IME672
GROUP PROJECT
TITLE: Predict price of flight ticket

Members(Group 8):

1. Patil Saurabh Narendra 200679
2. Navneet Singh 200625
3. Mrinal Kumar 200600
4. Mohd Rafique Khan 200592
5. Himanshu Beniwal 200441

Problem Description: predict price for flight tickets using data of Air Tickets Between Shanghai(SHA) and Beijing(PEK). (Link to problem)

Data Understanding:

We got 14 data columns out of which **price** data column is the one we're trying to predict.

Defining the columns-

1. **ID**: integer data type describing a person's identity
2. **flightNumber**: object data type describing which flight they should take
3. **craftTypeCode**: object data type describing ICAO aircraft type designators
4. **depAirport**: object data type mentioning departure airport
5. **traAirport**: object data type mentioning an airport where the aircraft has made a stopover
6. **arrAirport**: object data type mentioning arrival airport
7. **departureDate**: object data type mentioning departure date
8. **arrivalDate**: object data type mentioning arrival date
9. **cabinClass**: object data type describing which cabin class people chose on the flight
10. **priceClass**: object data type describing which price class people chose on the flight
11. **price**: integer data type describing the price of the flight
12. **rate:** float data type describing the discount rate
13. **createDate:** object data type describing ticket booking date
14. **dateDifference:** int data type describing date difference between the createdate and the actual departure date

Data Preprocessing:

Data format: csv

The dataset has been taken from Kaggle.

Showing 1st 10 rows of the dataset-

	ID	flightNumber	craftTypeCode	depAirport	traAirport	arrAirport	departureDate	arrivalDate	cabinClass	priceClass	price	rate	createDate	dateDifference
0	14393	HO1252	320	PEK	NaN	SHA	2019-01-04 06:35:00	2019-01-04 08:55:00	C	C	1860	1.00	2019-01-03 14:26:15	1
1	14409	MU5138	33L	PEK	NaN	SHA	2019-01-04 07:00:00	2019-01-04 09:15:00	C	I	1640	0.31	2019-01-03 14:26:15	1
2	14415	MU5138	33L	PEK	NaN	SHA	2019-01-04 07:00:00	2019-01-04 09:15:00	C	J	5360	1.00	2019-01-03 14:26:15	1
3	14429	HU7605	350	PEK	NaN	SHA	2019-01-04 07:20:00	2019-01-04 09:35:00	C	I	1635	0.29	2019-01-03 14:26:15	1
4	14431	HU7605	350	PEK	NaN	SHA	2019-01-04 07:20:00	2019-01-04 09:35:00	C	I	1640	0.29	2019-01-03 14:26:15	1
5	14433	HU7605	350	PEK	NaN	SHA	2019-01-04 07:20:00	2019-01-04 09:35:00	C	C	5660	1.00	2019-01-03 14:26:15	1
6	14445	CA1831	773	PEK	NaN	SHA	2019-01-04 07:30:00	2019-01-04 09:35:00	C	R	1860	0.34	2019-01-03 14:26:15	1
7	14451	CA1831	773	PEK	NaN	SHA	2019-01-04 07:30:00	2019-01-04 09:35:00	C	J	5530	1.00	2019-01-03 14:26:15	1
8	14488	MU5102	333	PEK	NaN	SHA	2019-01-04 08:00:00	2019-01-04 10:15:00	C	I	1800	0.34	2019-01-03 14:26:15	1
9	14491	MU5102	333	PEK	NaN	SHA	2019-01-04 08:00:00	2019-01-04 10:15:00	C	J	5360	1.00	2019-01-03 14:26:15	1

```
RangeIndex: 300336 entries, 0 to 300335
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   ID              300336 non-null  int64
 1   flightNumber    300336 non-null  object
 2   craftTypeCode   300336 non-null  object
 3   depAirport      300336 non-null  object
 4   traAirport      5749 non-null    object
 5   arrAirport      300336 non-null  object
 6   departureDate   300336 non-null  object
 7   arrivalDate     300336 non-null  object
 8   cabinClass      300336 non-null  object
 9   priceClass      300336 non-null  object
 10  price           300336 non-null  int64
 11  rate            300336 non-null  float64
 12  createDate      300336 non-null  object
 13  dateDifference  300336 non-null  int64
dtypes: float64(1), int64(3), object(10)
```

Deriving information from the data-

- So there are total 300336 entries
- traAirport only has 5749 non-zero entries
- Dataset consists of 3 data types

