```
import pandas as pd
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_selection import mutual_info_regression, RFE
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler, FunctionTransformer
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
```

```
dataset = pd.read_csv("insurance.csv")
dataset
```

{"type":"dataframe","variable_name":"dataset"}

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58592 entries, 0 to 58591
Data columns (total 45 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   policy_id                         58592 non-null  object
 1   policy_age                        58592 non-null  int64
 2   age_of_car                        58592 non-null  int64
 3   age_of_policyholder               58592 non-null  int64
 4   area                              58592 non-null  object
 5   population_density                58592 non-null  int64
 6   make                              58592 non-null  object
 7   segment                           58592 non-null  object
 8   model                             58592 non-null  object
 9   fuel_type                         58592 non-null  object
 10  max_torque                        58592 non-null  object
 11  max_power                         58592 non-null  object
 12  engine_type                       58592 non-null  object
 13  airbags                           58592 non-null  int64
 14  is_esc                            58592 non-null  object
 15  is_adjustable_steering            58592 non-null  object
 16  is_tpms                           58592 non-null  object
 17  is_parking_sensors                58592 non-null  object
 18  is_parking_camera                 58592 non-null  object
 19  rear_brakes_type                  58592 non-null  object
 20  displacement                      58592 non-null  object
 21  cylinder                          58592 non-null  int64  
 22  transmission_type                 58592 non-null  object 
 23  gear_box                          58592 non-null  int64  
 24  steering_type                     58592 non-null  object 
 25  turning_radius                    58592 non-null  float64
 26  length                            58592 non-null  object 
 27  width                             58592 non-null  object 
 28  height                            58592 non-null  object 
 29  gross_weight                      58592 non-null  object 
 30  is_front_fog_lights               58592 non-null  object 
 31  is_rear_window_wiper              58592 non-null  object 
 32  is_rear_window_washer             58592 non-null  object 
 33  is_rear_window_defogger           58592 non-null  object 
 34  is_brake_assist                   58592 non-null  object 
 35  is_power_door_locks               58592 non-null  object 
 36  is_central_locking               58592 non-null  object 
 37  is_power_steering                 58592 non-null  object 
 38  is_driver_seat_height_adjustable  58592 non-null  object 
 39  is_day_night_rear_view_mirror     58592 non-null  object 
 40  is_ecw                            58592 non-null  object 
 41  is_speed_alert                    58592 non-null  object 
 42  ncap_rating                       58592 non-null  int64  
 43  claims_in_5_years                 58592 non-null  int64  
 44  claim                             58592 non-null  object 
dtypes: float64(1), int64(9), object(35)
memory usage: 20.1+ MB
```

```
def clean_currency_column(column):
    return column.astype(str).apply(lambda x: round(float(re.sub(r'[^\d.]', '', x)), 2) if pd.notna(x) and x.strip() else 0)

dataset['claim'] = clean_currency_column(dataset['claim'])
dataset['claim'][:5]
```

```
0     773.69
1    2975.25
2     682.50
3    2544.25
4    3804.65
Name: claim, dtype: float64
```

```
def extract_number_till_letter(text, stop_letter):
    match = re.match(rf'(\d+\.?\d*)[{stop_letter}]', text)
    return float(match.group(1)) if match else None

dataset['max_torque'] = dataset['max_torque'].apply(lambda x: extract_number_till_letter(x, "N"))
dataset['max_power'] = dataset['max_power'].apply(lambda x: extract_number_till_letter(x, "b"))
```

```
dataset['length'] = dataset['length'].apply(lambda x:
extract_number_till_letter(x, "c"))
dataset['width'] = dataset['width'].apply(lambda x:
extract_number_till_letter(x, "c"))
dataset['height'] = dataset['height'].apply(lambda x:
extract_number_till_letter(x, "c"))

dataset['gross_weight'] = dataset['gross_weight'].apply(lambda x:
extract_number_till_letter(x, "k"))

dataset['displacement'] = dataset['displacement'].apply(lambda x:
extract_number_till_letter(x, "c"))

