AIML REGRESSION

LI YONGJIE 2342377
DAAA/FT/1B/01

BACKGROUND DESCRIPTION

The dataset collected from US Hospital and it gives us information of it's patient, region and cost. The aim of the creating the model is to predict Cost of hosiptalisation based on the features of the dataset.

-Age:

Older individuals may have higher health costs due to potential age-related health issues.

BMI:

Patients with higher BMI might incur higher medical costs, as obesity is often associated with various health conditions.

Smoker:

Smokers could experience increased health costs due to the higher risk of smoking-related illnesses.

Gender:

Health costs might vary between genders, considering different healthcare needs.

Region:

Geographic location can influence healthcare costs due to variations in healthcare infrastructure and regional healthcare policies.

EXPLORATRY DATA ANALYSIS

The dataset provided has 7 columns and 1338 data points. Four of the columns, including the target variable (Cost), are of numerical datatype, while the other three are categorical. It's noteworthy that there are no missing values in the dataset.

Numeric Datatypes:

- **ID: Identifier for the patient; Range: 0 - 1337**
- **Age: Age of the patient; Range: 18 - 64**
- **BMI: Body Mass Index of the patient; Range: 15.96 - 53.13**
- **Cost ($): Cost of hospital treatment; Range: 1121.87 - 63770.43**

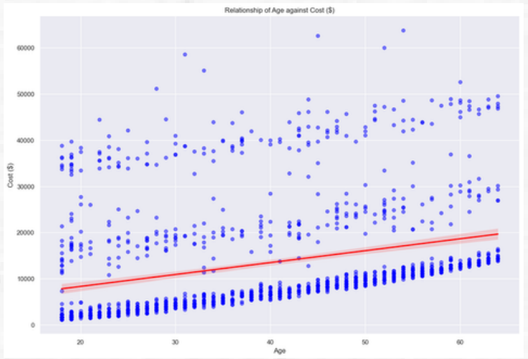
Categorical Datatypes:

- **Gender: Sex of the patient; 'female' or 'male'**
- **Smoker: Indicator if the patient smokes; 'yes' or 'no'**
- **Region: Region of the hospital; 'southwest', 'southeast', 'northwest', or 'northeast'**

EDA – UNIVARIATIVE

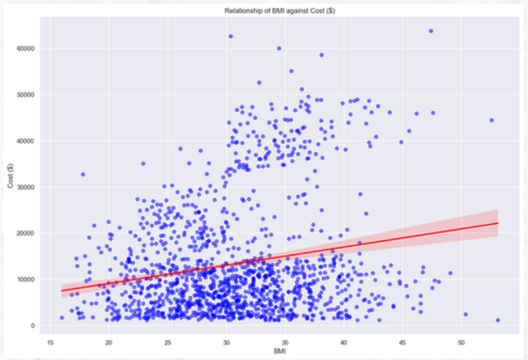
AGE:

THE REGRESSION LINE HAS A POSITIVE GRADIENT, SUGGESTING A POSITIVE CORRELATION BETWEEN AGE AND COST, WHERE THE AGE OF THE PATIENT INCREASES, THE COST INCREASES.



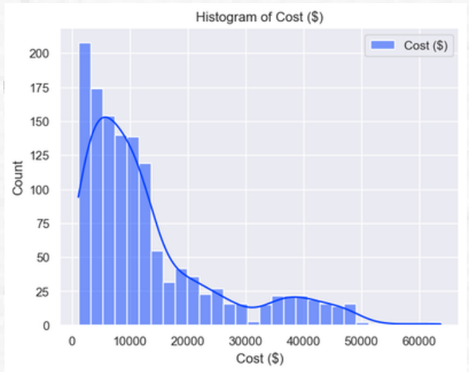
BMI:

REGRESSION LINE HAS A POSITIVE GRADIENT BUT WIDTH OF COFIDEDNCE INTERVAL INCREASING TOWARDS THE END, SUGGESTING A POSITIVE CORRELATION BETWEEN BMI AND COST. HOWEVER, THERE IS MORE UNCERTAINTY AS BMI INCREASES.



