GA-ANN House Price

Octiba Nima Group

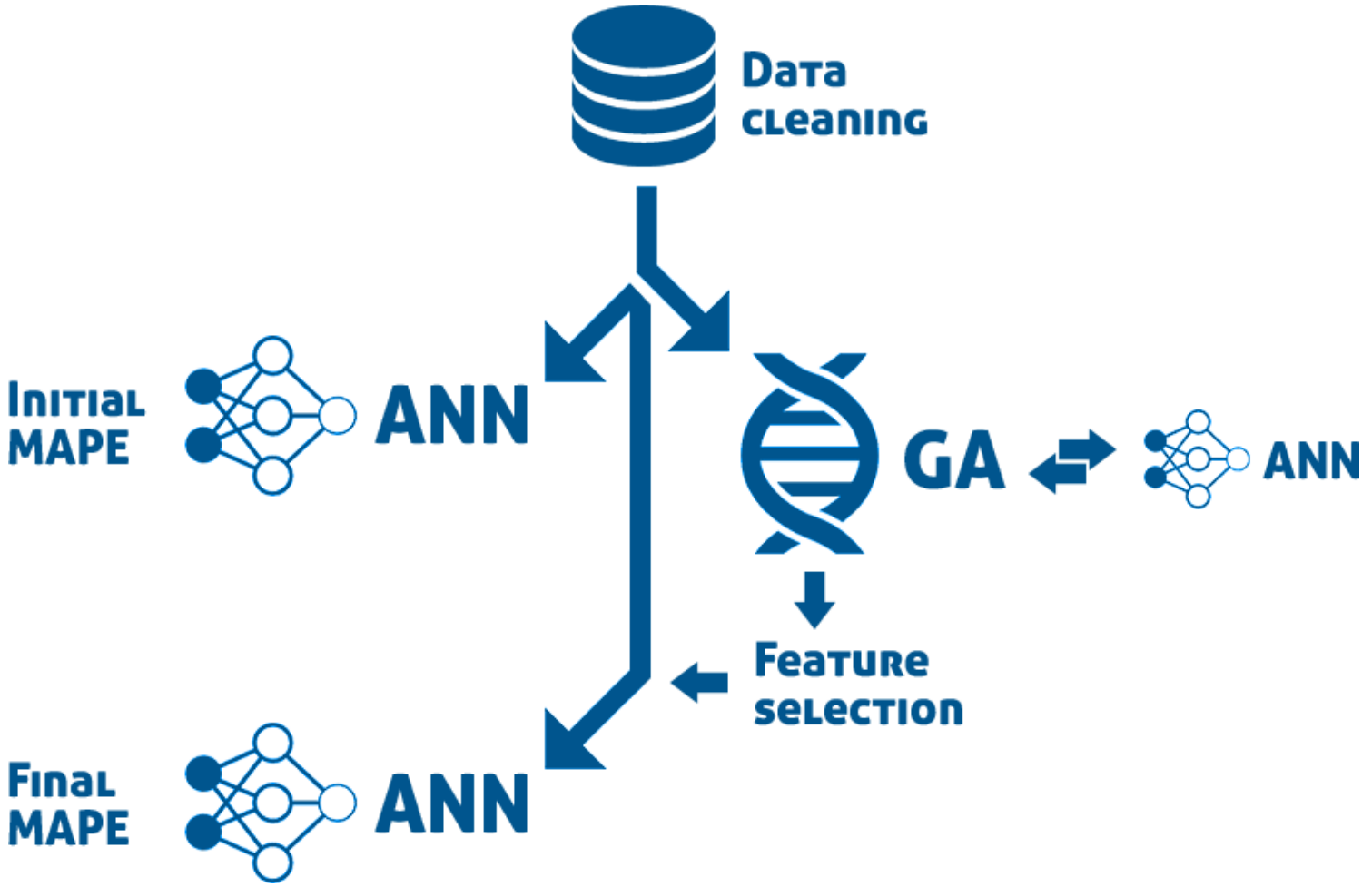
Nils | Markus | Renny | Febrianti

Assignment Description

- Given a dataset with around 80 features describing characteristics of residential homes in Ames Iowa
- The task is to predict the final price of each home
- Optimize performance of predictive models by using Genetic Algorithm for features selection

Technology Used

Modules summary



Data Cleaning

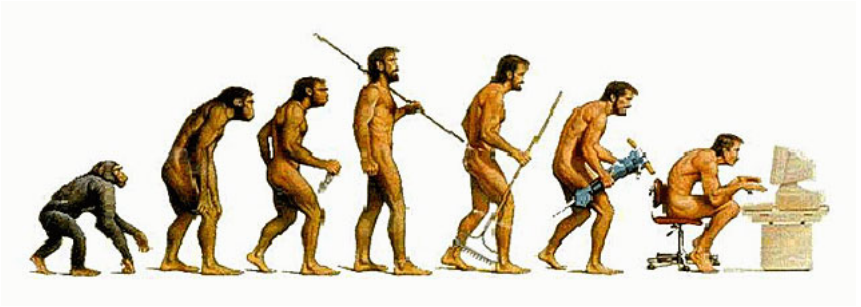
- 1 dataset of residential homes in Ames Iowa with around 80 features
- Drop columns that is not useful such as, PID and Order
- Handle missing values
  - Zero / null as values : fill in as Nan / zero
  - Missing values : fill with median values
- Cleanse inconsistency of paired features

Data Cleaning

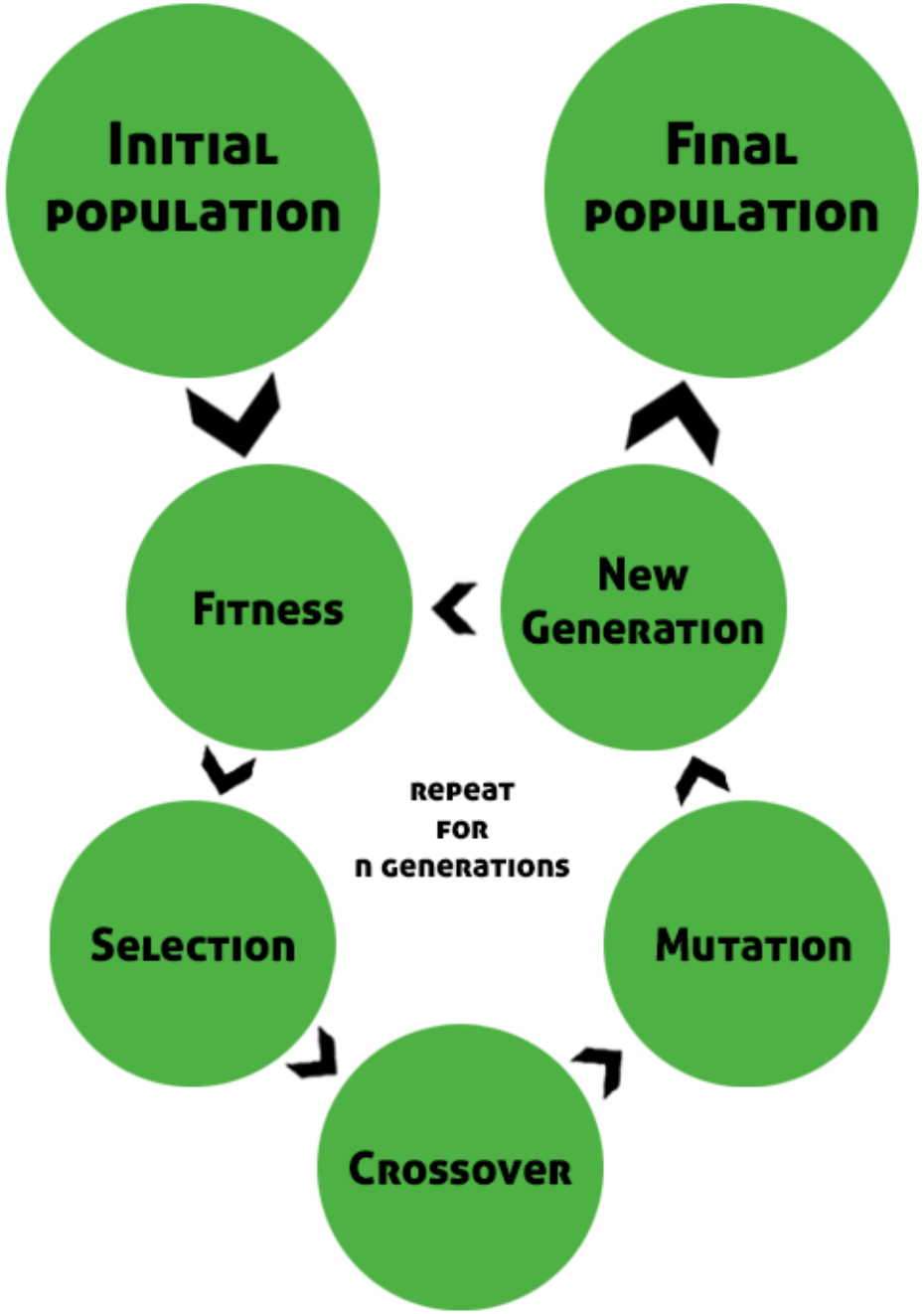
- Convert categorical to **numerical** features
  - Rank based on mean of SalePrice Feature
  - Rank based on common understanding
  - Without rank
- Remove outliers
- Features engineering
  - Totalsize(total squarefeet)
  - Remodeled
  - TotalBathroom
  - TotalGarage