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58592 entries, 0 to 58591
Data columns (total 45 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   policy_id                         58592 non-null  object
 1   policy_age                        58592 non-null  int64
 2   age_of_car                        58592 non-null  int64
 3   age_of_policyholder               58592 non-null  int64
 4   area                              58592 non-null  object
 5   population_density                58592 non-null  int64
 6   make                              58592 non-null  object
 7   segment                           58592 non-null  object
 8   model                             58592 non-null  object
 9   fuel_type                         58592 non-null  object
 10  max_torque                        58592 non-null  float64
 11  max_power                         58592 non-null  float64
 12  engine_type                       58592 non-null  object
 13  airbags                           58592 non-null  int64
 14  is_esc                            58592 non-null  object
 15  is_adjustable_steering            58592 non-null  object
 16  is_tpms                           58592 non-null  object
 17  is_parking_sensors                58592 non-null  object
 18  is_parking_camera                 58592 non-null  object
 19  rear_brakes_type                  58592 non-null  object
 20  displacement                      58592 non-null  float64
 21  cylinder                          58592 non-null  int64
 22  transmission_type                 58592 non-null  object
 23  gear_box                          58592 non-null  int64
 24  steering_type                     58592 non-null  object
 25  turning_radius                    58592 non-null  float64
 26  length                            58592 non-null  float64
 27  width                             58592 non-null  float64
 28  height                            58592 non-null  float64
 29  gross_weight                      58592 non-null  float64
 30  is_front_fog_lights               58592 non-null  object 
 31  is_rear_window_wiper              58592 non-null  object 
 32  is_rear_window_washer             58592 non-null  object 
 33  is_rear_window_defogger           58592 non-null  object 
 34  is_brake_assist                   58592 non-null  object 
 35  is_power_door_locks               58592 non-null  object 
 36  is_central_locking                58592 non-null  object 
 37  is_power_steering                 58592 non-null  object 
 38  is_driver_seat_height_adjustable  58592 non-null  object 
 39  is_day_night_rear_view_mirror     58592 non-null  object 
 40  is_ecw                            58592 non-null  object 
 41  is_speed_alert                    58592 non-null  object 
 42  ncap_rating                       58592 non-null  int64  
 43  claims_in_5_years                 58592 non-null  int64  
 44  claim                             58592 non-null  float64
dtypes: float64(9), int64(9), object(27)
memory usage: 20.1+ MB
```

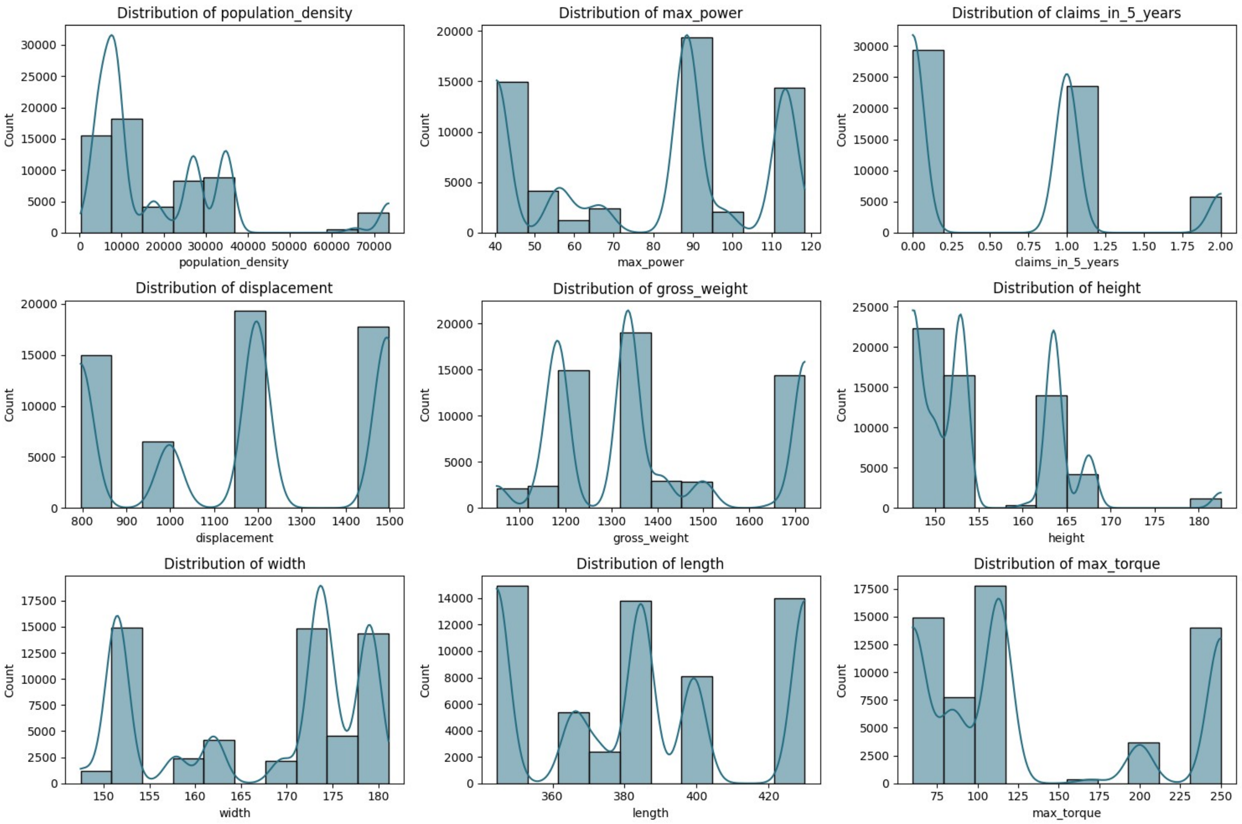
```
yes_no_columns = ['is_esc', 'is_adjustable_steering', 'is_tpms', 
'is_parking_sensors', 'is_parking_camera',
                  'is_front_fog_lights', 'is_rear_window_wiper', 
'is_rear_window_washer', 'is_rear_window_defogger',
                  'is_brake_assist', 'is_power_door_locks', 
'is_central_locking', 'is_power_steering',
                  'is_driver_seat_height_adjustable', 
'is_day_night_rear_view_mirror', 'is_ecw', 'is_speed_alert']