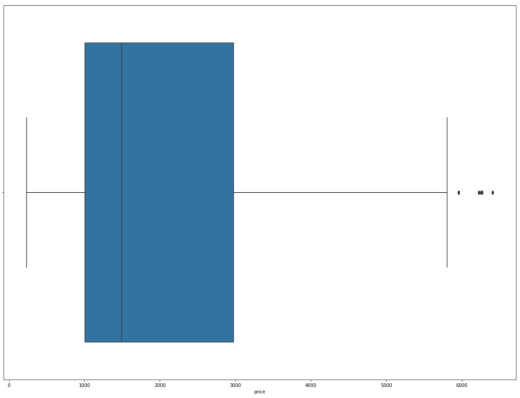
Describing the dataset-

	ID	price	rate	dateDifference
count	3.003360e+05	300336.000000	300336.000000	300336.000000
mean	3.785370e+07	2209.684696	0.764667	6.682762
std	2.320571e+07	1755.003424	0.251366	7.501340
min	1.438400e+04	235.000000	0.000000	1.000000
25%	1.770736e+07	1005.000000	0.530000	2.000000
50%	3.718240e+07	1490.000000	0.860000	5.000000
75%	5.705183e+07	2980.000000	1.000000	7.000000
max	8.016500e+07	6410.000000	1.000000	30.000000

Now We will drop some useless columns

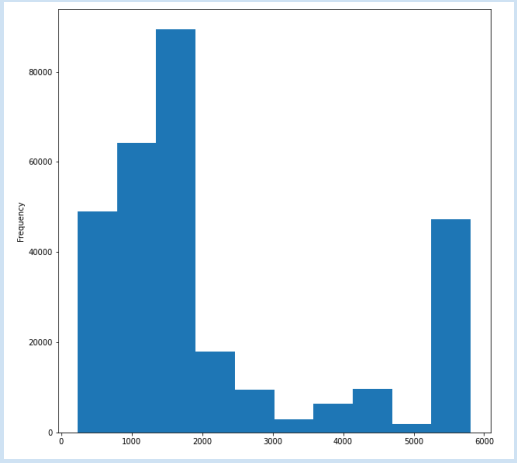
- **ID** (since its a random integer with no significance to price)
- **depAirport & arrAirport** (as they have 1, 1 unique values respectively)
- **createDate** (as dateDifference serves the same purpose)

- **departureDate & arrivalDate** (we will make 2 different columns on **depDay** and **daysaway**(in sec.) instead.

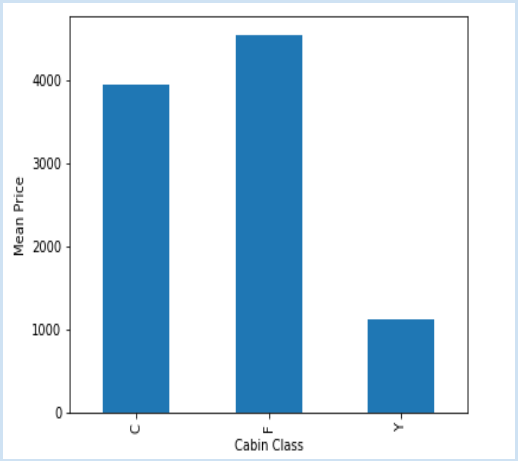


Boxplot for Price label

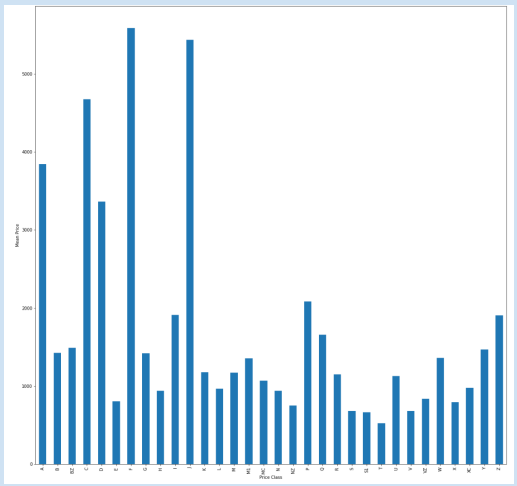
Checking for Outliers in the data-
There are 2205 outliers



