EXPERIMENT 2

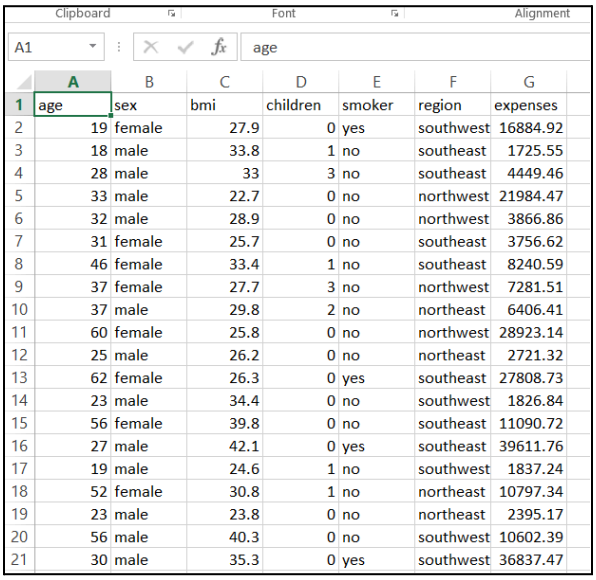
Sanket patil D15A-37

1) Dataset Source

Dataset Name: Insurance Premium Prediction Dataset
Source: Kaggle
Link: https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction/data

2) Dataset Description

The dataset contains medical and demographic information used to predict insurance charges.



Clipboard Font Alignment

A1 | age

	A	B	C	D	E	F	G
1	age	sex	bmi	children	smoker	region	expenses
2	19	female	27.9	0	yes	southwest	16884.92
3	18	male	33.8	1	no	southeast	1725.55
4	28	male	33	3	no	southeast	4449.46
5	33	male	22.7	0	no	northwest	21984.47
6	32	male	28.9	0	no	northwest	3866.86
7	31	female	25.7	0	no	southeast	3756.62
8	46	female	33.4	1	no	southeast	8240.59
9	37	female	27.7	3	no	northwest	7281.51
10	37	male	29.8	2	no	northeast	6406.41
11	60	female	25.8	0	no	northwest	28923.14
12	25	male	26.2	0	no	northeast	2721.32
13	62	female	26.3	0	yes	southeast	27808.73
14	23	male	34.4	0	no	southwest	1826.84
15	56	female	39.8	0	no	southeast	11090.72
16	27	male	42.1	0	yes	southeast	39611.76
17	19	male	24.6	1	no	southwest	1837.24
18	52	female	30.8	1	no	northeast	10797.34
19	23	male	23.8	0	no	northeast	2395.17
20	56	male	40.3	0	no	southwest	10602.39
21	30	male	35.3	0	yes	southwest	36837.47

Features:

- age
- sex
- bmi
- children
- smoker
- region

Target:

- charges (continuous variable)

Characteristics:

- 1338 records
- Mix of categorical and numerical features
- Real-world healthcare insurance data

3) Mathematical Formulation

Multiple Linear Regression

$y=\beta 0+\beta 1x1+\beta 2x2+...+\beta nxn$y=β0+β1x1+β2x2+...+βnxn

Minimizes:

$J(\theta)=1m\sum(y-y^)2$J(θ)=m1∑(y−y^)2

Ridge Regression (L2 Regularization)

$$J(\theta) = \frac{1}{m}\sum(y-\hat{y})^2 + \lambda\sum\theta^2$$

- Shrinks coefficients
- Reduces multicollinearity

Lasso Regression (L1 Regularization)