onehot_columns = ['area', 'make', 'segment', 'model', 'fuel_type', 
'engine_type', 'rear_brakes_type',
                  'transmission_type', 'steering_type']

for col in yes_no_columns:
    dataset[col] = dataset[col].map({'Yes':1, 'No':0})

dataset = pd.get_dummies(dataset, columns=onehot_columns, 
drop_first=True)

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58592 entries, 0 to 58591
Data columns (total 72 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   policy_id                                   58592 non-null  object 
 1   policy_age                                  58592 non-null  int64  
 2   age_of_car                                  58592 non-null  int64  
 3   age_of_policyholder                         58592 non-null  int64  
 4   area                                        58592 non-null  object 
 5   population_density                          58592 non-null  int64  
```

```
6   max_torque                             58592 non-null  float64
7   max_power                              58592 non-null  float64
8   airbags                                58592 non-null  int64
9   is_esc                                 58592 non-null  int64
10  is_adjustable_steering                 58592 non-null  int64
11  is_tpms                                58592 non-null  int64
12  is_parking_sensors                     58592 non-null  int64
13  is_parking_camera                      58592 non-null  int64
14  displacement                           58592 non-null  float64
15  cylinder                               58592 non-null  int64
16  gear_box                               58592 non-null  int64
17  turning_radius                         58592 non-null  float64
18  length                                 58592 non-null  float64
19  width                                  58592 non-null  float64
20  height                                 58592 non-null  float64
21  gross_weight                           58592 non-null  float64
22  is_front_fog_lights                    58592 non-null  int64
23  is_rear_window_wiper                   58592 non-null  int64
24  is_rear_window_washer                  58592 non-null  int64
25  is_rear_window_defogger                58592 non-null  int64
26  is_brake_assist                        58592 non-null  int64
27  is_power_door_locks                    58592 non-null  int64
28  is_central_locking                     58592 non-null  int64
29  is_power_steering                      58592 non-null  int64
30  is_driver_seat_height_adjustable       58592 non-null  int64
31  is_day_night_rear_view_mirror          58592 non-null  int64
32  is_ecw                                 58592 non-null  int64
33  is_speed_alert                         58592 non-null  int64
34  ncap_rating                            58592 non-null  int64
35  claims_in_5_years                      58592 non-null  int64
36  claim                                  58592 non-null  float64
37  make_Ford                              58592 non-null  bool
38  make_GM                                58592 non-null  bool
39  make_Honda                             58592 non-null  bool
40  make_Toyota                            58592 non-null  bool
41  segment_B1                             58592 non-null  bool
42  segment_B2                             58592 non-null  bool
43  segment_C1                             58592 non-null  bool
44  segment_C2                             58592 non-null  bool
45  segment_Utility                        58592 non-null  bool
46  model_M10                              58592 non-null  bool
47  model_M11                              58592 non-null  bool
48  model_M2                               58592 non-null  bool
49  model_M3                               58592 non-null  bool
50  model_M4                               58592 non-null  bool
51  model_M5                               58592 non-null  bool
52  model_M6                               58592 non-null  bool
53  model_M7                               58592 non-null  bool
54  model_M8                               58592 non-null  bool
```

```
 55  model_M9                               58592 non-null  bool
 56  fuel_type_Diesel                       58592 non-null  bool
 57  fuel_type_Petrol                       58592 non-null  bool
 58  engine_type_1.2 L K Series Engine      58592 non-null  bool
 59  engine_type_1.2 L K12N Dualjet         58592 non-null  bool
 60  engine_type_1.5 L U2 CRDi              58592 non-null  bool
 61  engine_type_1.5 Turbocharged Revotorq  58592 non-null  bool
 62  engine_type_1.5 Turbocharged Revotron  58592 non-null  bool
 63  engine_type_F8D Petrol Engine          58592 non-null  bool
 64  engine_type_G12B                       58592 non-null  bool
 65  engine_type_K Series Dual jet          58592 non-null  bool
 66  engine_type_K10C                       58592 non-null  bool
 67  engine_type_i-DTEC                     58592 non-null  bool
 68  rear_brakes_type_Drum                  58592 non-null  bool
 69  transmission_type_Manual               58592 non-null  bool
 70  steering_type_Manual                   58592 non-null  bool
 71  steering_type_Power                    58592 non-null  bool
dtypes: bool(35), float64(9), int64(26), object(2)
memory usage: 18.5+ MB
```

