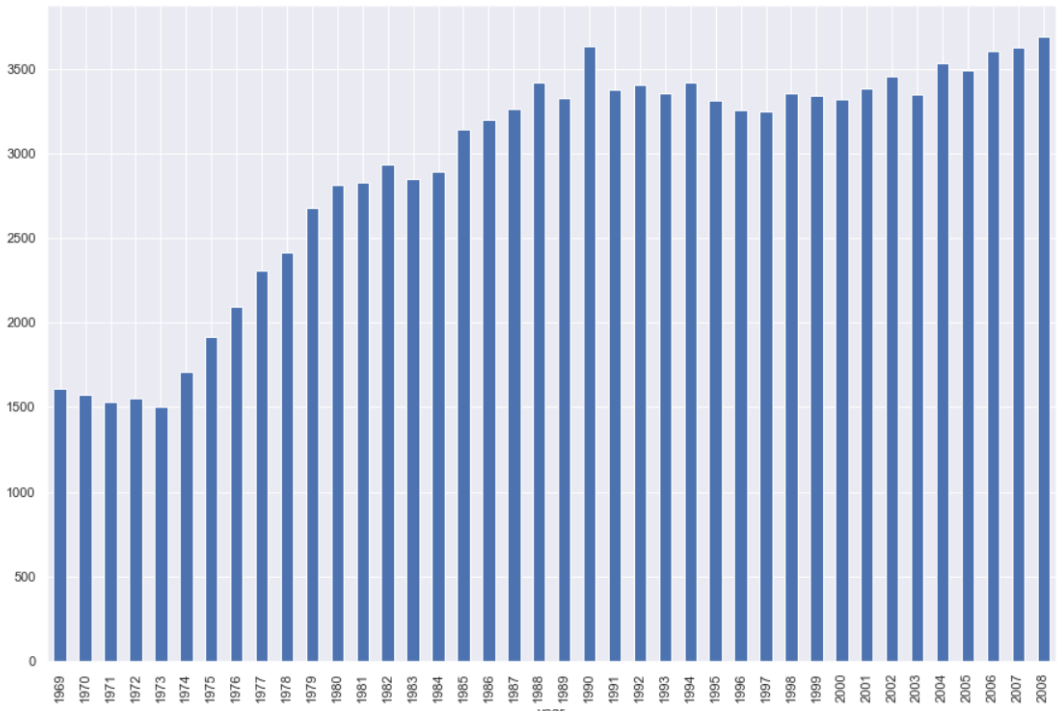
```
a = np.array(df['apgar_5min'].values.tolist())
print (a)
df['apgar_5min'] = np.where(a > 10, np.nan, a).tolist()

a = np.array(df['apgar_1min'].values.tolist())
print (a)
df['apgar_1min'] = np.where(a > 10, np.nan, a).tolist()

df.max()
```

```
# Replace Bool with 1/0
df_r["cigarette_use"] = (df['cigarette_use'].fillna(0).astype(int).astype(object).where(df['cigarette_use'].notnull()) )
df_r["alcohol_use"] = (df['alcohol_use'].fillna(0).astype(int).astype(object).where(df['alcohol_use'].notnull()) )
df_r["is_male"] = (df['is_male'].fillna(0).astype(int).astype(object).where(df['is_male'].notnull()) )
```