Genetic Algorithm

- **Darwins theory of evolution, natural selection.**
  - preserve and accumulate advantageous mutations
  - advantage is passed on to offspring
  - inferior individuals gradually die off
  - superior individuals survive
- **Survival of the fittest.**
  - The goal is to find the most fit individual with the best possible genes.
- **Pitfalls**
  - definition of advantageous
  - preserve and accumulate vs variance
  - local minimums



Genetic Algorithm



Genetic Algorithm

- **Initial population**
  - population consists of chromosomes, individuals
  - chromosomes consists of genes
  - our case, boolean vector, features on or off
- **Pitfalls**
  - when initializing, how many features on?
  - population size, how many individuals neccesary to acheive enough diversity?

Genetic Algorithm

- **Fitness**
  - goal is to predict house prices
  - scikit-learns MLPRegressor
  - translate chromosome into feature selections
  - predict house prices in train data, calculate MSE
  - get training score
  - convert the MSE or training score to fitness score to be used as probabilities.
    - MSE: invert, high is better, low is worse
    - Score: scale to positive numbers, minmax
- **Pitfalls**
  - never mix test data with training!
  - efficiency, enough for disseminating the individuals from eachother

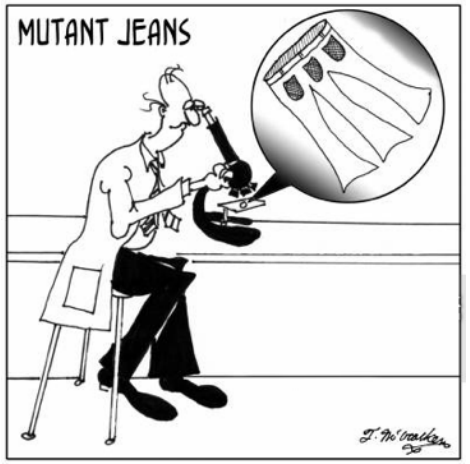
Genetic Algorithm

- **Selection**
  - mating pool, different strategies:
    - pure roulette
    - roulette with elitism
    - exponential rank
- **Pitfalls**
  - roulette, strong individuals become dominating and vice versa with low variance and high population size difficult to preserve and accumulate advantage

Genetic Algorithm

- **Crossover**
  - choice of mate, different strategies:
    - random
    - sequenced coupling
    - roulette based on fitness (not implemented)
  - crossover implementations
    - multipoint
    - uniform
  - elitism
    - reserve a portion of the next generation for the best fit individuals in current generation
    - exempted from crossover (but in mating pool) and mutation
- **Pitfalls**
  - too much crossover can ruin advantageous chromosomes
  - too little crossover can lead to slow convergence

Genetic Algorithm



- **Mutation**
  - mutation is implemented so that at a set rate, parts of the chromosome flip
  - introduces variance in population
- **Pitfalls**
  - similar to crossover, too much can ruin, too little slow convergence

Genetic Algorithm

- **Parameter values, what to set?**
  - GA implementation has a lot of parameters affecting the evolution:
    - initrate
    - crossover rate
    - mutation rates
    - population size
    - number of generations
    - how much elitism
    - fitness function efficiency
      - iterations
      - hidden layers

Genetic Algorithm

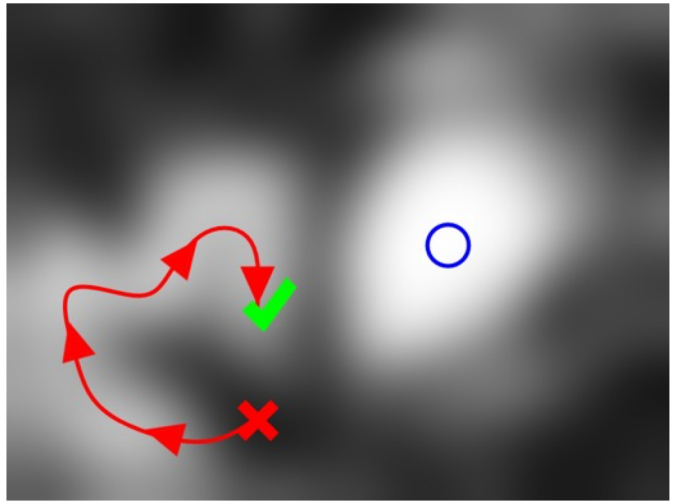
- **Parameter sweeps**
  - baseline
  - systematic, easy to implement
  - requires a basic knowledge of what ranges to sweep
    - sometimes values outside of what is expected is good
  - produces a lot of data
    - difficult to make sense
    - statistics!
  - not good for finding out what parameters work together

Genetic Algorithm

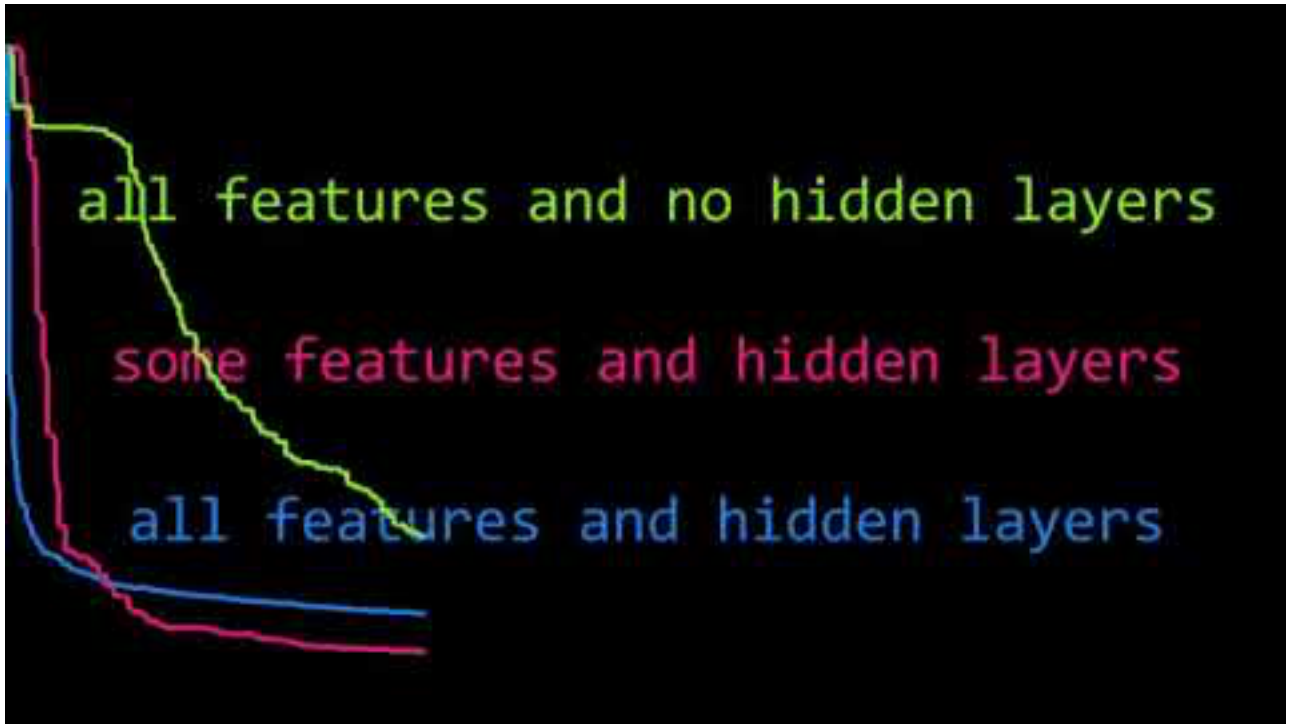
- **Meta-GA**
  - GA is good for this kind of problems
  - better chance of finding synergy effects of diff. parameters
  - a new chromosome, fitness function and mutation algorithm
  - population of GA's with different parameters
  - very time consuming!