```
continuous_features = ['population_density','max_power',
'claims_in_5_years', 'displacement',
                       'gross_weight', 'height', 'width', 'length',
'max_torque']

plt.figure(figsize=(15, 10))

for i, col in enumerate(continuous_features, 1):
    plt.subplot(3, 3, i)
    sns.histplot(dataset[col], kde=True, bins=10, color='#246D82')
    plt.title(f"Distribution of {col}")

plt.tight_layout()
plt.show()
```


Distribution of population_density

Distribution of max_power

Distribution of claims_in_5_years

Distribution of displacement

Distribution of gross_weight

Distribution of height

Distribution of width

Distribution of length

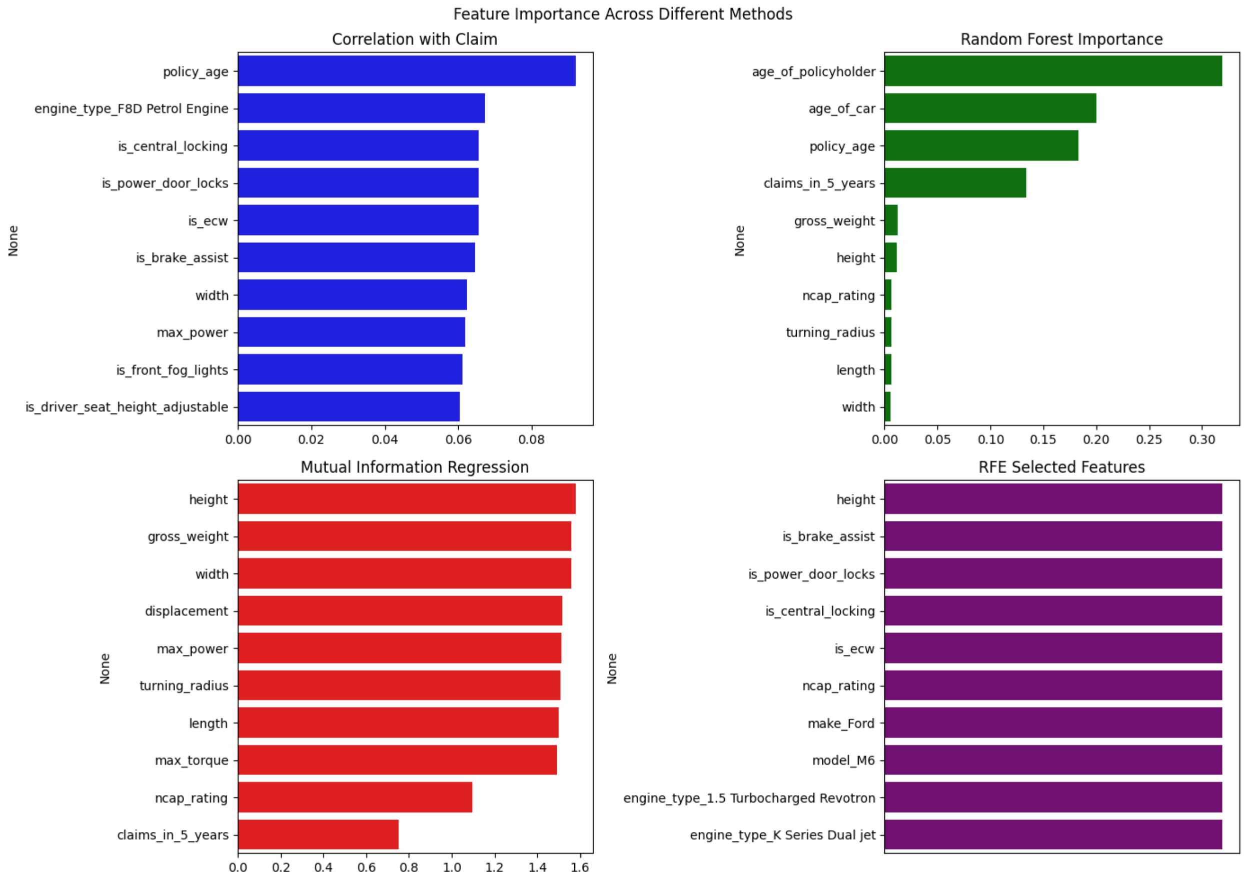
Distribution of max_torque

```
log_transformer = FunctionTransformer(np.log1p, validate=True)
scaler = StandardScaler()

dataset['max_power'] =
log_transformer.fit_transform(dataset[['max_power']])
dataset['displacement'] =
log_transformer.fit_transform(dataset[['displacement']])

for col in ['gross_weight', 'height', 'width', 'length',
'max_torque']:
    dataset[col] = scaler.fit_transform(dataset[[col]])