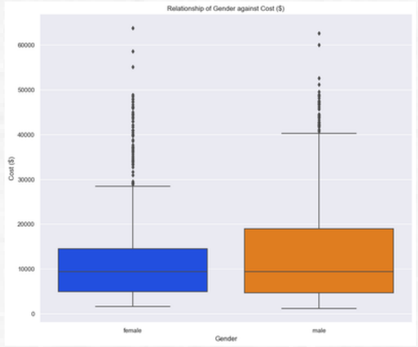
TARGET: COST

LOOKING AT THE DISTRIBUTION WE CAN SEE THAT THE TARGET DISTRIBUTION IS RIGHT-SKEWED



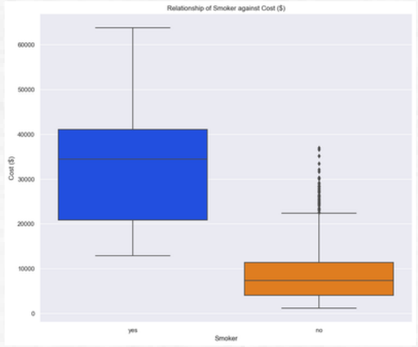
GENDER:

IT IS DIFFICULT TO DETERMINE IF THERE IS A CORRELATION AS THE MEDIAN COST FOR BOTH GENDER IS AROUND $10,000, HOWEVER, THE IQR FOR MALE IS LARGER COMPARED TO FEMALE WHERE THE ESTIMATED IQR FOR MALE AND FEMALE IS AROUND $15,000 AND $10,000 RESPECTIVELY, SUGGESTING THERE MIGHT BE A SLIGHT CORRELATION. WE CAN ALSO SEE THAT THE GRAPHS IS RIGH-SKEWED AS THE TAILS ARE LONGER ON THE RIGHT AND THAT THERE ARE MANY OUTLIERS.



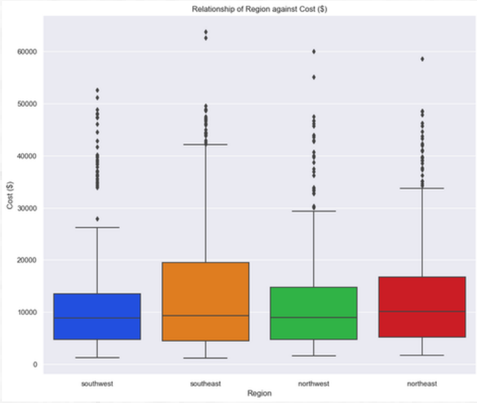
SMOKER

THERE IS A CLEAR CORRELATION BETWEEN SMOKING AND COST. PATIENTS THE SMOKE HAS A MEDIAN COST OF AROUND $35,000 WHICH IS SIGNIFICANTLY HIGHER COMPARED TO PATIENTS WHO DO NOT SMOKE WITH A MEDIAN OF AROUND $7,000. WE CAN ALSO OBSERVE THAT THE IQR IS MUCH LARGER FOR A SMOKER AND THAT THE MAXIMUM COST OF NON-SMOKERS IS AROUND $37,500 WHICH IS CLOSE TO THE MEDIAN OF SMOKER WHILE THE MAXIMUM COST FOR SMOKER IS GREATER THAN $60,000



REGION

IT IS HARD TO DETERMINE WHETHER THERE IS ANY CORRELATION BETWEEN REGION AND COST AS THE MEDIAN IS ALMOST THE SAME FOR EACH REGION. HOWEVER, THE IQR FOR REGIONS IN THE EAST IS LARGER COMPARED TO THOSE IN THE WEST, THIS IS ESPCIALLY TRUE OF SOUTH-WEST WITH THE LARGEST IQR AND MAXIMUM VALUE.