Artificial Neural Network

- ANN module
- *The data structure*
- *The learning*
- The activation function
- Local minimums
- Over-fitting



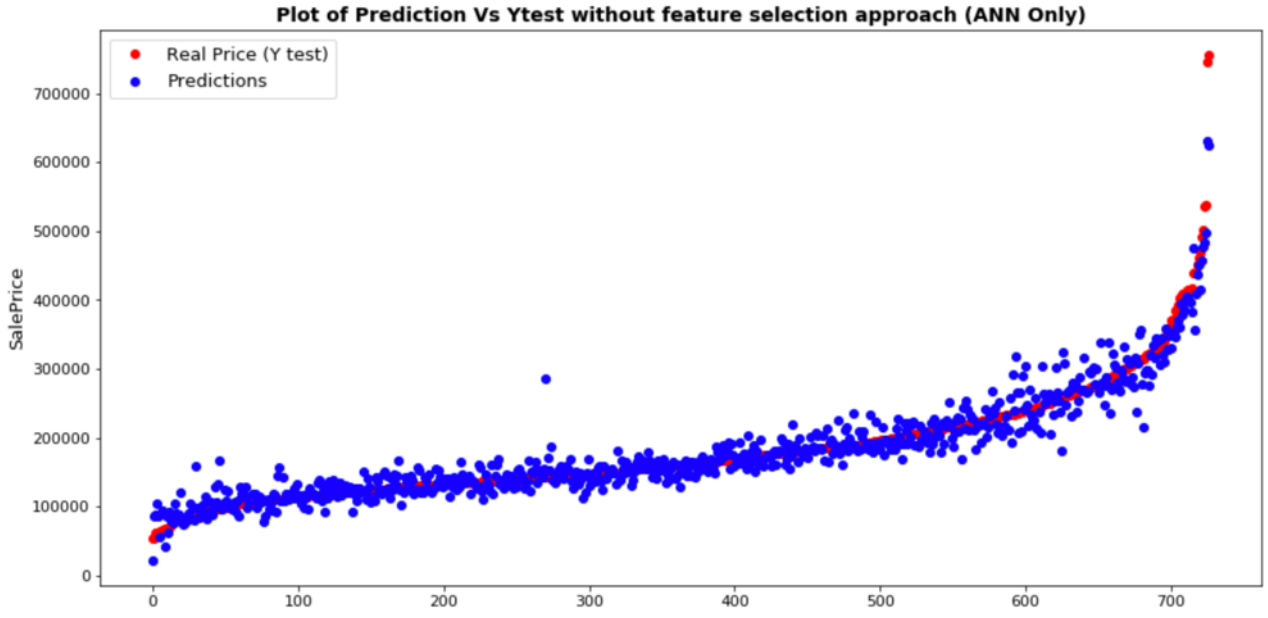
Artificial Neural Network



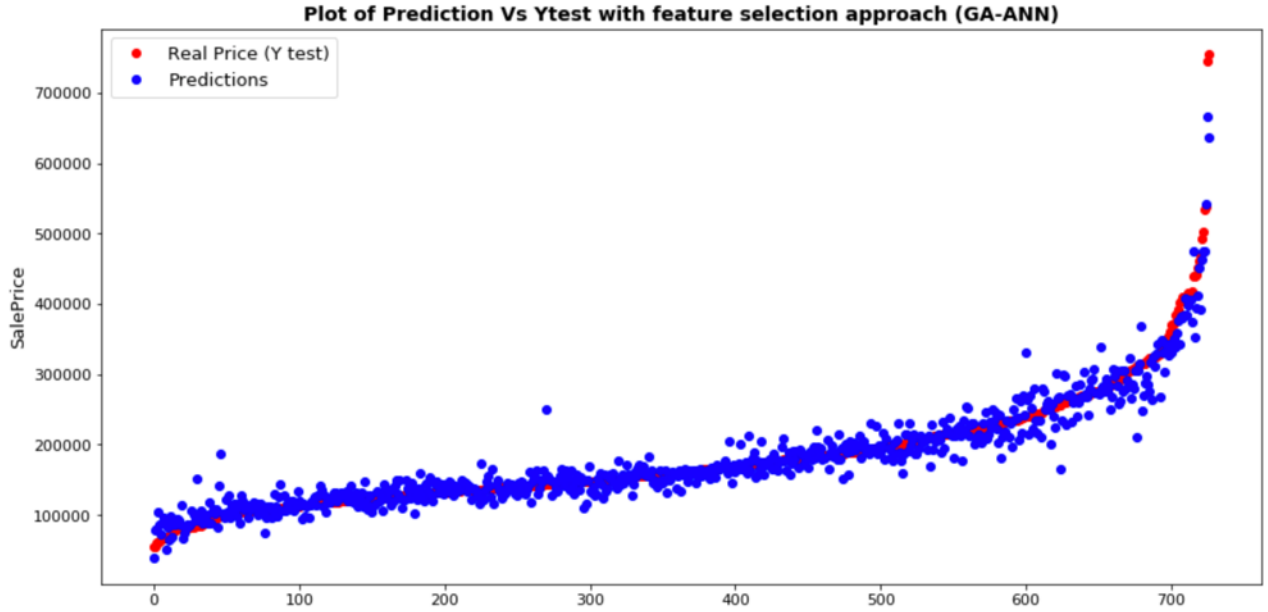
Results

- **Example 1 :** results of running with below parameters
  - init_Ratio = 0.5
  - cross_rate = 0.5
  - mutate_rate = 0.002
  - pop_size = 120
  - n_generations = 100
  - elitism = 0.05
  - ga_ann_iterations =100
  - ga_ann_layers =2
  - mape_ann_iterations =1000
  - mape_ann_layers = 4
  - ga_score='score’
  - ga_evolve='elitism',
  - final_mape_idx='best')

```
Mape for prediction result without feature selection approach (ANN only) =  8.475532981326609 %
```