dataset.head()
```

{"type":"dataframe","variable_name":"dataset"}

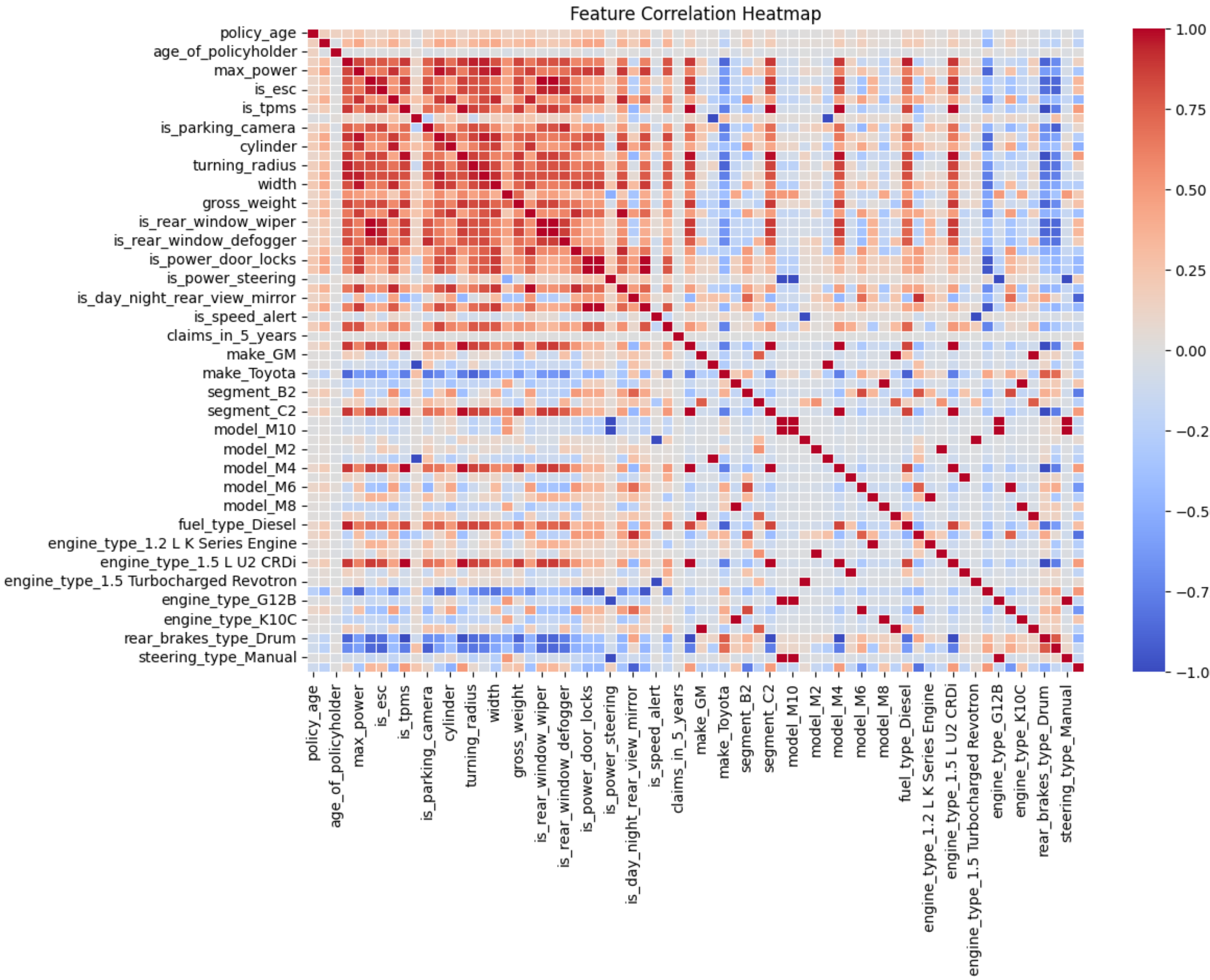
```
dataset2=dataset.copy()

dataset = dataset.drop(columns=['policy_id'])

X = dataset.drop(columns=['claim','population_density','area'])
y = dataset['claim']

corr_matrix =
dataset.drop(columns=['population_density','area']).corr()
['claim'].abs().sort_values(ascending=False)[1:11]
```

```
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X, y)
rf_importances = pd.Series(rf.feature_importances_,
index=X.columns).sort_values(ascending=False).head(10)

mi_importances = pd.Series(mutual_info_regression(X, y,
random_state=42),
index=X.columns).sort_values(ascending=False).head(10)

rfe = RFE(estimator=LinearRegression(), n_features_to_select=10)
rfe.fit(X, y)
rfe_importances = pd.Series(rfe.support_, index=X.columns)
rfe_selected_features = rfe_importances[rfe_importances == True].index