EDA & Data cleaning

Data exploring

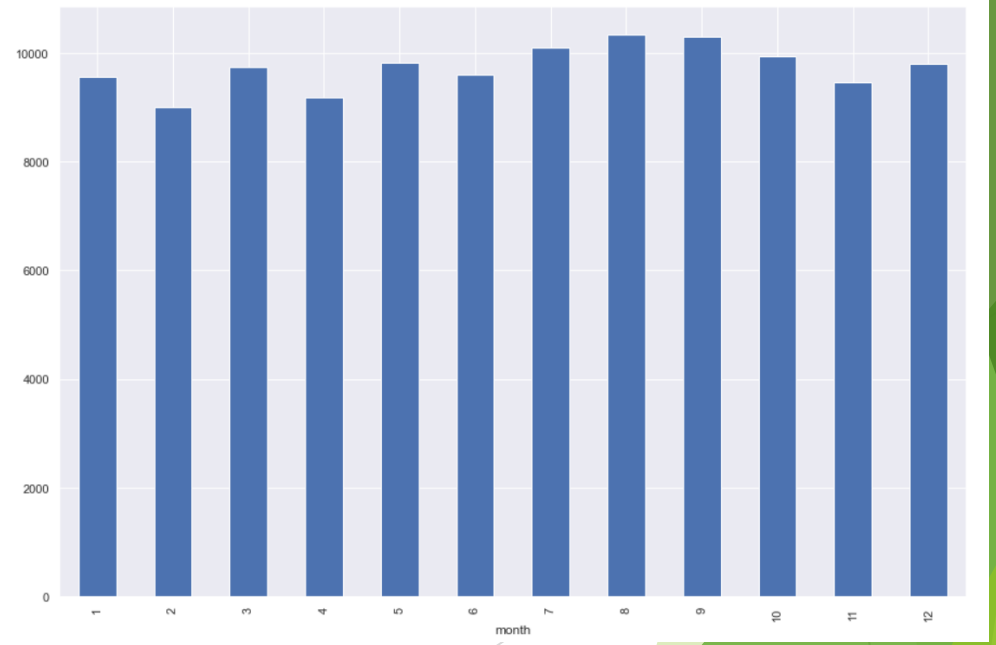
Samples distribution per year



Check if there are less samples in certain months

```
# Samples distribution per month - Check if there are less samples in certain months

month_dist = df.groupby('month').size()
month_dist.plot(kind='bar')
```