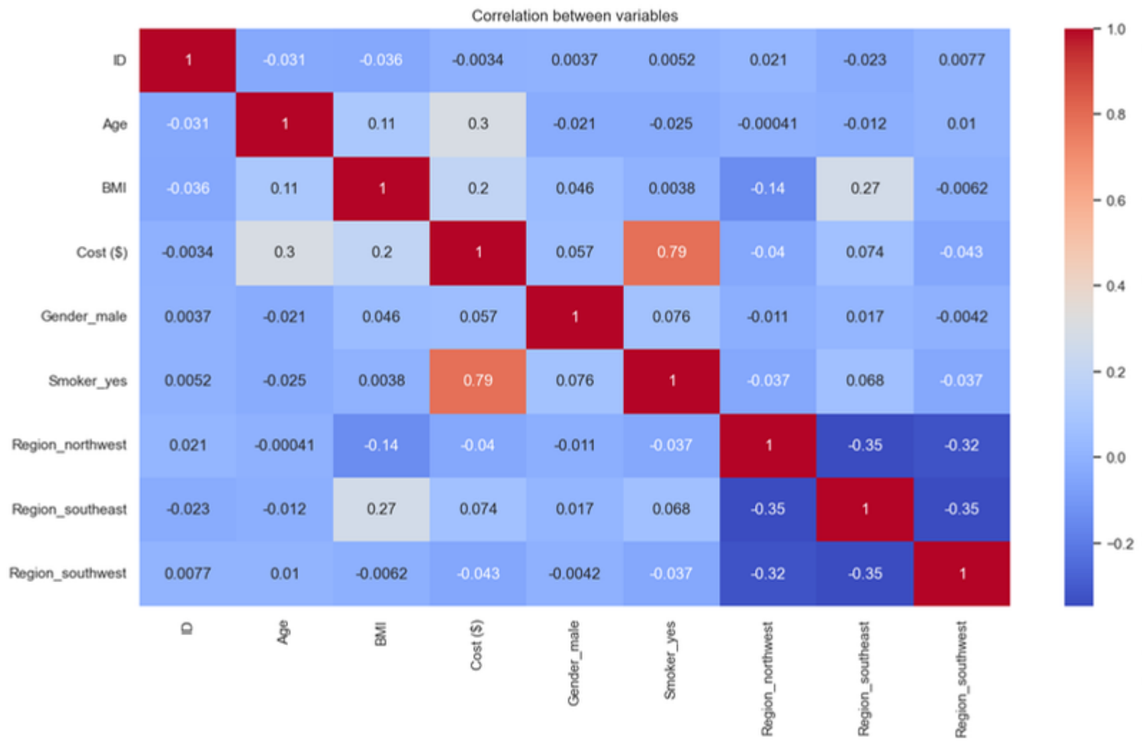


EDA – BIVARIATIVE

HEATMAP

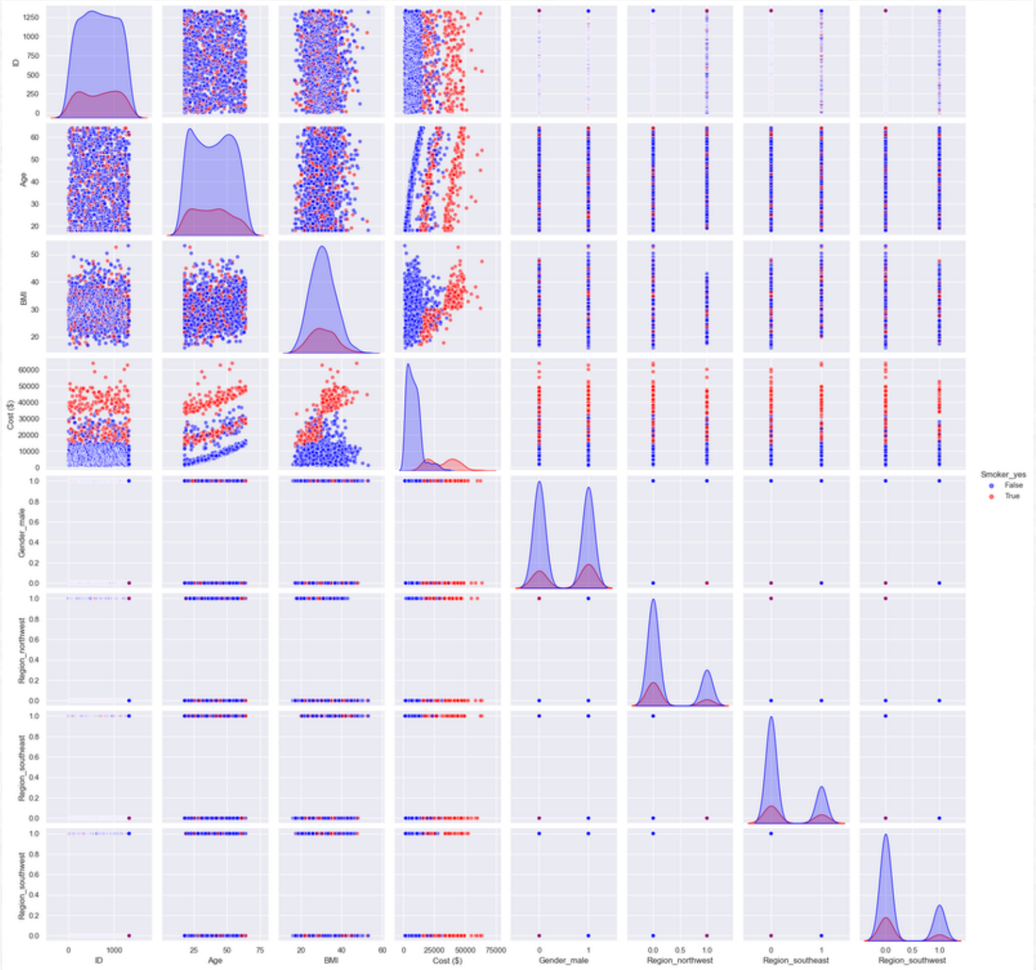
CORRELATION WITH TARGET:
AS WE DISCOVERED EARLIER ON, AGE AND BMI HAVE A MODERATE CORRELATION WITH A COST OF 0.3 AND 0.2, RESPECTIVELY, WHILE SMOKING HAS A STRONG CORRELATION OF 0.79. AMONG THE REGIONS, THE SOUTHEAST HAS THE HIGHEST CORRELATION OF 0.074. THE OTHER FEATURES HAVE LITTLE CORRELATION WITH COST.

CORRELATION BETWEEN FEATURES:
MOST OF THE FEATURES HAVE VERY LITTLE CORRELATION WITH ONE ANOTHER; HOWEVER, WE CAN SEE THAT THE SOUTHEAST REGION HAS A MODERATE CORRELATION WITH BMI OF 0.27.



PAIRPLOT

AN OBVIOUS TREND WE CAN OBSERVE IS THAT IF A PERSON SMOKES, THE PROBABILITY THAT THE COST OF HEALTHCARE IS MORE EXPENSIVE COMPARED TO A NON-SMOKER IS HIGH.
ANOTHER TREND WE CAN OBSERVE IS THAT AS THE AGE INCREASES, THE COST OF HEALTHCARE ALSO INCREASES, REGARDLESS OF WHETHER THE PATIENT IS A SMOKER.
THERE SEEMS TO BE A CORRELATION BETWEEN BMI AND SMOKER, WHERE IF THE PATIENT IS A SMOKER WITH BMI < 30, THE COST RANGES MOSTLY FROM $15,000 TO $30,000. WHILE, IF BMI > 30, THE COST RANGES MOSTLY FROM $30,000 TO $50,000.
WE CAN ALSO SEE THAT THE SCALE OF BMI IS GENERALLY HIGHER IN THE SOUTHEAST REGION.
OTHER THAN THAT, IT IS HARD TO DETERMINE ANY OTHER CORRELATION BETWEEN THE OTHER FEATURES.