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
fig.suptitle('Feature Importance Across Different Methods')

sns.barplot(x=corr_matrix.values, y=corr_matrix.index, ax=axes[0, 0],
color='blue')
axes[0, 0].set_title('Correlation with Claim')

sns.barplot(x=rf_importances.values, y=rf_importances.index,
ax=axes[0, 1], color='green')
axes[0, 1].set_title('Random Forest Importance')

sns.barplot(x=mi_importances.values, y=mi_importances.index,
ax=axes[1, 0], color='red')
axes[1, 0].set_title('Mutual Information Regression')

sns.barplot(x=[1] * len(rfe_selected_features),
y=rfe_selected_features, ax=axes[1, 1], color='purple')
axes[1, 1].set_title('RFE Selected Features')
axes[1, 1].set_xticks([])

plt.tight_layout()
plt.show()
```



```
mi_importances

height               1.578313
gross_weight         1.558201
width                1.556319
displacement         1.514891
max_power            1.510224
turning_radius       1.508697
length               1.497975
max_torque           1.492158
ncap_rating          1.097225
claims_in_5_years    0.750461
dtype: float64

rf_importances

age_of_policyholder    0.319349
age_of_car             0.200270
policy_age             0.183155
claims_in_5_years      0.134020
gross_weight           0.012498
height                 0.011694
ncap_rating            0.006433
```

```
turning_radius           0.006429
length                   0.006161
width                    0.005678
dtype: float64

corr_matrix =
dataset.drop(columns=['population_density','area','claim']).corr()

plt.figure(figsize=(12, 8))

sns.heatmap(corr_matrix, fmt=".2f", cmap="coolwarm", linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()
```

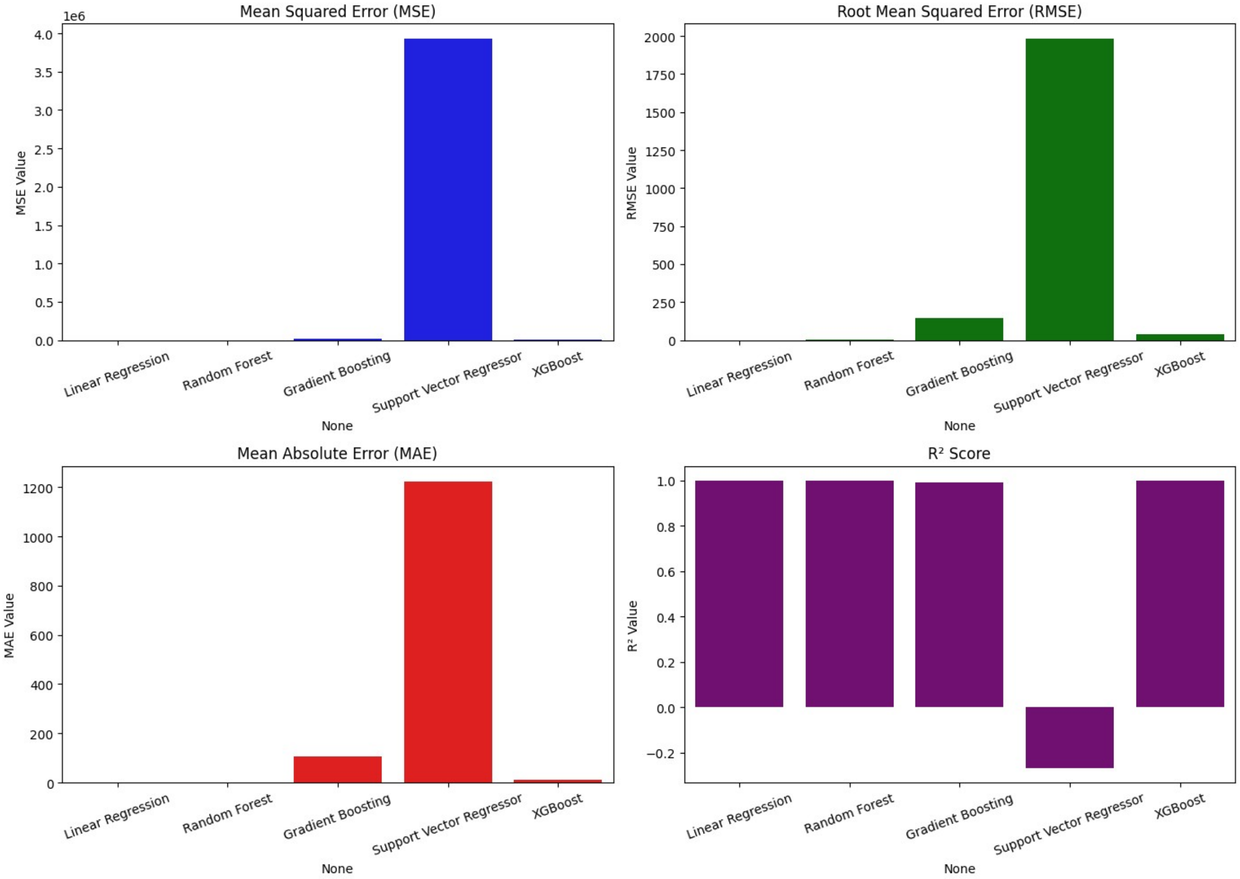