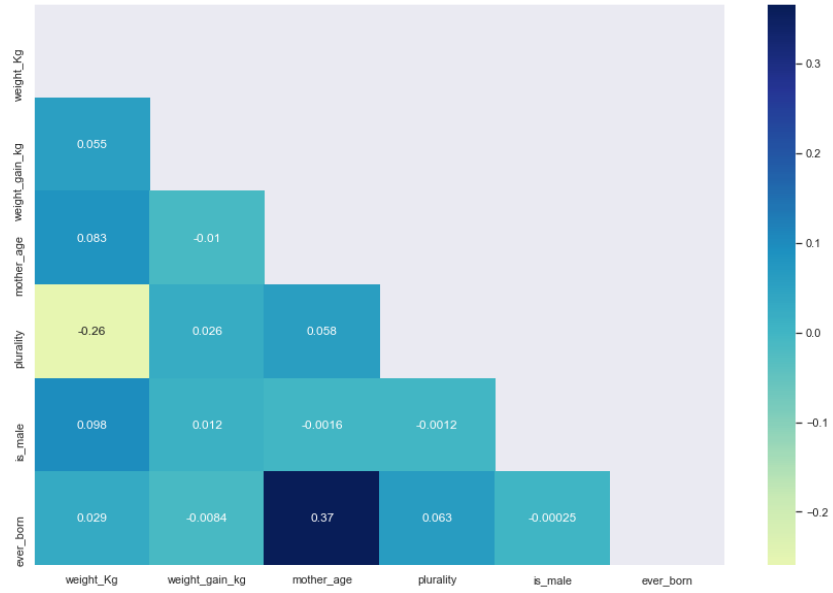
```
<AxesSubplot:xlabel='month'>
```

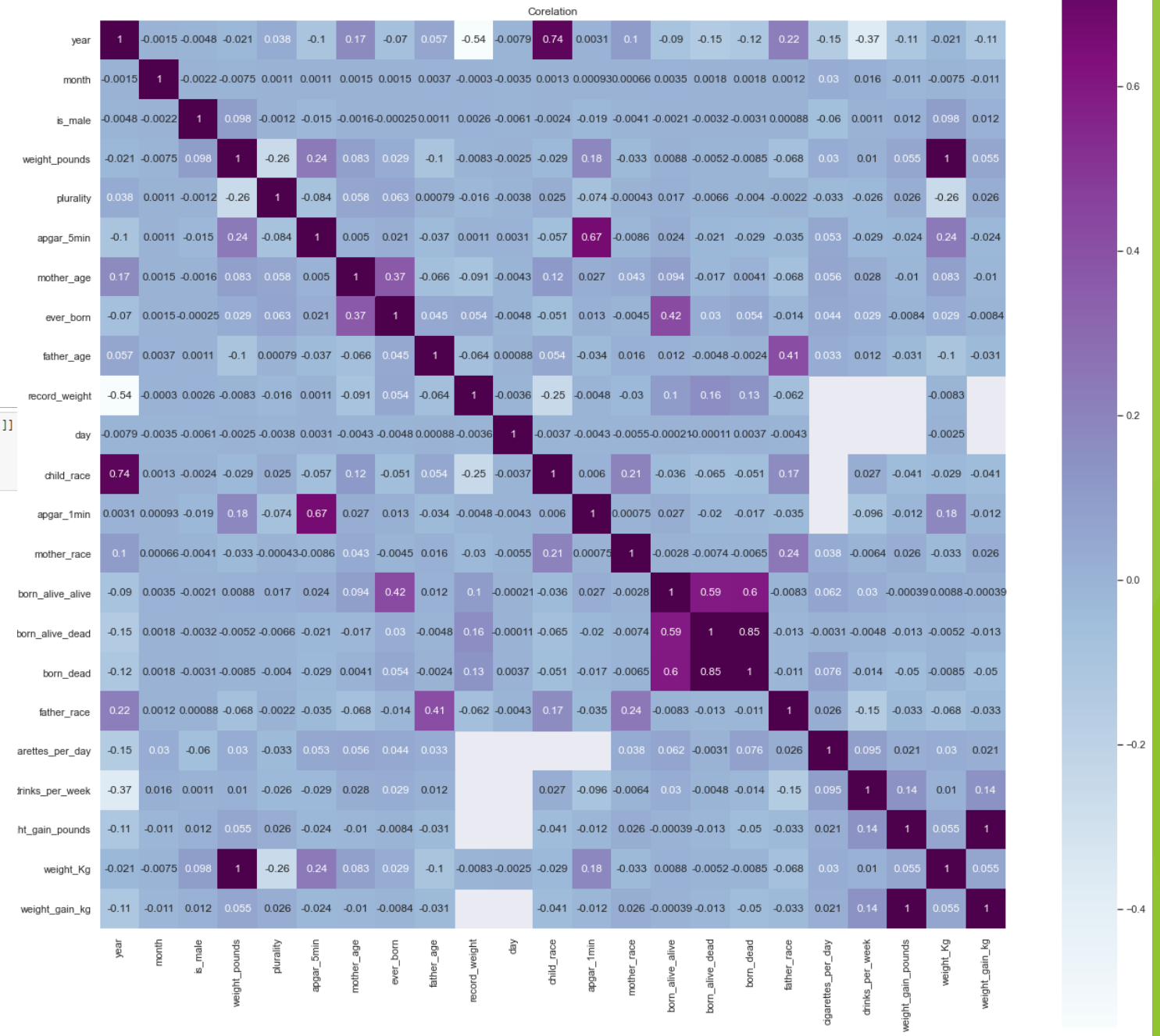


Data correlation

We couldn't find any meaningful correlations (Except expected) between features.

```
df_r=df[['Sex','weight_Kg','weight_gain_kg','mother_age','plurality','alcohol_use','cigarette_use','is_male','ever_born']]
matrix = np.triu(df_r.corr())
fig = plt.figure()
sns.heatmap(df_r.corr(),annot = True,mask=matrix , cmap= 'YlGnBu', center = 0)
plt.show()
```