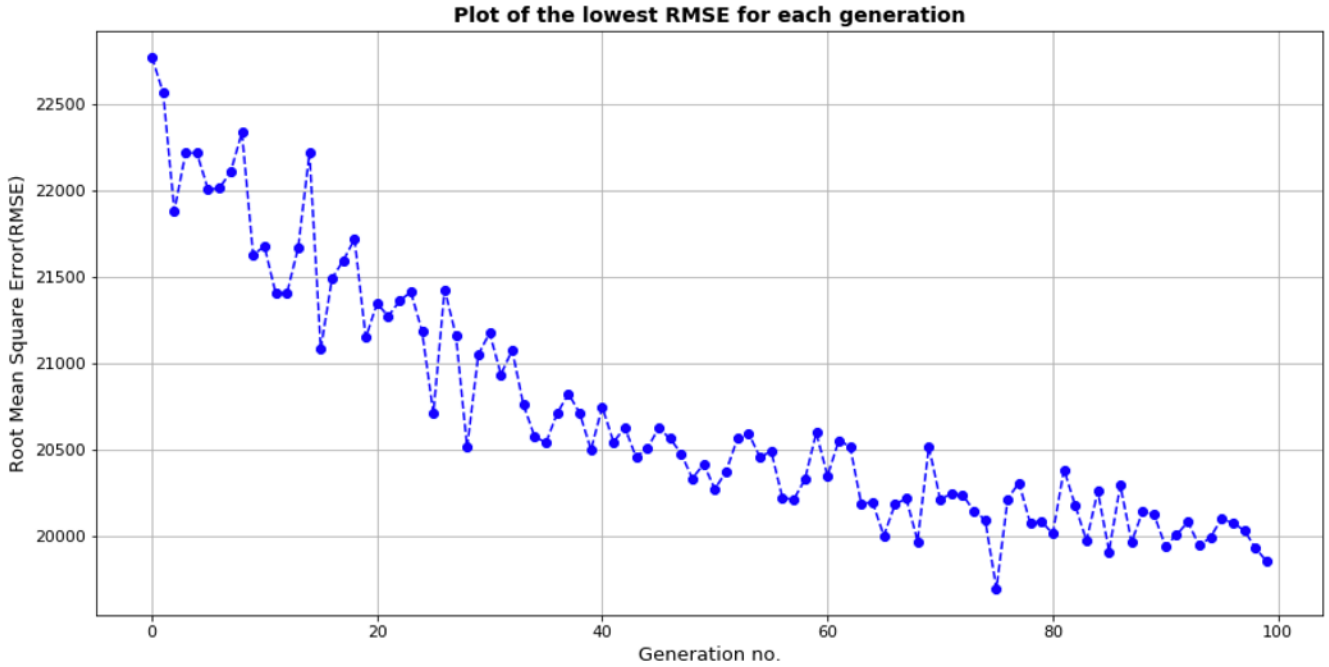


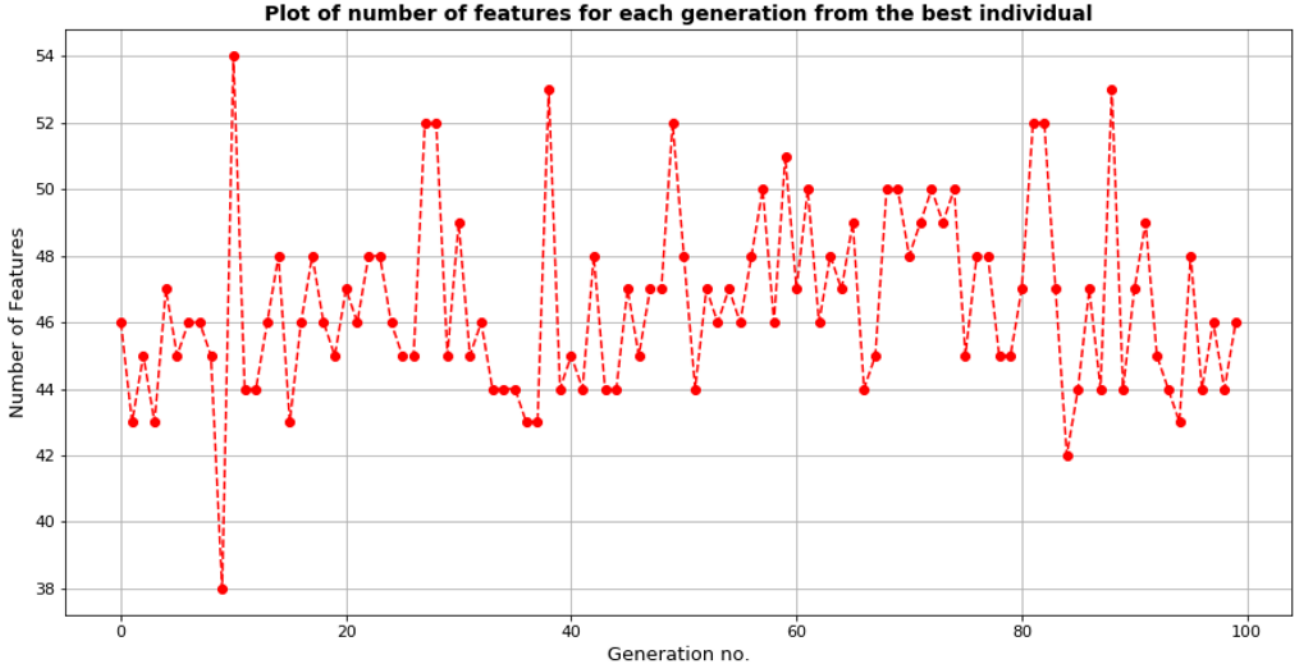

```
Mape for prediction result with feature selection approach (GA-ANN): 7.916424526075956 %
```



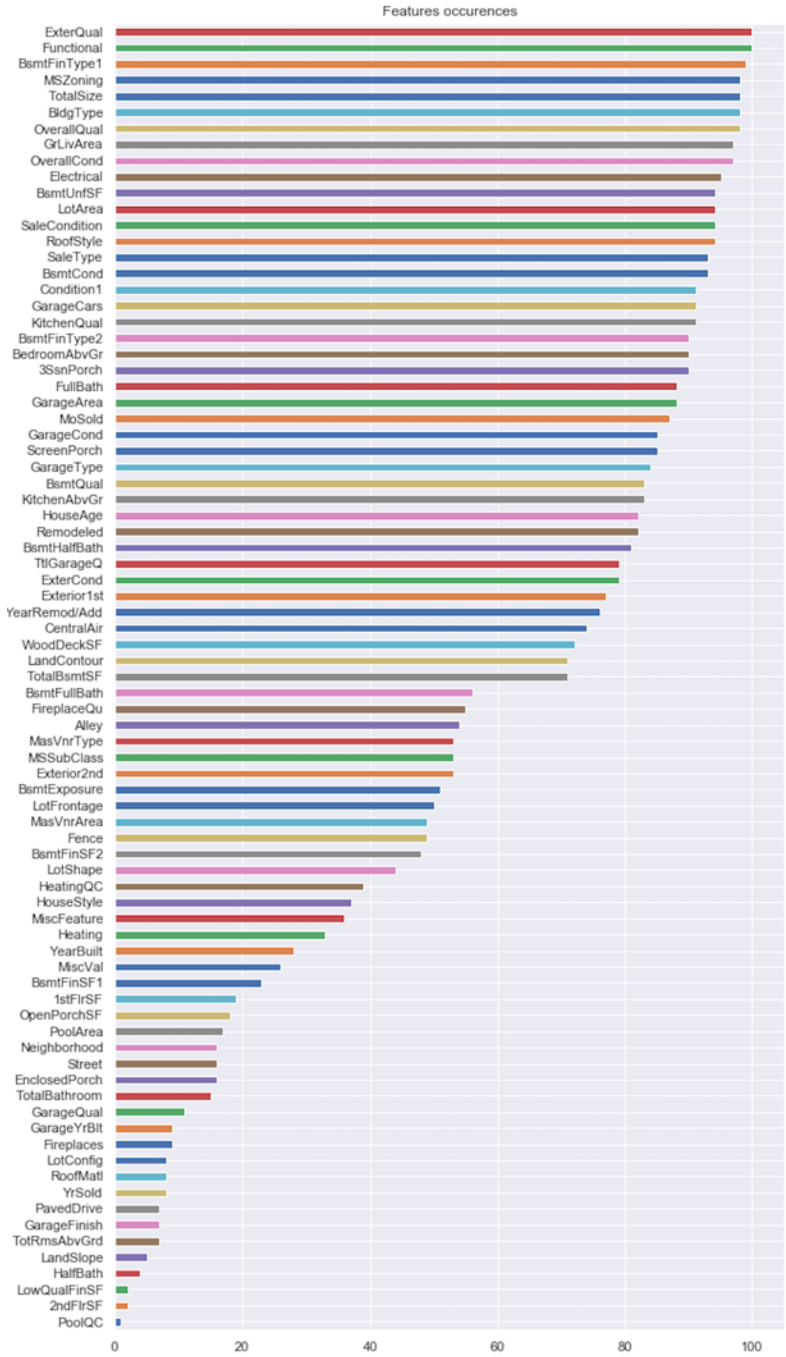
	MAPE	Run Time	No. of Features
Without Feature Selection	8.48%	0.734 s	81
With Feature Selection GA-ANN	7.92%	0.612 s	45
Improvement	**6.6% accuracy**	**16.614 % run time**	