```
important_features = ['max_power',
  'is_ecw',
  'is_front_fog_lights',
  'claims_in_5_years',
  'gross_weight',
  'policy_age',
  'ncap_rating',
  'is_power_steering',
  'height',
  'width',
  'age_of_policyholder',
  'is_brake_assist',
  'displacement',
  'max_torque',
  'is_driver_seat_height_adjustable',
  'is_central_locking',
  'length',
  "area_Boston",
  "area_Charlotte",
  "area_Chicago",
  "area_Columbus",
  "area_Dallas",
  "area_Denver",
  "area_Fort Worth",
  "area_Houston",
  "area_Indianapolis",
  "area_Jacksonville",
  "area_Los Angeles",
  "area_Nashville",
  "area_New York",
  "area_Philadelphia",
  "area_Phoenix",
  "area_San Antonio",
  "area_San Diego",
  "area_San Francisco",
  "area_San Jose",
  "area_Seattle",
  "area_Washington D.C."]

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58592 entries, 0 to 58591
Data columns (total 91 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   policy_age                             58592 non-null  int64
 1   age_of_car                             58592 non-null  int64
 2   age_of_policyholder                    58592 non-null  int64
 3   population_density                     58592 non-null  int64
 4   max_torque                             58592 non-null  float64
 5   max_power                              58592 non-null  float64
 6   airbags                                58592 non-null  int64
 7   is_esc                                 58592 non-null  int64
 8   is_adjustable_steering                 58592 non-null  int64
 9   is_tpms                                58592 non-null  int64
 10  is_parking_sensors                     58592 non-null  int64
 11  is_parking_camera                      58592 non-null  int64
 12  displacement                           58592 non-null  float64
 13  cylinder                               58592 non-null  int64
 14  gear_box                               58592 non-null  int64
 15  turning_radius                         58592 non-null  float64
 16  length                                 58592 non-null  float64
 17  width                                  58592 non-null  float64
 18  height                                 58592 non-null  float64
 19  gross_weight                           58592 non-null  float64
 20  is_front_fog_lights                    58592 non-null  int64
 21  is_rear_window_wiper                   58592 non-null  int64
 22  is_rear_window_washer                  58592 non-null  int64
 23  is_rear_window_defogger                58592 non-null  int64
 24  is_brake_assist                        58592 non-null  int64
 25  is_power_door_locks                    58592 non-null  int64
 26  is_central_locking                     58592 non-null  int64
 27  is_power_steering                      58592 non-null  int64
 28  is_driver_seat_height_adjustable       58592 non-null  int64
 29  is_day_night_rear_view_mirror          58592 non-null  int64
 30  is_ecw                                 58592 non-null  int64
 31  is_speed_alert                         58592 non-null  int64
 32  ncap_rating                            58592 non-null  int64
 33  claims_in_5_years                      58592 non-null  int64
 34  claim                                  58592 non-null  float64
 35  area_Boston                            58592 non-null  bool
 36  area_Charlotte                         58592 non-null  bool
 37  area_Chicago                           58592 non-null  bool
 38  area_Columbus                          58592 non-null  bool
 39  area_Dallas                            58592 non-null  bool
 40  area_Denver                            58592 non-null  bool
 41  area_Fort Worth                        58592 non-null  bool
 42  area_Houston                           58592 non-null  bool
 43  area_Indianapolis                      58592 non-null  bool
 44  area_Jacksonville                      58592 non-null  bool
 45  area_Los Angeles                       58592 non-null  bool
 46  area_Nashville                         58592 non-null  bool
 47  area_New York                          58592 non-null  bool
 48  area_Philadelphia                      58592 non-null  bool
 49  area_Phoenix                           58592 non-null  bool
 50  area_San Antonio                       58592 non-null  bool
 51  area_San Diego                         58592 non-null  bool
 52  area_San Francisco                     58592 non-null  bool
 53  area_San Jose                          58592 non-null  bool
 54  area_Seattle                           58592 non-null  bool
 55  area_Washington D.C.                   58592 non-null  bool
 56  make_Ford                              58592 non-null  bool
 57  make_GM                                58592 non-null  bool
 58  make_Honda                             58592 non-null  bool
 59  make_Toyota                            58592 non-null  bool
 60  segment_B1                             58592 non-null  bool
 61  segment_B2                             58592 non-null  bool
 62  segment_C1                             58592 non-null  bool
 63  segment_C2                             58592 non-null  bool
 64  segment_Utility                        58592 non-null  bool
 65  model_M10                              58592 non-null  bool
 66  model_M11                              58592 non-null  bool
 67  model_M2                               58592 non-null  bool
 68  model_M3                               58592 non-null  bool
 69  model_M4                               58592 non-null  bool
 70  model_M5                               58592 non-null  bool
 71  model_M6                               58592 non-null  bool
 72  model_M7                               58592 non-null  bool
 73  model_M8                               58592 non-null  bool
 74  model_M9                               58592 non-null  bool
 75  fuel_type_Diesel                       58592 non-null  bool
 76  fuel_type_Petrol                       58592 non-null  bool
 77  engine_type_1.2 L K Series Engine      58592 non-null  bool
 78  engine_type_1.2 L K12N Dualjet         58592 non-null  bool
 79  engine_type_1.5 L U2 CRDi              58592 non-null  bool
 80  engine_type_1.5 Turbocharged Revotorq  58592 non-null  bool
 81  engine_type_1.5 Turbocharged Revotron  58592 non-null  bool
 82  engine_type_F8D Petrol Engine          58592 non-null  bool
 83  engine_type_G12B                       58592 non-null  bool
 84  engine_type_K Series Dual jet          58592 non-null  bool
 85  engine_type_K10C                       58592 non-null  bool
 86  engine_type_i-DTEC                     58592 non-null  bool
 87  rear_brakes_type_Drum                  58592 non-null  bool
 88  transmission_type_Manual               58592 non-null  bool
 89  steering_type_Manual                   58592 non-null  bool
 90  steering_type_Power                    58592 non-null  bool
dtypes: bool(56), float64(9), int64(26)
memory usage: 18.8 MB