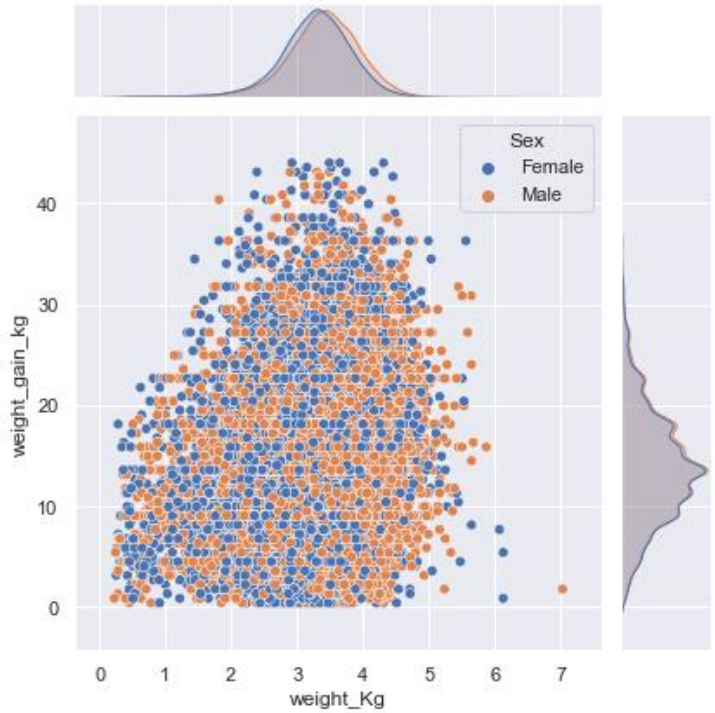




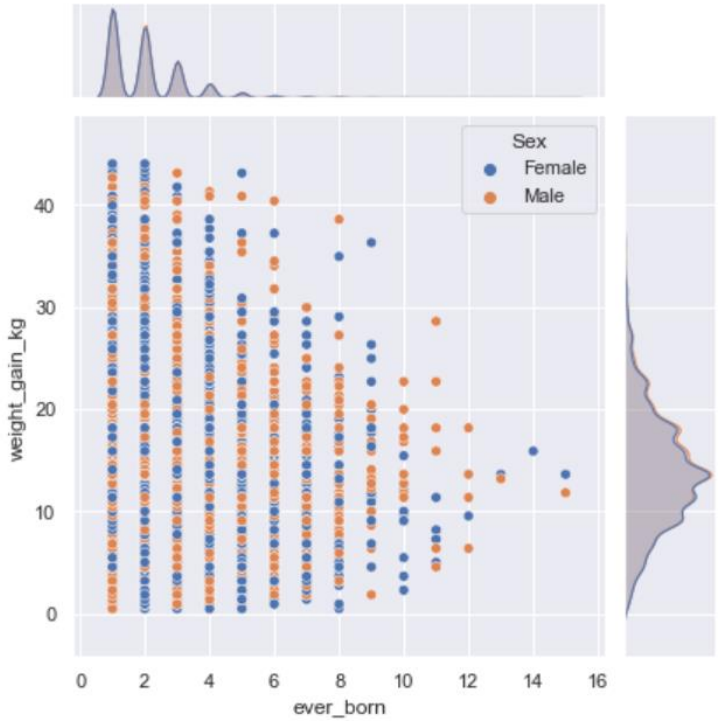
EDA & Data cleaning

Assumptions

```
#weight_gain_kg = 44.905645 is an Outlier an is removed.
df_r_weight=df_r[df_r['weight_gain_kg']<44]
# df.groupby('weight_gain_kg').size().plot(kind='bar')

sns.jointplot(x='weight_Kg', y='weight_gain_kg', data=df_r_weight, hue='Sex');
```




```
sns.jointplot(x='ever_born', y='weight_gain_kg', data=df_r_weight, hue='Sex');
```