Results

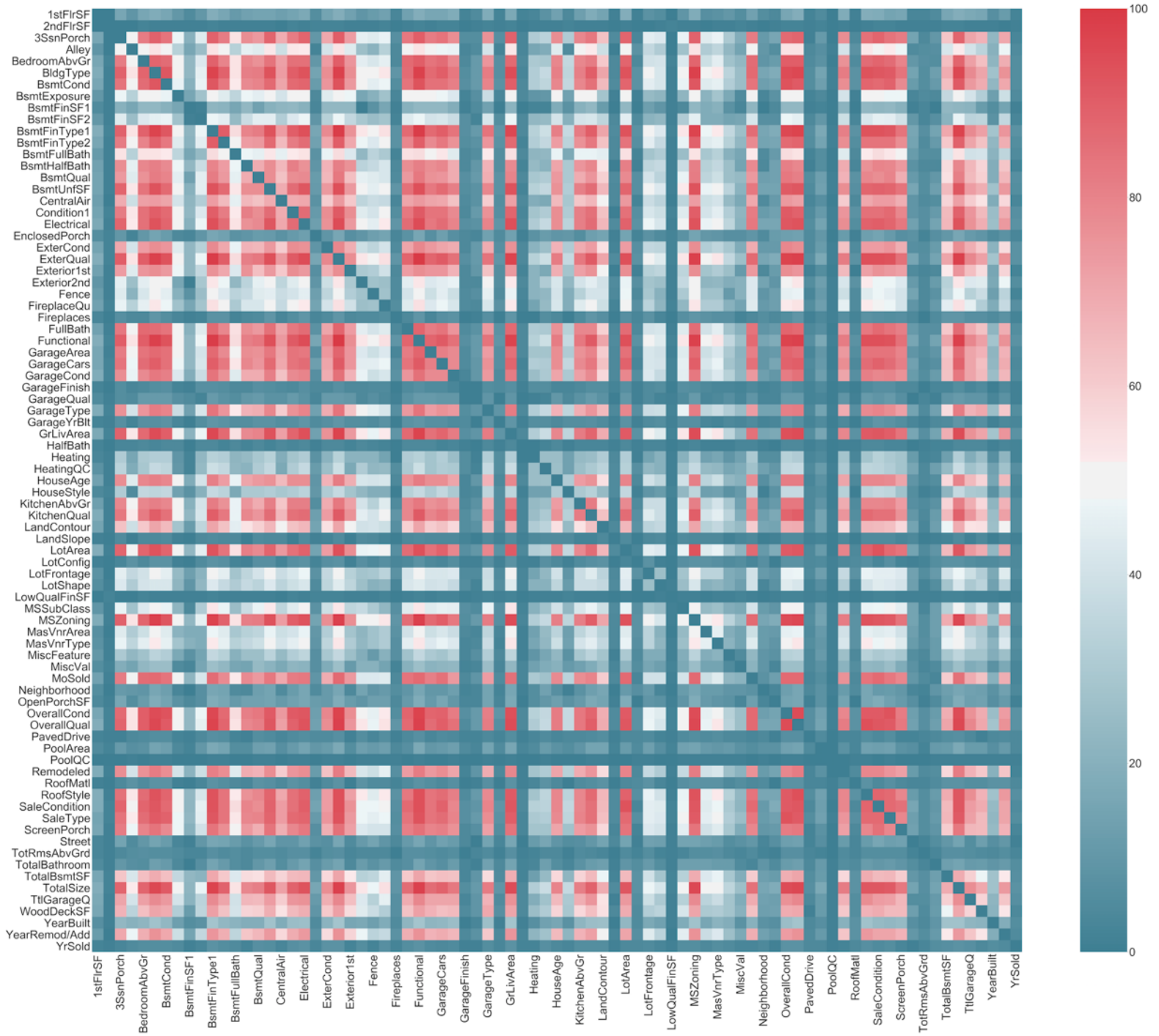

Plot of the lowest RMSE for each generation
Root Mean Square Error(RMSE)
Generation no.
22500
22000
21500
21000
20500
20000
0
20
40
60
80
100


Plot of number of features for each generation from the best individual
Number of Features
Generation no.
54
52
50
48
46
44
42
40
38
0
20
40
60
80
100

Results


Features occurences
ExterQual
Functional
BsmtFinType1
MSZoning
TotalSize
BldgType
OverallQual
GrLivArea
OverallCond
Electrical
BsmtUnfSF
LotArea
SaleCondition
RoofStyle
SaleType
BsmtCond
Condition1
GarageCars
KitchenQual
BsmtFinType2
BedroomAbvGr
3SsnPorch
FullBath
GarageArea
MoSold
GarageCond
ScreenPorch
GarageType
BsmtQual
KitchenAbvGr
HouseAge
Remodeled
BsmtHalfBath
TtlGarageQ
ExterCond
Exterior1st
YearRemod/Add
CentralAir
WoodDeckSF
LandContour
TotalBsmtSF
BsmtFullBath
FireplaceQu
Alley
MasVnrType
MSSubClass
Exterior2nd
BsmtExposure
LotFrontage
MasVnrArea
Fence
BsmtFinSF2
LotShape
HeatingQC
HouseStyle
MiscFeature
Heating
YearBuilt
MiscVal
BsmtFinSF1
1stFlrSF
OpenPorchSF
PoolArea
Neighborhood
Street
EnclosedPorch
TotalBathroom
GarageQual
GarageYrBlt
Fireplaces
LotConfig
RoofMatl
YrSold
PavedDrive
GarageFinish
TotRmsAbvGrd
LandSlope
HalfBath
LowQualFinSF
2ndFlrSF
PoolQC
0
20
40
60
80
100