X = dataset[important_features]
y = dataset['claim']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

models = {
    "Linear Regression": LinearRegression(),
    "Random Forest": RandomForestRegressor(n_estimators=100,
random_state=42),
    "Gradient Boosting": GradientBoostingRegressor(n_estimators=100,
random_state=42),
    "Support Vector Regressor": SVR(),
    "XGBoost": XGBRegressor(n_estimators=100, random_state=42)
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    results[name] = {"MSE": mse, "RMSE": rmse, "MAE": mae, "R² Score":
r2}

results_df = pd.DataFrame(results).T

fig, axes = plt.subplots(2, 2, figsize=(14, 10))

sns.barplot(x=results_df.index, y=results_df['MSE'], ax=axes[0, 0],
color='blue')
axes[0, 0].set_title('Mean Squared Error (MSE)')
axes[0, 0].set_ylabel('MSE Value')
axes[0, 0].set_xticklabels(results_df.index, rotation=20)

sns.barplot(x=results_df.index, y=results_df['RMSE'], ax=axes[0, 1],
color='green')
axes[0, 1].set_title('Root Mean Squared Error (RMSE)')
axes[0, 1].set_ylabel('RMSE Value')
axes[0, 1].set_xticklabels(results_df.index, rotation=20)

sns.barplot(x=results_df.index, y=results_df['MAE'], ax=axes[1, 0],
color='red')
axes[1, 0].set_title('Mean Absolute Error (MAE)')
axes[1, 0].set_ylabel('MAE Value')
axes[1, 0].set_xticklabels(results_df.index, rotation=20)

sns.barplot(x=results_df.index, y=results_df['R² Score'], ax=axes[1,
1], color='purple')
axes[1, 1].set_title('R² Score')
axes[1, 1].set_ylabel('R² Value')
axes[1, 1].set_xticklabels(results_df.index, rotation=20)

plt.tight_layout()
plt.show()
```

```
<ipython-input-34-e95f68489882>:7: UserWarning: set_ticklabels()
should only be used with a fixed number of ticks, i.e. after
set_ticks() or using a FixedLocator.
  axes[0, 0].set_xticklabels(results_df.index, rotation=20)
```

```
<ipython-input-34-e95f68489882>:13: UserWarning: set_ticklabels()
should only be used with a fixed number of ticks, i.e. after
set_ticks() or using a FixedLocator.
  axes[0, 1].set_xticklabels(results_df.index, rotation=20)
<ipython-input-34-e95f68489882>:19: UserWarning: set_ticklabels()
should only be used with a fixed number of ticks, i.e. after
set_ticks() or using a FixedLocator.
  axes[1, 0].set_xticklabels(results_df.index, rotation=20)
<ipython-input-34-e95f68489882>:25: UserWarning: set_ticklabels()
should only be used with a fixed number of ticks, i.e. after
set_ticks() or using a FixedLocator.
  axes[1, 1].set_xticklabels(results_df.index, rotation=20)
```



```
results_df
```

```
{"summary":"{\n  \"name\": \"results_df\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"MSE\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1755024.4142618526,\n        \"min\": 0.0692928672673422,\n
\"max\": 3929820.6522851484,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          1.0316953587004944,\n
1379.9056242508052,\n          20682.220268141984\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"RMSE\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 868.1547643785032,\n
\"min\": 0.2632353837677264,\n        \"max\": 1982.3775251664724,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
1.0157240563757928,\n          37.147080965411064,\n
143.8131435861896\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"MAE\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 536.3139785710244,\n        \"min\": 0.2070779678762725,\n
\"max\": 1224.0919539652052,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.2530196136192616,\n
10.50190264170198,\n          104.56140340786972\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"R\\u00b2 Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.5664842536290661,\n        \"min\": -0.2684618521629818,\n
\"max\": 0.999999977633748,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.9999996669908575,\n
0.9995545960493313,\n          0.9933242227191933\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"results_df"}
```