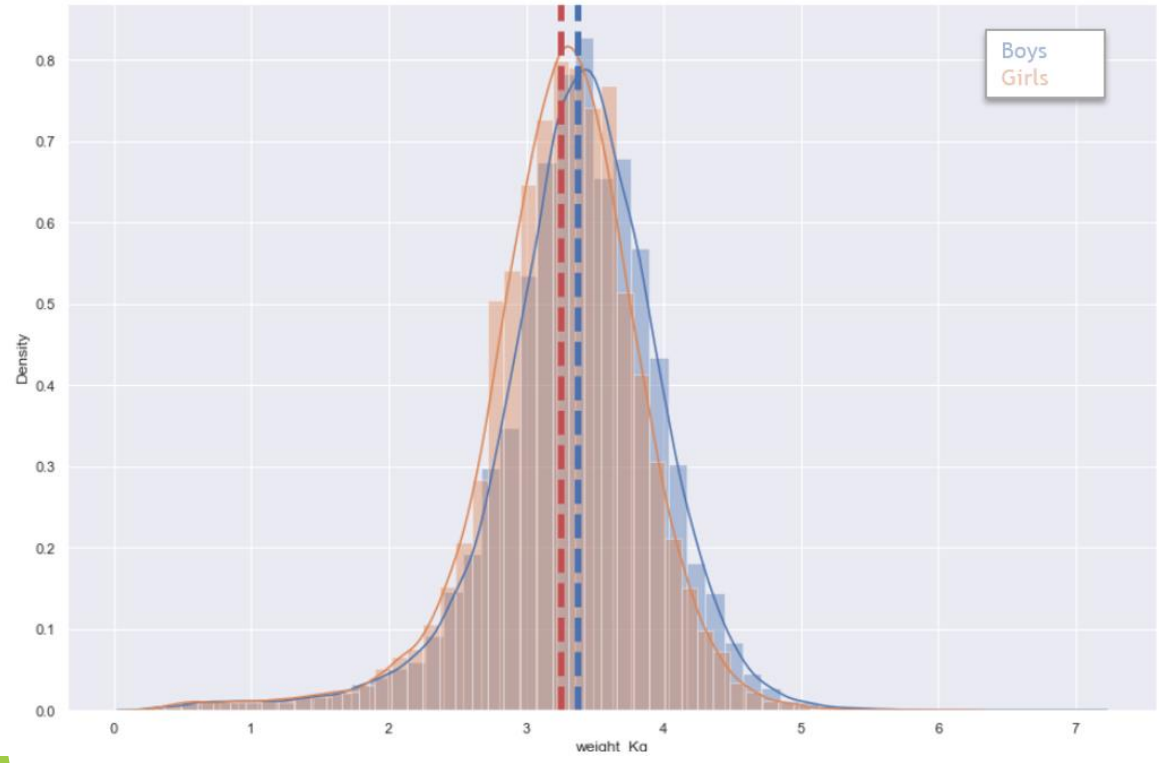


EDA & Data cleaning

T-Test – Is there a difference between boys and girls

```
plt.figure()
ax1=sns.distplot(boys)
ax2=sns.distplot(girls)
plt.axvline(np.mean(boys), color='b' , linestyle='dashed', linewidth=5)
plt.axvline(np.mean(girls), color='r' , linestyle='dashed', linewidth=5)
```

```
<matplotlib.lines.Line2D at 0x27e3df57e80>
```




```
# H0: There is no differance between boys and girls weight.
# Alpha = 0.05

boys=df[df['is_male']==True].weight_Kg
girls=df[df['is_male']!=True].weight_Kg

#T-test
ttest,pval = ttest_ind(boys,girls, equal_var=False)
print('H0 : There is no differance between boys and girls weight\n ')
print(f'The t-test is: {ttest} the pval is: {pval}\n')
print( 'alpha was 0.05, so we will reject H0 \n\n ')
```

```
H0 : There is no differance between boys and girls weight

The t-test is: 33.7207401545592 the pval is: 4.5148351045504876e-248

alpha was 0.05, so we will reject H0
```

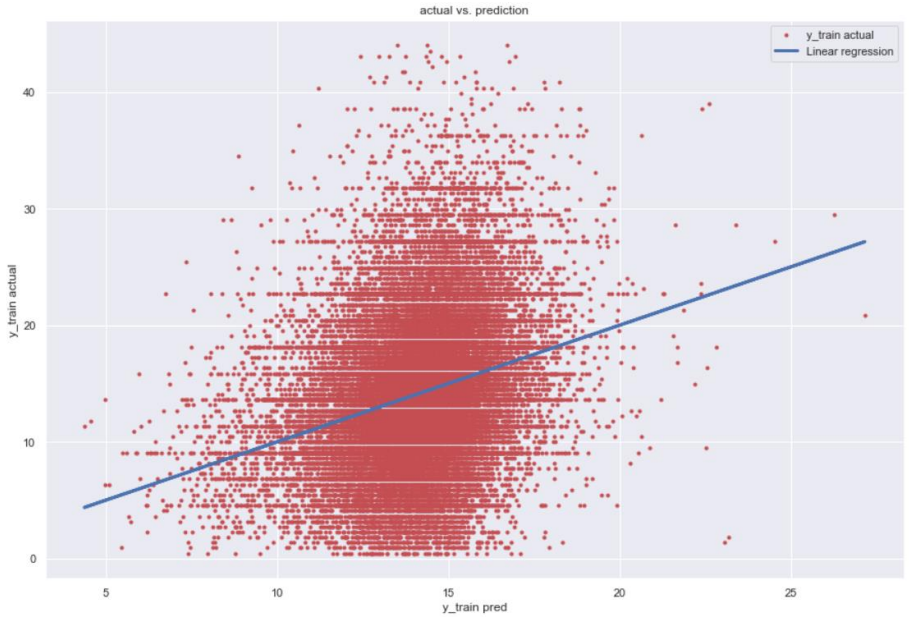
EDA summary

- We couldn’t find any significant feature that affects the child weight.
- Based on our T-test there is a statistically significance difference between boys & girls weight.
- In our next step we will try to predict the Mother weight gain based on different features.