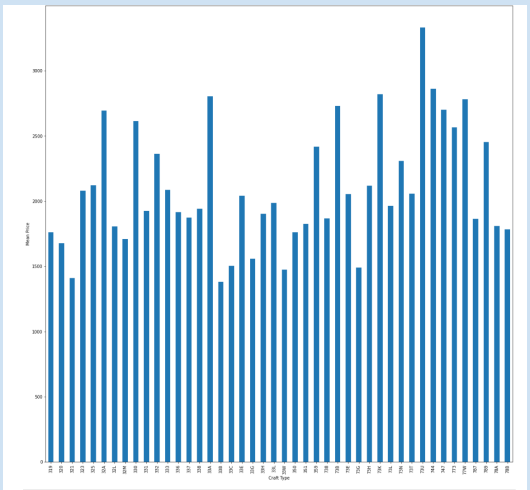
Histogram for price



Bar Plot for Mean Price vs CabinClass



Bar Plot for Mean Price vs PriceClass



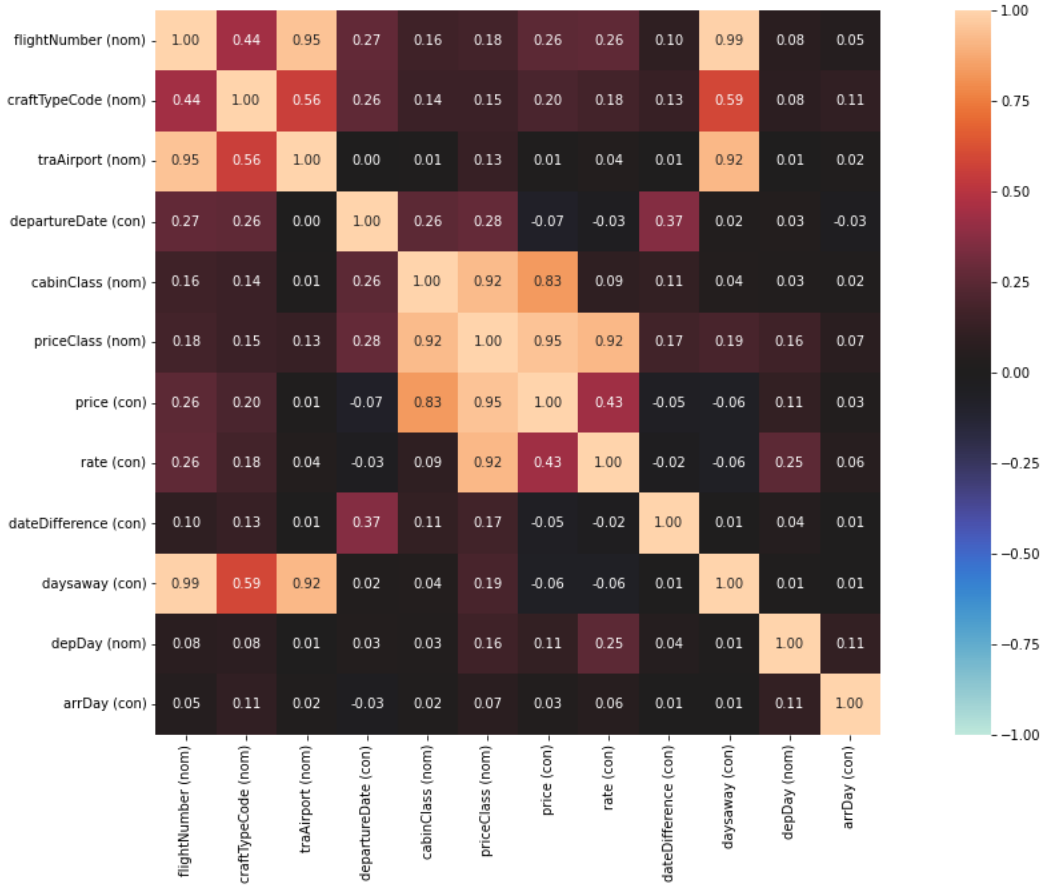
Bar Plot for Mean Price vs craftType

	price	rate	dateDifference	daysaway	arrDay
price	1.000000	0.430178	-0.052950	-0.057445	0.025186
rate	0.430178	1.000000	-0.021908	-0.060896	0.060926
dateDifference	-0.052950	-0.021908	1.000000	0.007318	0.005487
daysaway	-0.057445	-0.060896	0.007318	1.000000	0.005867
arrDay	0.025186	0.060926	0.005487	0.005867	1.000000

Plotting the Heatmap-

After looking at the heatmap, we can conclude the following.