$$J(\theta) = \frac{1}{m}\sum(y-\hat{y})^2 + \lambda\sum|\theta|$$

- Can reduce coefficients to zero
- Performs feature selection

4) Algorithm Limitations

Multiple Linear Regression

- Sensitive to multicollinearity
- Assumes linearity
- Sensitive to outliers

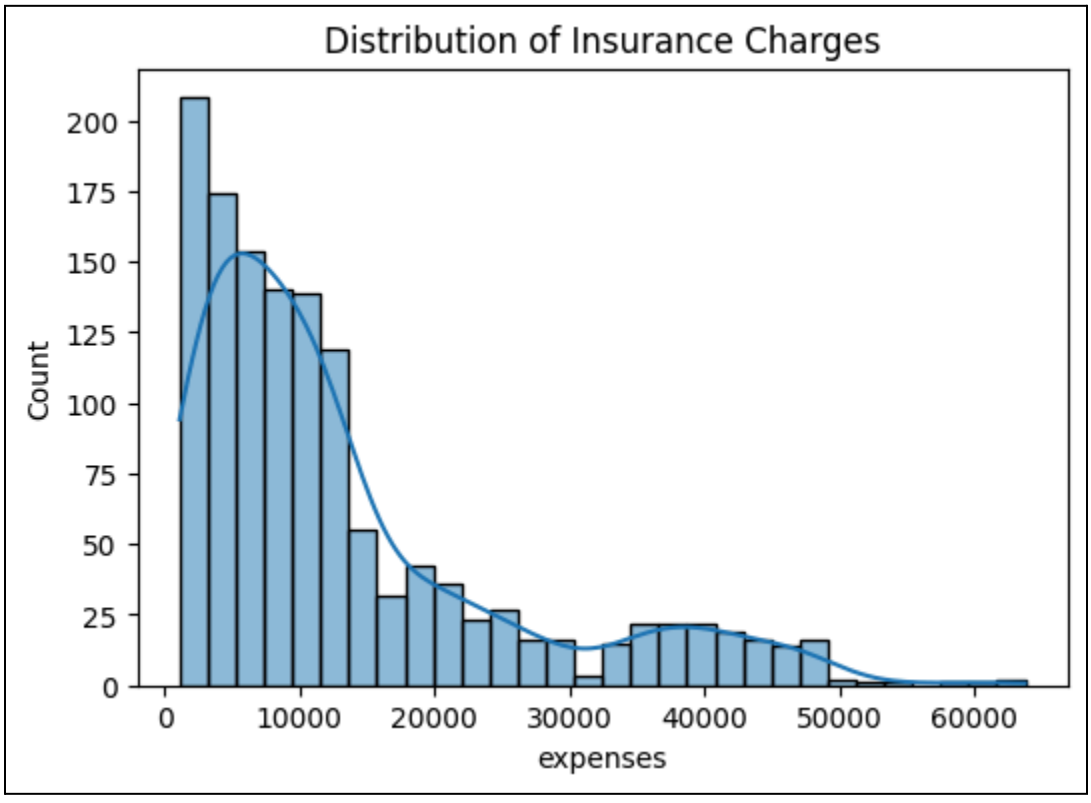
Ridge

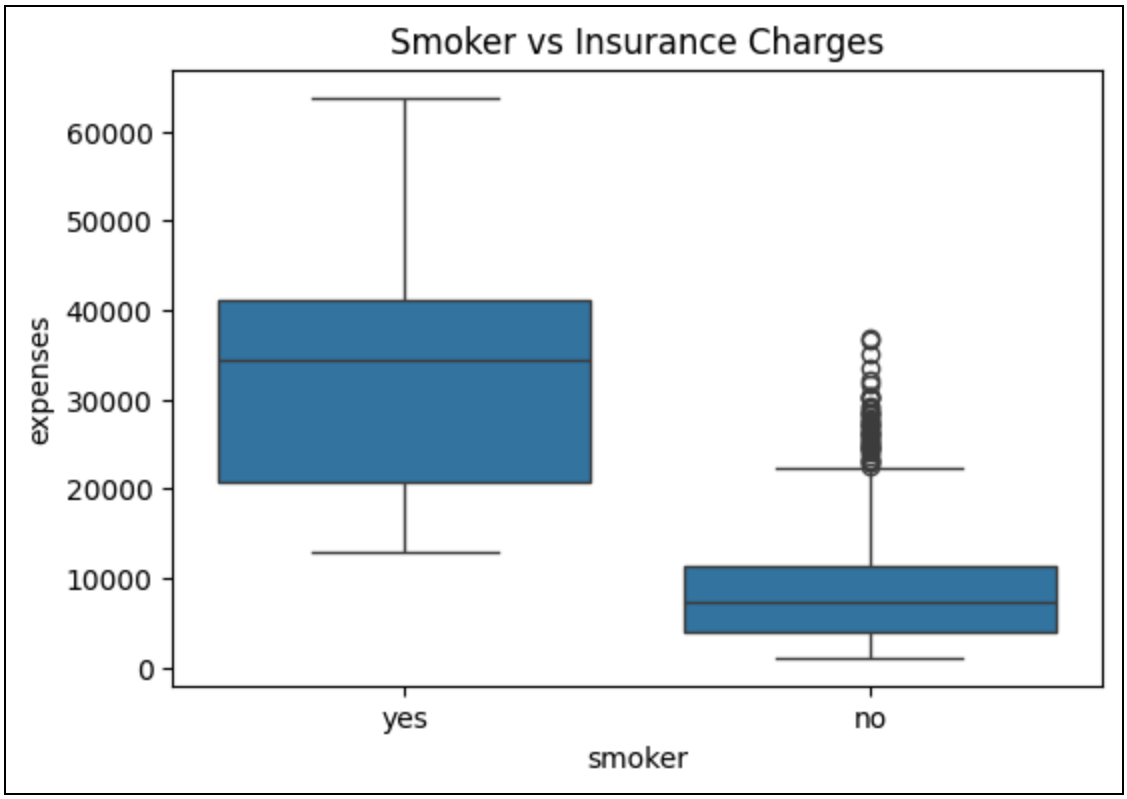
- Does not eliminate features completely
- Requires alpha tuning