Regression - LR

```
#   Column
---  ------
0   weight_Kg
1   mother_age
2   plurality
3   is_male
4   alcohol_use
5   cigarette_use
6   ever_born
```

R^2 train Data = 0.07
RMSE train Data = 5.66

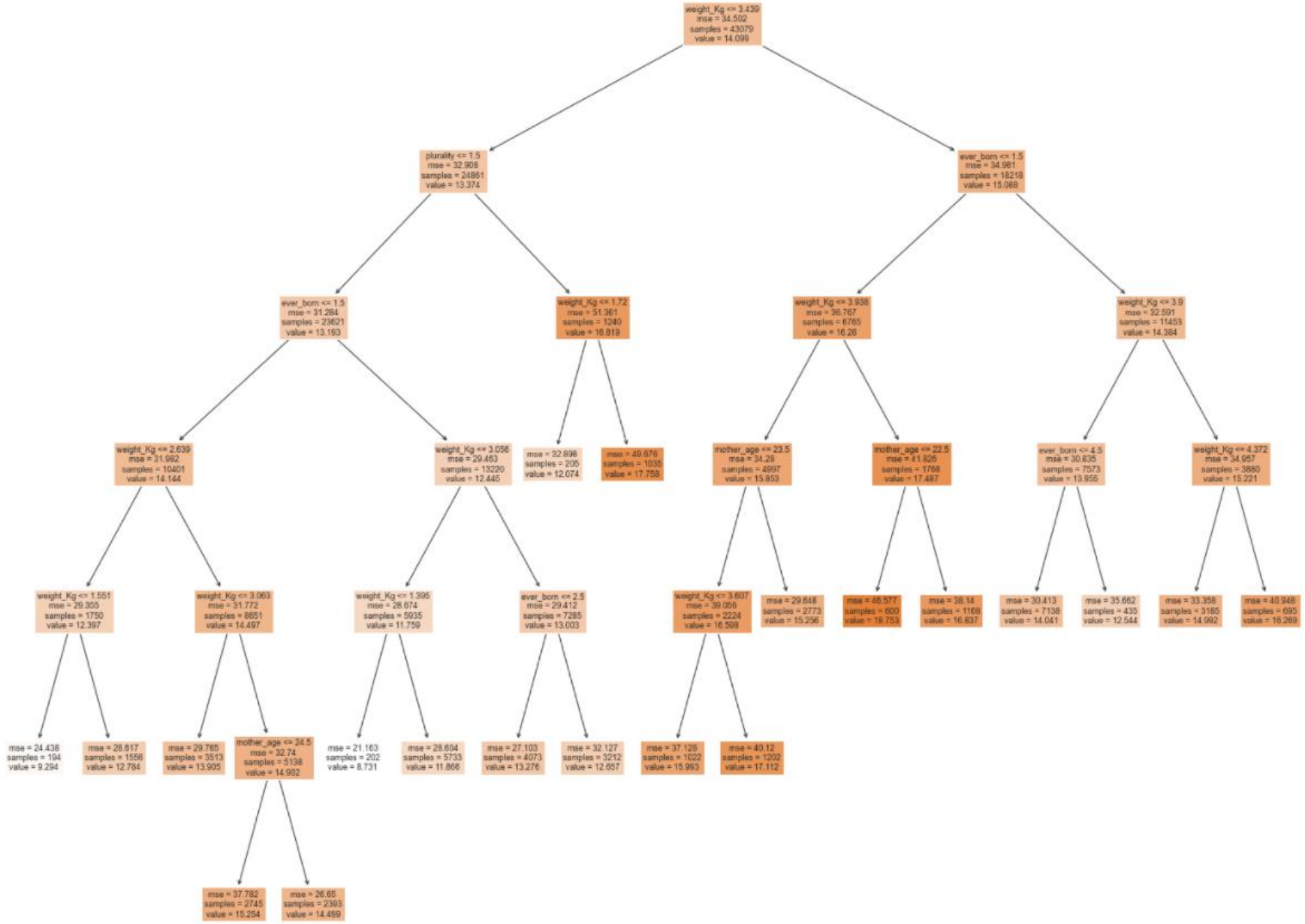


R^2 Test Data = 0.07
RMSE Test Data = 5.65

weight_gain_Kg =
3.466+2.303*weight_Kg-0.041*mother_age+5.206*plurality-0.072*is_male-0.008*alcohol_use+0.915*cigarette_use-0.607*ever_born

*Over fitting?