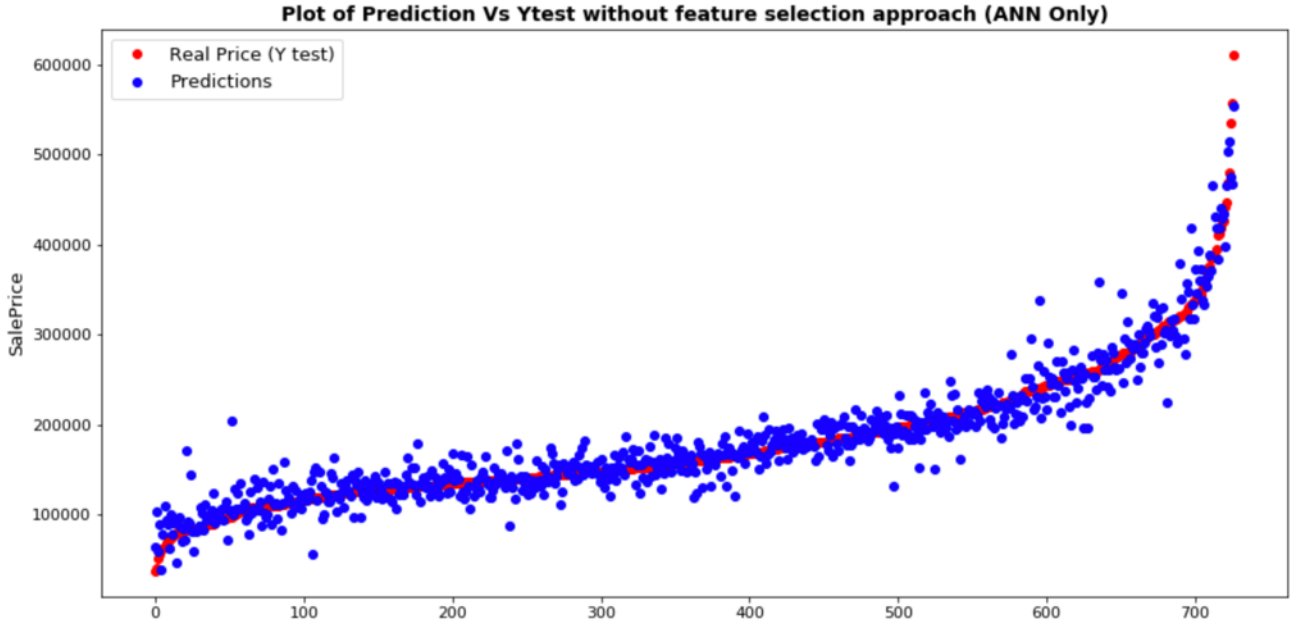
Results


1stFlrSF
2ndFlrSF
3SsnPorch
Alley
BedroomAbvGr
BldgType
BsmtCond
BsmtExposure
BsmtFinSF1
BsmtFinSF2
BsmtFinType1
BsmtFinType2
BsmtFullBath
BsmtHalfBath
BsmtQual
BsmtUnfSF
CentralAir
Condition1
Electrical
EnclosedPorch
ExterCond
ExterQual
Exterior1st
Exterior2nd
Fence
FireplaceQu
Fireplaces
FullBath
Functional
GarageArea
GarageCars
GarageCond
GarageFinish
GarageQual
GarageType
GarageYrBlt
GrLivArea
HalfBath
Heating
HeatingQC
HouseAge
HouseStyle
KitchenAbvGr
KitchenQual
LandContour
LandSlope
LotArea
LotConfig
LotFrontage
LotShape
LowQualFinSF
MSSubClass
MSZoning
MasVnrArea
MasVnrType
MiscFeature
MiscVal
MoSold
Neighborhood
OpenPorchSF
OverallCond
OverallQual
PavedDrive
PoolArea
PoolQC
Remodeled
RoofMatl
RoofStyle
SaleCondition
SaleType
ScreenPorch
Street
TotRmsAbvGrd
TotalBathroom
TotalBsmtSF
TotalSize
TtlGarageQ
WoodDeckSF
YearBuilt
YearRemod/Add
YrSold
100
80
60
40
20
0

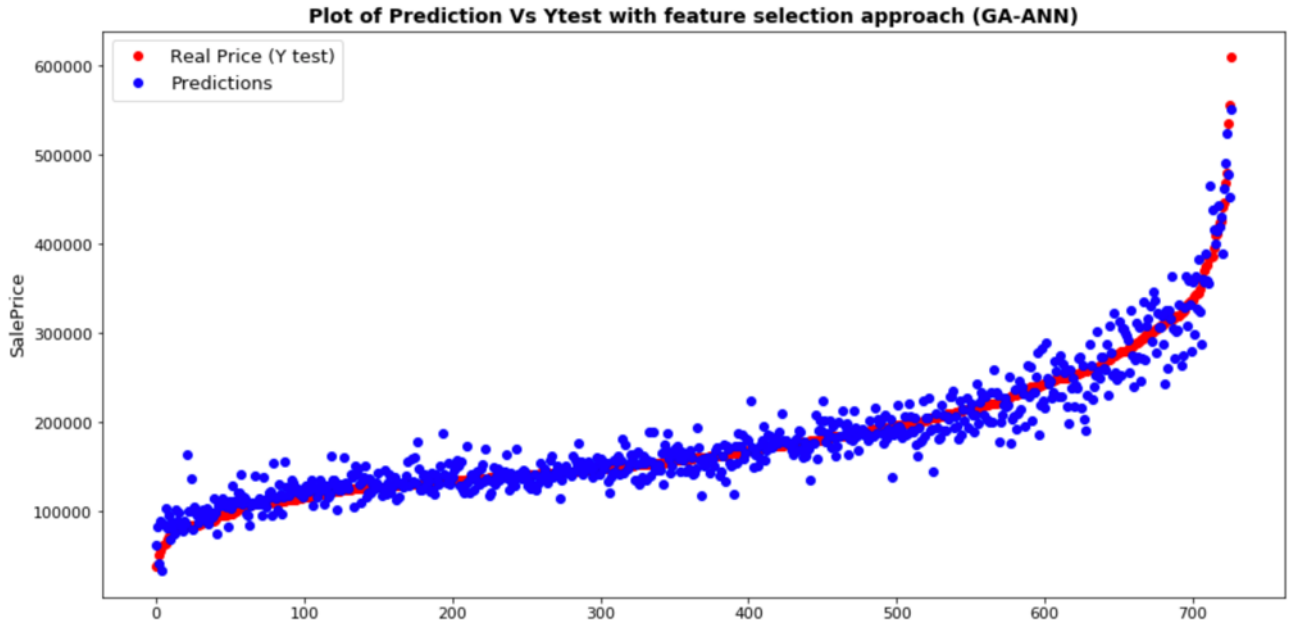
Results

- **Example 2 :** results of running with below parameters
  - **init_Ratio = 0.1**
  - cross_rate = 0.5
  - mutate_rate = 0.002
  - pop_size = 120
  - n_generations = 100
  - elitism = 0.05
  - ga_ann_iterations =100
  - ga_ann_layers =2
  - mape_ann_iterations =1000
  - mape_ann_layers = 4

```
Mape for prediction result without feature selection approach (ANN only) =  9.081127250637053 %
```




```
Mape for prediction result with feature selection approach (GA-ANN): 8.63879207408662 %
```