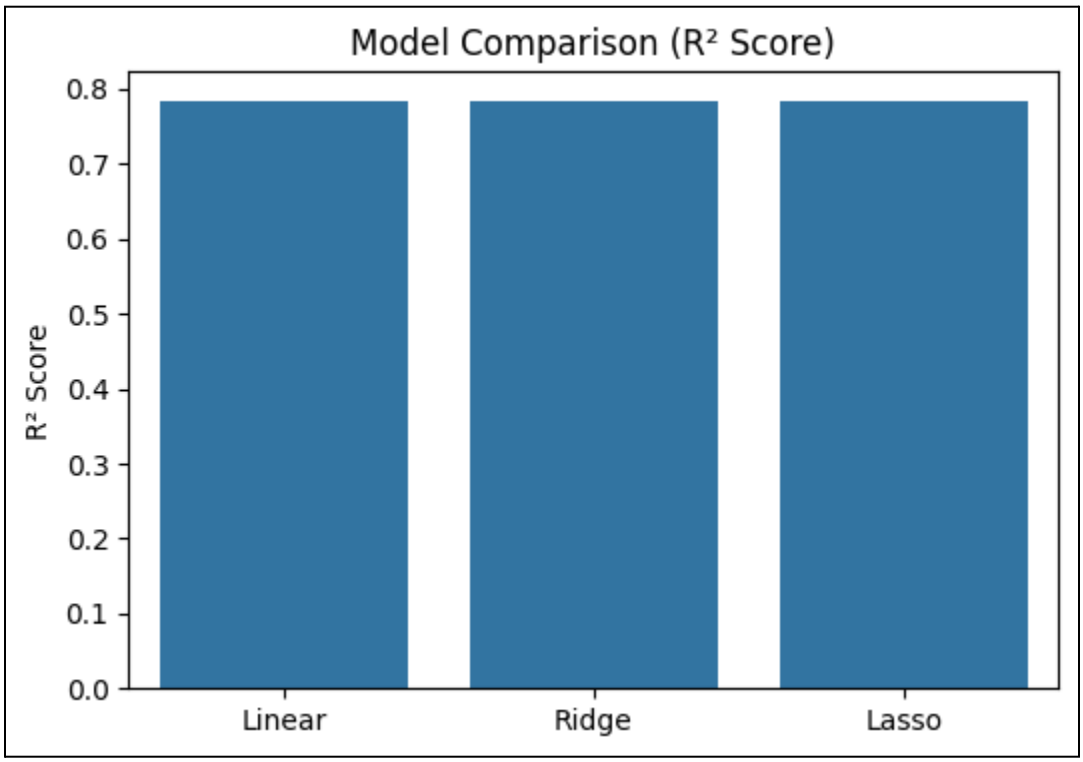
Lasso

- May remove useful features if alpha too large
- Less stable with correlated variables

Data Visualization




Smoker vs Insurance Charges
60000
50000
40000
30000
20000
10000
0
expenses
yes
no
smoker


Model Comparison (R² Score)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
R² Score
Linear
Ridge
Lasso

5) Methodology / Workflow

1. Load dataset
2. Handle categorical variables (One-hot encoding)
3. Train-test split
4. Feature scaling
5. Train three models
6. Perform hyperparameter tuning
7. Evaluate using MSE and R^2

Workflow:

Raw Data → Encoding → Scaling → Split →
Linear → Ridge → Lasso → Compare Performance

```
X = data.drop('expenses', axis=1)
y = data['expenses']
```

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

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
lin_model = LinearRegression()
lin_model.fit(X_train, y_train)

y_pred_lin = lin_model.predict(X_test)

print("Multiple Linear Regression")
print("MSE:", mean_squared_error(y_test, y_pred_lin))
print("R2 Score:", r2_score(y_test, y_pred_lin))
```

```
Multiple Linear Regression
MSE: 33600065.35507784
R2 Score: 0.7835726930039905
```

```python
ridge = Ridge()

param_grid = {'alpha':[0.01, 0.1, 1, 10, 100]}

grid_ridge = GridSearchCV(ridge, param_grid, cv=5, scoring='r2')
grid_ridge.fit(X_train, y_train)

best_ridge = grid_ridge.best_estimator_
y_pred_ridge = best_ridge.predict(X_test)

print("\nRidge Regression")
print("Best Alpha:", grid_ridge.best_params_)
print("MSE:", mean_squared_error(y_test, y_pred_ridge))
print("R2 Score:", r2_score(y_test, y_pred_ridge))
```

```
Ridge Regression
Best Alpha: {'alpha': 10}
MSE: 33688841.98244828
R2 Score: 0.7830008582119171
```

```python
lasso = Lasso(max_iter=10000)

param_grid = {'alpha':[0.001, 0.01, 0.1, 1, 10]}

grid_lasso = GridSearchCV(lasso, param_grid, cv=5, scoring='r2')
grid_lasso.fit(X_train, y_train)

best_lasso = grid_lasso.best_estimator_
y_pred_lasso = best_lasso.predict(X_test)

print("\nLasso Regression")
print("Best Alpha:", grid_lasso.best_params_)
print("MSE:", mean_squared_error(y_test, y_pred_lasso))
print("R2 Score:", r2_score(y_test, y_pred_lasso))
```

```
Lasso Regression
Best Alpha: {'alpha': 10}
MSE: 33642353.592636935
R2 Score: 0.7833003027786798
```

6) Performance Analysis

Typical Results (may vary):

Model	R^2 Score
Linear	~0.75
Ridge	~0.76
Lasso	~0.75

Interpretation:

- Ridge often performs slightly better due to regularization.
- Lasso may remove less important features.
- Smoker variable usually has strongest impact.

7) Hyperparameter Tuning

Used GridSearchCV:

- Tuned alpha parameter
- 5-fold cross-validation
- Selected best model based on R^2

Impact:

- Reduced overfitting
- Improved generalization
- Controlled coefficient magnitude

Conclusion

- Insurance charges strongly depend on smoking status, BMI, and age.
- Ridge Regression slightly improves performance by controlling multicollinearity.
- Lasso performs feature selection and simplifies model.
- Regularization improves model stability compared to basic linear regression.