- **cabinClass, priceClass, rate** are the only 3 attributes which have significant corrrelation with **price**(target attribute)**.**
- All other attributes arenot significant, hence, we will drop all the other columns.
- Now we derive different categories based on the unique values in cabinClass & priceClass by implementing One hot encoding.



New Dataframe-

The new dataset obtained after applying OneHotEncoding on CabinClass and PriceClass.

	price	rate	0	1	2	3	4	5	6	7	...	25	26	27	28	29	30	31	32	33	34
0	1860	1.00	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1640	0.31	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5360	1.00	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1635	0.29	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1640	0.29	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	5660	1.00	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1860	0.34	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	5530	1.00	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	1800	0.34	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	5360	1.00	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Model Building:

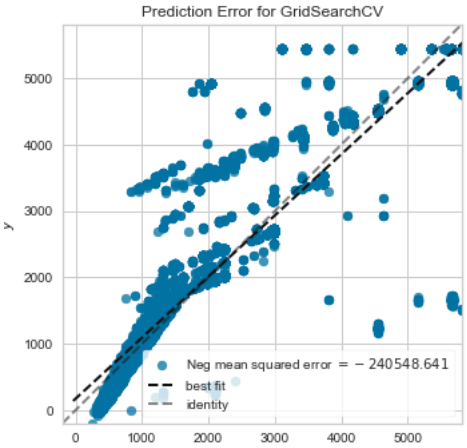
We build the following models for training the data:

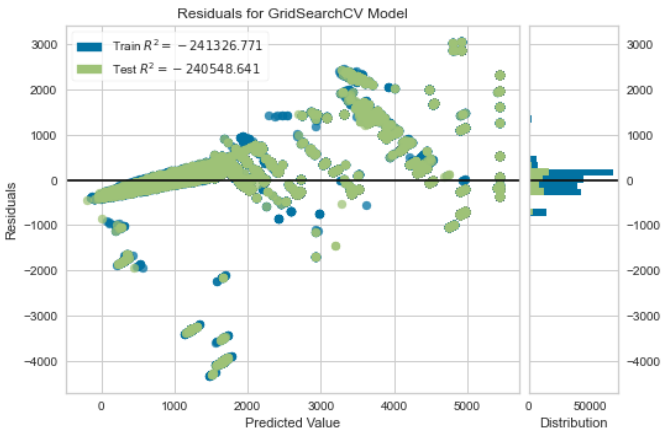
1. Ridge Regressor(Grid search based)
2. Lasso Regressor
3. Random Forest Regressor
4. Bayesian Regressor
5. Stochastic Gradient Descent Regressor
6. Artificial Neural Network

Model Evaluation:

The prediction error and residual plots for various models are:

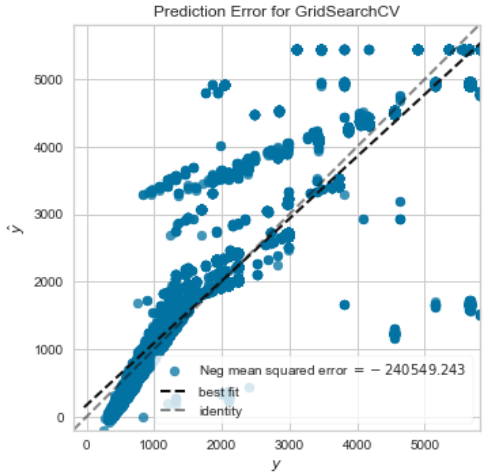
1. Ridge regressor:

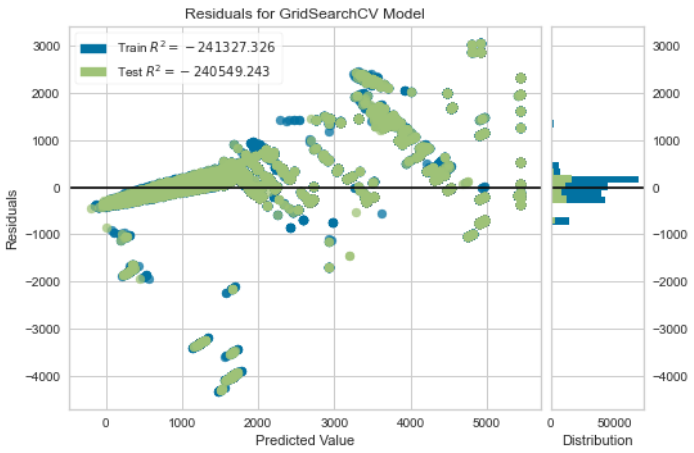




RMSE Error on test dataset: 491.25

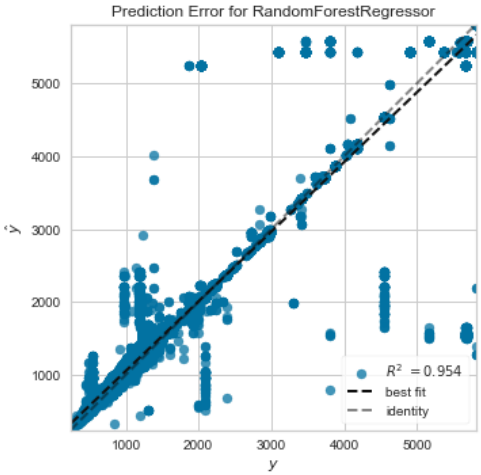
2. Lasso Regressor:

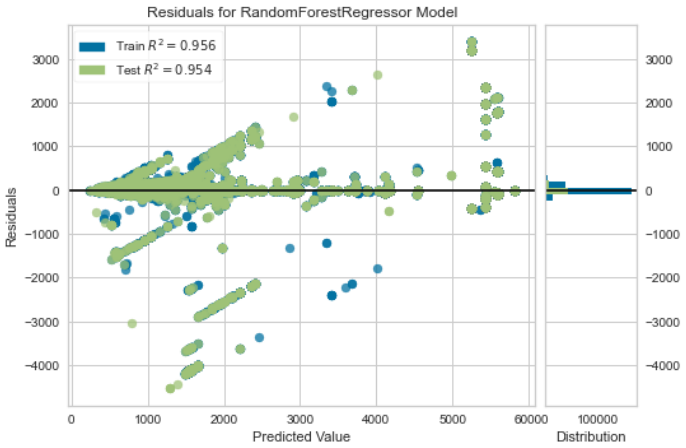




RMSE Error on test dataset: 491.251

3. Random Forrest Regressor:



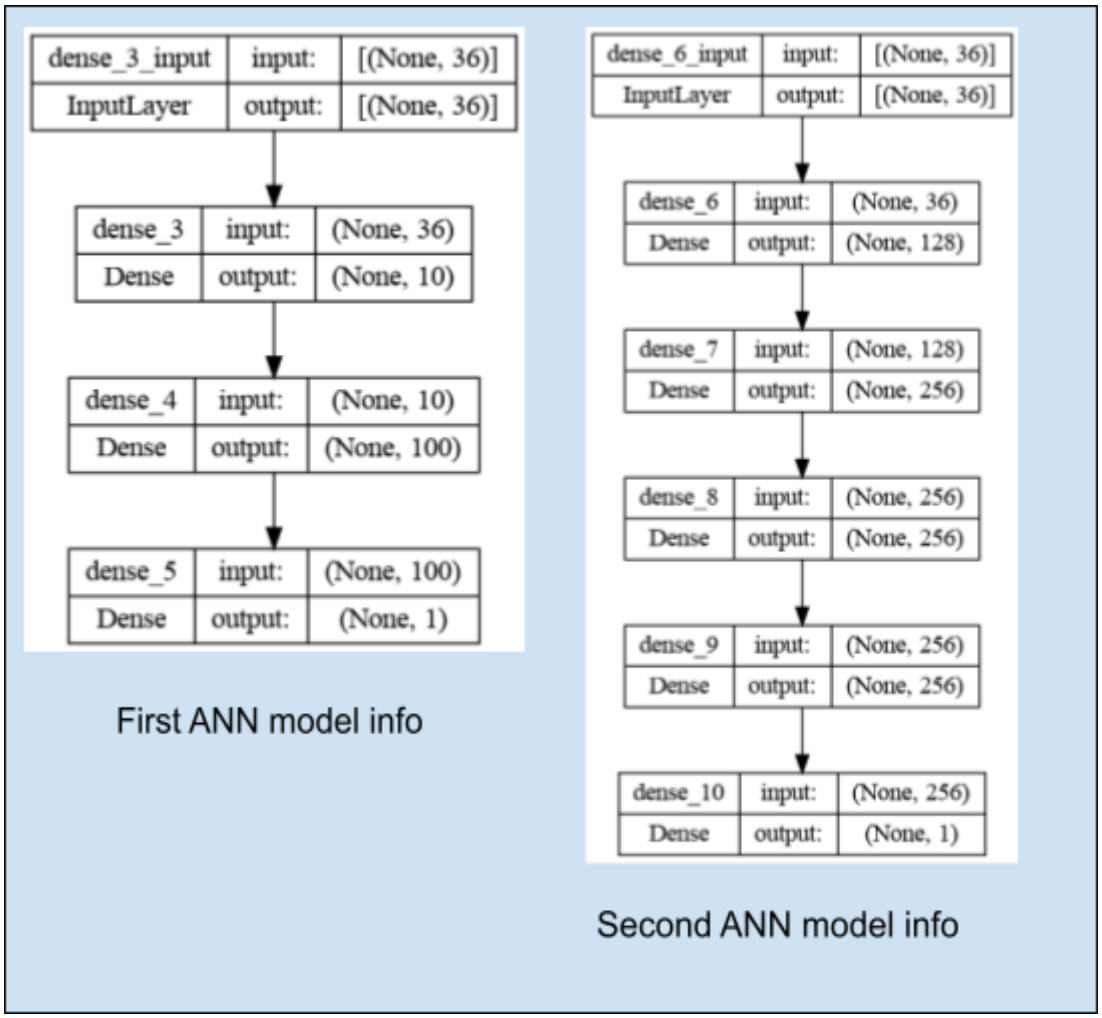


RMSE Error on test dataset: 362.416

Graphs of Bayesian Ridge Model and SGD Regressor model were same as that of Ridge and Lasso models.

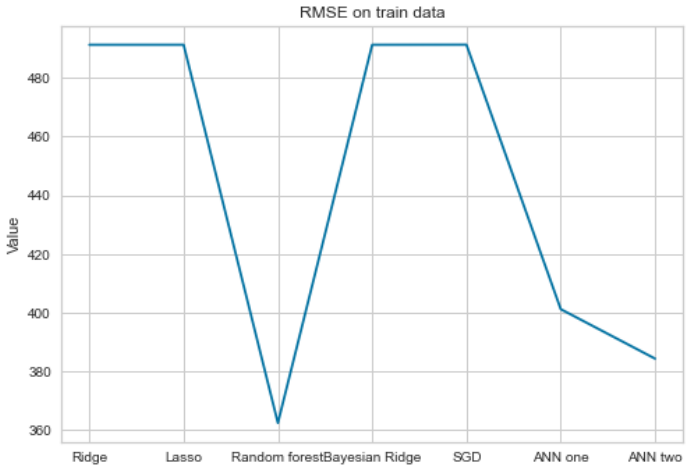
- Artificial Neural Network
  We build 2 models:
  1. One with a single hidden layer having 100 units
  2. One with 3 hidden layers having 256 hidden units each.

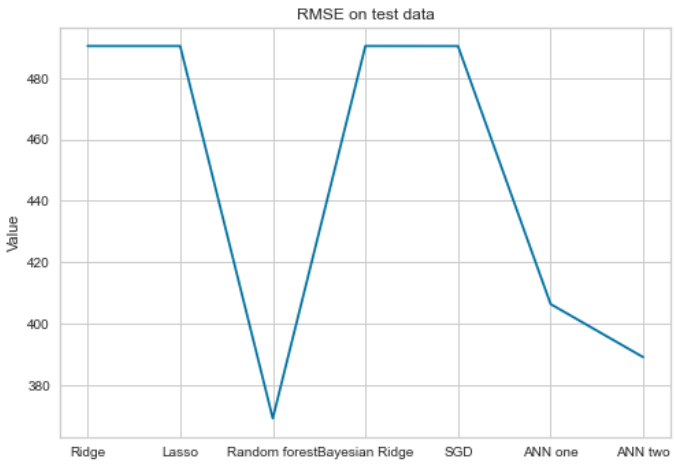
The RMSE errors of the respective models on the training and test data were:
406.301, 401.184
388.973, 384.338



Result Interpretation:

RMSE comparison for various models:





Since most part of the given dataset was redundant and had to be dropped. There was hardly any high correlation between the attributes and the target attribute price, except priceClass, cabinClass and rate.

We notice here that the Artificial Neural Network model fails to beat the Random Regressor, that can be due to the non-complexity of the problem where the model is deeply complex. Also the redundancy of the dataset too affects this fact.

Hence, based on the RMSE of the various models that were implemented, The Random Forest regressor seems to be the relatively best model here for predicting the flight prices depending on the attributes.