Without GA : 0.416 **s**

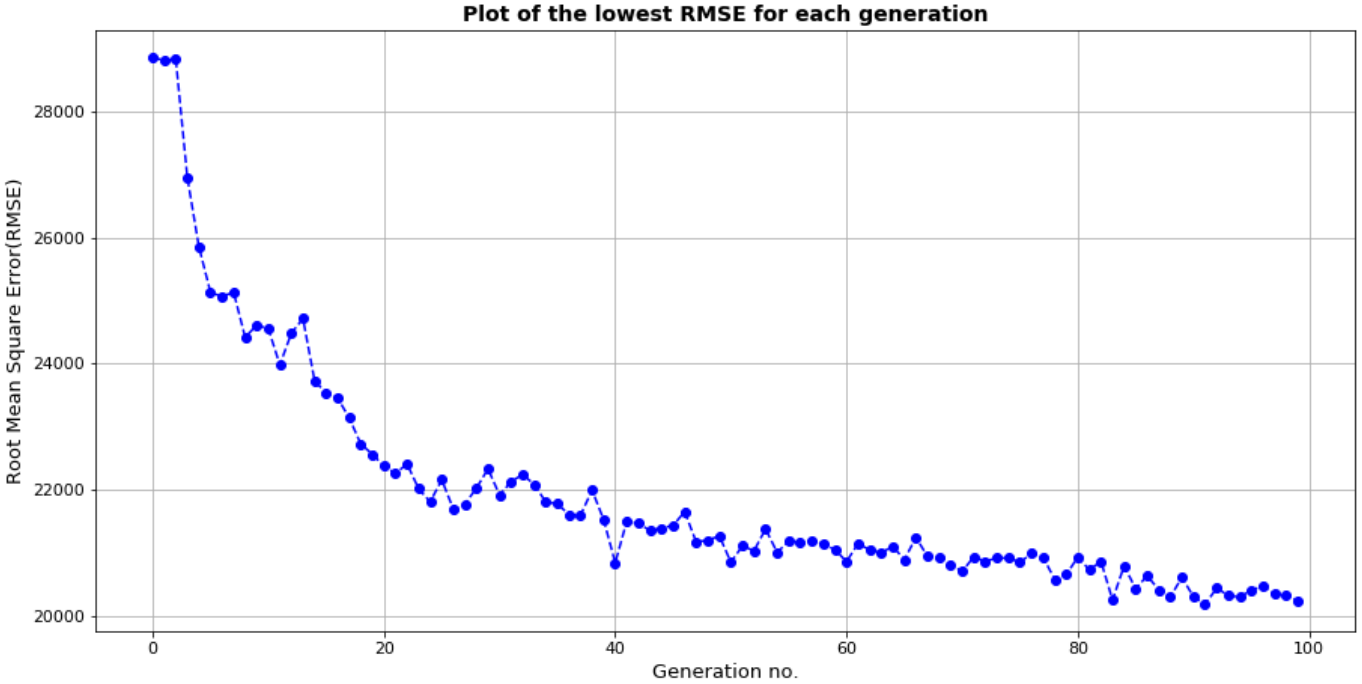
With GA : 0,343 **s**

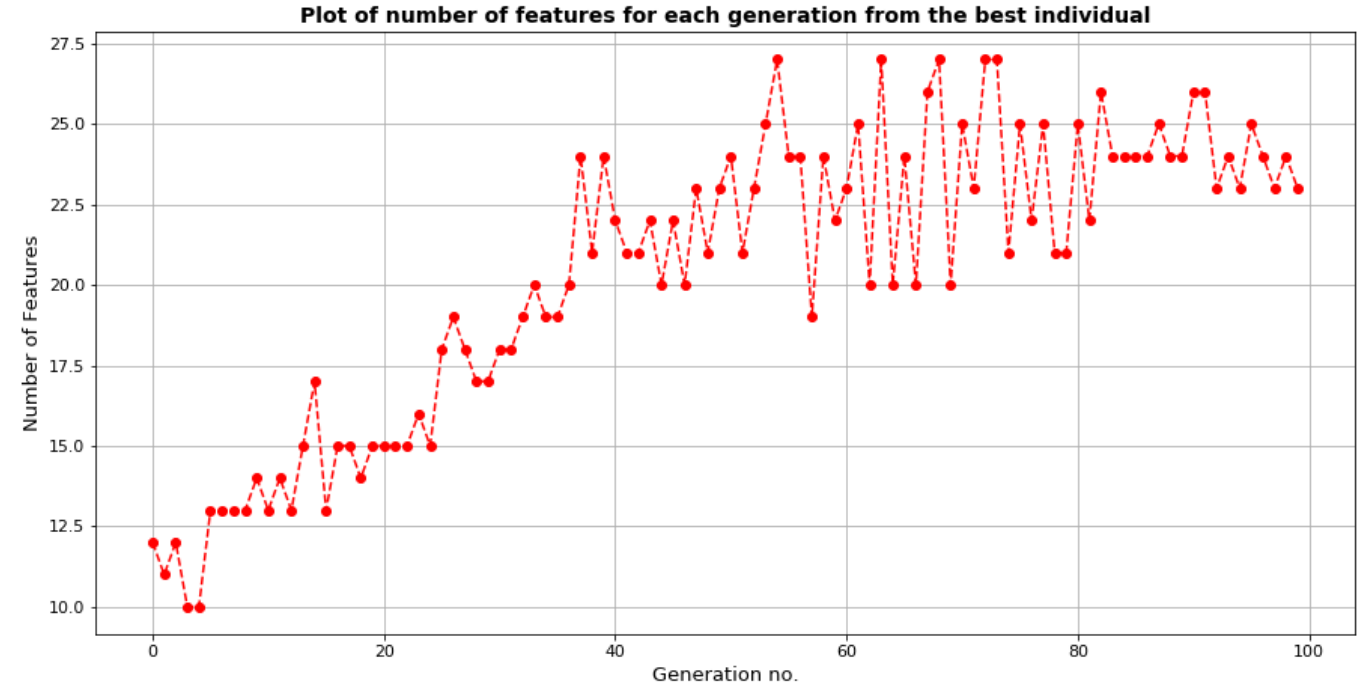
Selected Features : 26

Improvement : 17.38%

	MAPE	Run Time	No. of Features
Without Feature Selection	9.08%	0.416 s	81
With Feature Selection GA-ANN	8.64%	0.343 s	26
Improvement	**4.84% accuracy**	**17.38 % run time**	

Results

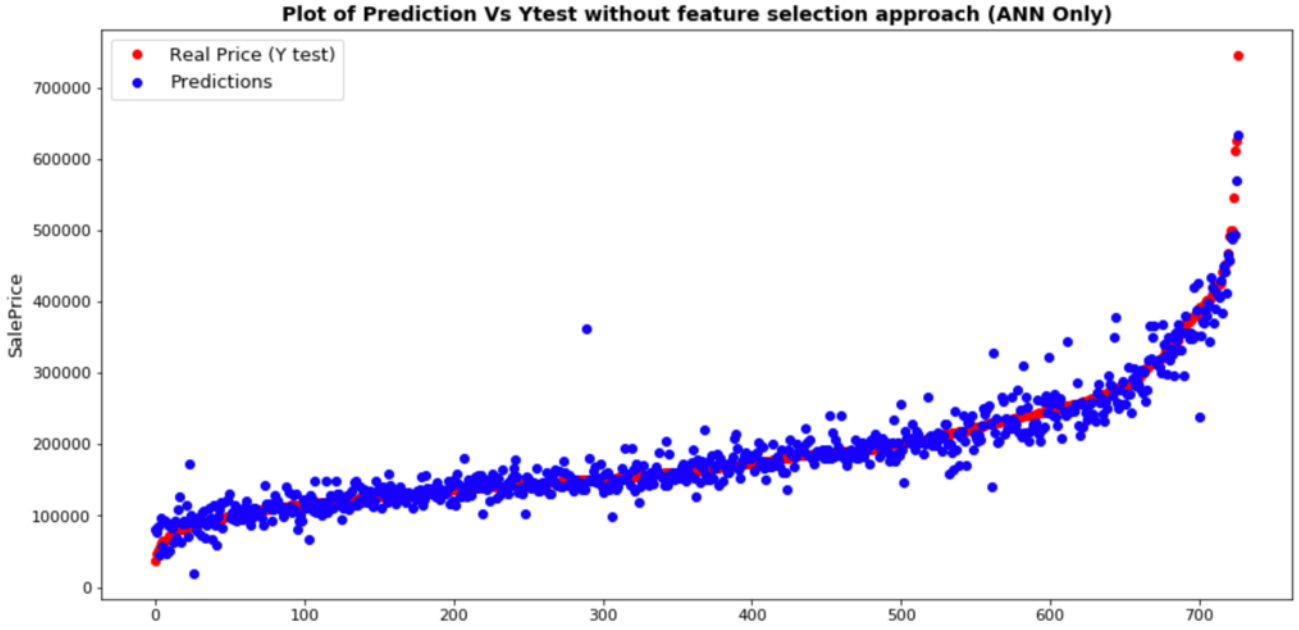

Plot of the lowest RMSE for each generation
Root Mean Square Error(RMSE)
Generation no.
28000
26000
24000
22000
20000
0
20
40
60
80
100


Plot of number of features for each generation from the best individual
Number of Features
Generation no.
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
0
20
40
60
80
100

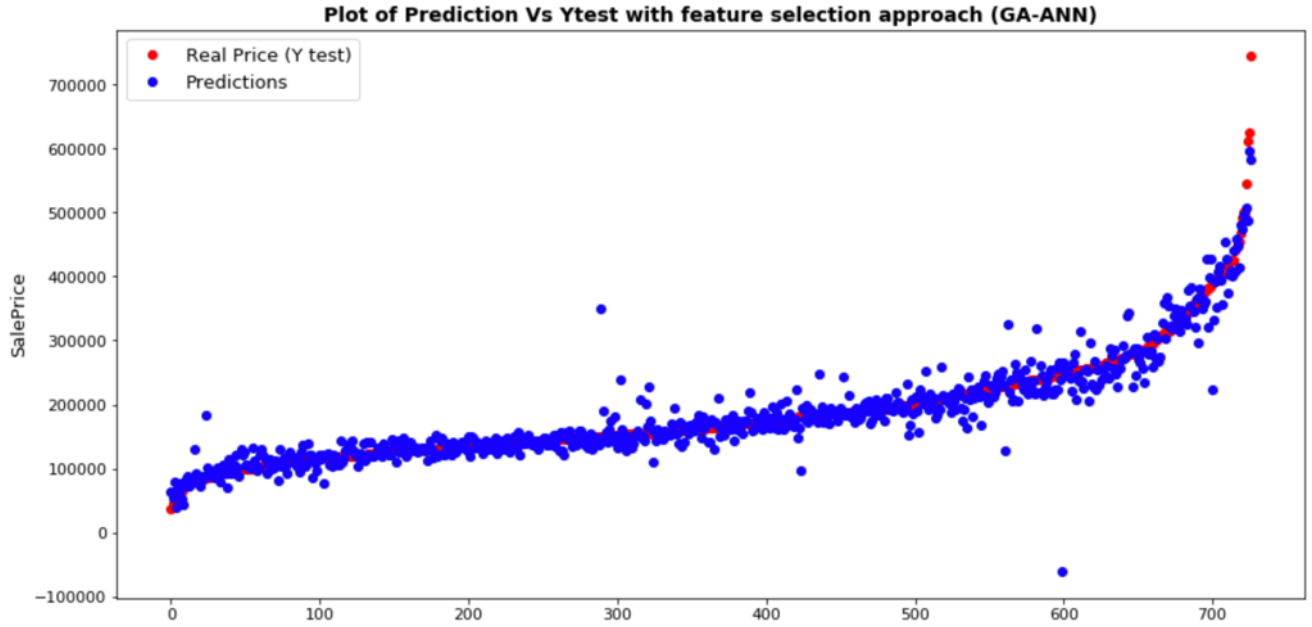
Results

- **Example 3 :** results of running with below parameters
  - **init_Ratio = 0.9**
  - cross_rate = 0.5
  - mutate_rate = 0.002
  - pop_size = 120
  - n_generations = 100
  - elitism = 0.05
  - ga_ann_iterations =100
  - ga_ann_layers =2
  - mape_ann_iterations =1000
  - mape_ann_layers = 4

```
Mape for prediction result without feature selection approach (ANN only) =  9.246340278044626 %
```