Regression –Tree



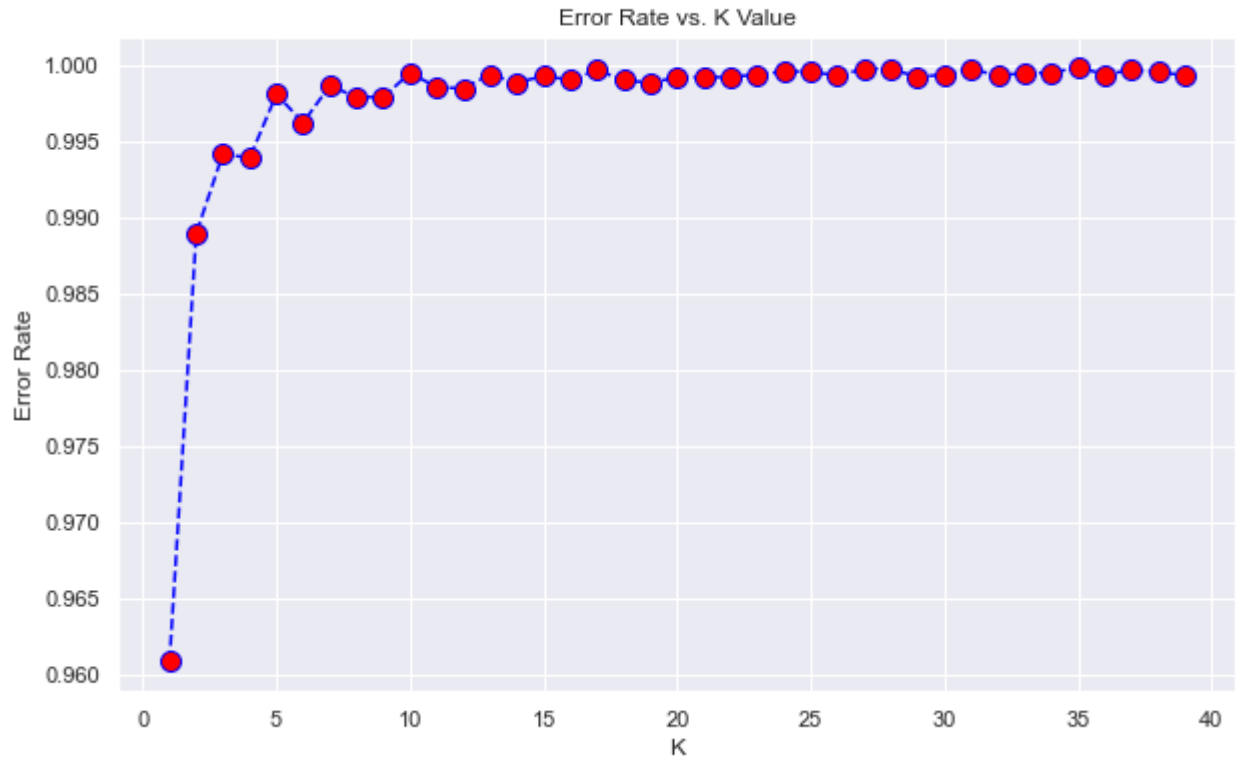

```
Train TREE RMSE= 5.64
Test TREE RMSE= 5.64
```

```
#max_leaf_nodes = 10 --> RMSE = 5.67
#max_leaf_nodes = 20 --> RMSE = 5.64
#max_leaf_nodes = 20 --> RMSE = 5.63
#max_leaf_nodes = 40 --> RMSE = 5.62
```

Regression –KNN

Finding optimal K

```
Minimum error:- 0.9609099350046425 at K = 0
```




```
KNN RMSE TRAIN= 4.88
KNN RMSE TEST= 6.87
```