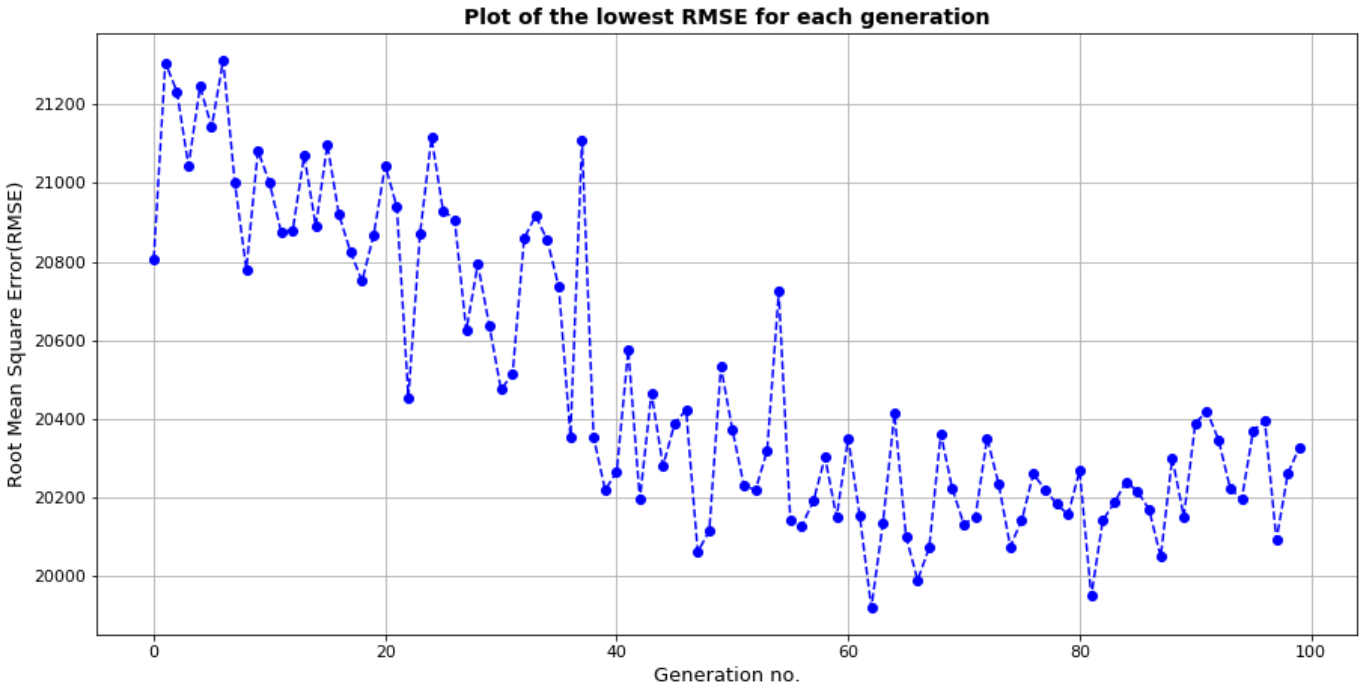


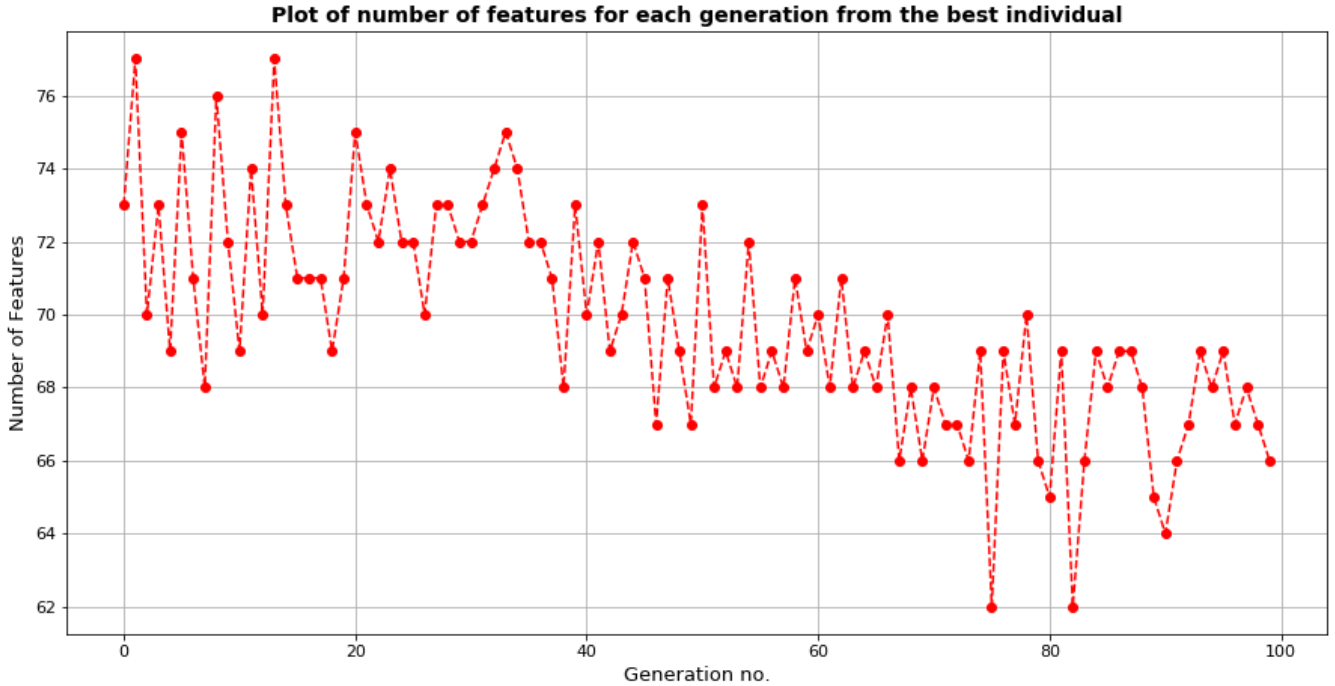

```
Mape for prediction result with feature selection approach (GA-ANN): 8.305010474158319 %
```



	MAPE	Run Time	No. of Features
Without Feature Selection	9.25%	0.4101 s	81
With Feature Selection GA-ANN	8.31%	0.4011 s	71
Improvement	**10.16% accuracy**	**2.17% run time**	

Results


Plot of the lowest RMSE for each generation
Root Mean Square Error(RMSE)
Generation no.
21200
21000
20800
20600
20400
20200
20000
0
20
40
60
80
100


Plot of number of features for each generation from the best individual
Number of Features
Generation no.
76
74
72
70
68
66
64
62
0
20
40
60
80
100

Conclusion & Lesson Learned

- It is possible to improve efficiency by feature selection, and keep a reasonable or same level of accuracy
- A lot of parameters to play around with and need time to tune the parameters for getting the best result
- The results varies dependent on initial split of data, so could be valuable to run the process many times and do some statistics