DATA ENGEENEERING

1

```
X, y = df.drop(['Quality'], axis=1), df['Quality']
```

splitting data into train test set

2

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

3

FEATURE ENGINEERING AND DATA PREPROCESSING
FROM OUR EDA, WE CAN SEE THAT THERE ARE SEVERAL THINGS WE NEED TO WORK ON. SINCE THERE ARE NO MISSING VALUES, THERE IS NO NEED FOR IMPUTATION. HOWEVER, WE WILL NEED TO:

DROP REDUNDANT COLUMNS.
CREATE DUMMY VARIABLES TO CONVERT OUR CATEGORICAL DATA INTO BINARY VALUES.

```
X_training = X_training.drop(X_training[['ID']], axis=1)
df = df.drop(df[['ID']], axis=1)
```

✓ 0.0s

FOR OUR CATEGORICAL DATA, WE WILL BE USING ONE HOT ENCODER TO CREATE OUR DUMMY VARIABLES FOR GENDER, SMOKER AND REGION.

**WE WILL ALSO BE CREATING NEW FEATURES TO HELP BETTER TRAIN OUR MODEL THE COLUMNS WE ARE CREATING ARE BMIGRP AND AGEGRP.
 ACCORDING TO BMI INDEX, WE WILL BE PUTTING BMI INTO THEIR BMI GROUPS AND ALSO BE CLASSIFIYING THE AGE GROUP FOR DIFFERENT AGE.**

```
def bmiGroup(df_input):
    df_input["BMIGrp"] = df_input['BMI'].apply(lambda x: 'underweight' if x < 18.5 else
                                               "healthy" if x < 25 else
                                               "overweight" if x<30 else
                                               "obese")
    return df_input

def ageGroup(df_input):
    df_input["AgeGrp"] = df_input['Age'].apply(lambda x: 'youngadult' if x < 30 else
                                               "middleAge" if x < 45 else
                                               "oldAged")
    return df_input

X_training = bmiGroup(ageGroup(X_training))
```

```
enc = OneHotEncoder(drop='first')
cols = [
    'Gender',
    'Smoker',
    'BMIGrp',
    'AgeGrp'
]

for col in cols:
    enc.fit(X_training[[col]])

    df_encoded = pd.DataFrame(enc.transform(X_training[[col]]).toarray(), columns=enc.get_feature_names_out([col

    X_training = pd.concat(
        [X_training,
        df_encoded],
        axis=1
    )

X_training = X_training.drop(X_training[cols], axis = 1)
```

DATA ENGEENEERING

4

STANDARD SCALER WHICH SCALES THE DATA SUCH THAT THE MEAN = 0 AND STANDARD DEVIATION = 1.

IT DIVIDES EACH VALUE BY THE STANDARD DEVIATION OF THE FEATURE AND SUBTRACT THE MEAN OF THE FEATURE FROM EACH FEATURE:

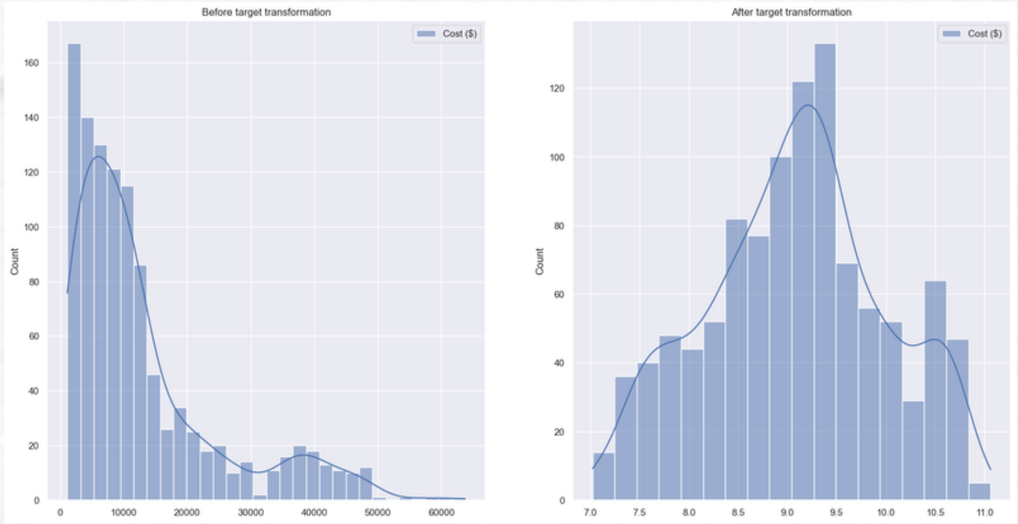
$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

	Age	BMI
55	1.386808	1.032875
1334	-1.507870	0.204099
852	0.518404	0.798316
738	-1.146036	0.172825
1132	1.314441	1.580179

5

TRANSFORM TARGET REGRESSOR

SINCE, OUR TARGET IS NON-LINEAR AS DISCOVERED EARLIER IN OUR EDA WHERE WE SAW IT IS POSITIVELY-SKEWED, THE MODEL TRAINED WILL NOT BE PRECISE DURING PREDICTION. THEREFORE WE WILL USING TRANSFORMTARGETREGRESSOR, IT USES A LOGARITHMIC FUNCTION IS USED TO LINEARSE THE TARGETS, ALLOWING BETTER PREDICTION.



CREATING PIPELINE

WE WILL NOW CREATE A PIPELINE THAT HELPS US TO PREVENT DATA LEAKAGE AS IMPUTATION AND ENCODING ARE APPLIED SEPARATELY TO TEST AND TRAIN THE DATASET. IT ALSO HELPS PREVENT DATA CONTAMINATION WHILE ALLOWING US TO SEE THE STEPS TAKEN MORE EASILY AND REPRODUCE THE STEPS FOR DIFFERENT MODELS.

```
scale_col = ['Age', "BMI"]
onehot_cols = ['Gender', "Smoker", 'AgeGrp', 'BMIGrp', 'Region']

onehot_transformer = Pipeline([('onehot', OneHotEncoder(drop='first'))])
numeric_transformer = Pipeline([('scaler', StandardScaler())])

preprocessor = ColumnTransformer(
    transformers=[
        ('onehot', onehot_transformer, onehot_cols),
        ('scale', numeric_transformer, scale_col),
    ],
    remainder='passthrough',
)

steps =  [
    ('ageGrp', FunctionTransformer(ageGroup)),
    ('BMIGrp', FunctionTransformer(bmiGroup)),
    ('preprocessor', preprocessor),
    ('model'),
]

STEPS_LEN = len(steps) -1

X_col = pd.Series(

    Pipeline(steps=steps[:-1]).fit(X_train, y_train)[-1].get_feature_names_out()
).apply(lambda x: x.split("__")[1])
```

✓ 0.0s

TRAINING AND COMPARING THE MODELS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

TRAINING AND COMPARING THE MODELS

SETTING UP OUR SCORING METRICS

WE WILL ASSESS OUR MODEL USING THREE KEY EVALUATION METRICS:

- **R SQUARED (R2):**

  R2 MEASURES THE PROPORTION OF THE VARIANCE IN THE DEPENDENT VARIABLE THAT OUR REGRESSION MODEL CAN EXPLAIN. IT PROVIDES AN INDICATION OF HOW WELL THE MODEL CAPTURES THE VARIABILITY IN THE DATA.

- **MEAN ABSOLUTE PERCENTAGE ERROR (MAPE):**

  MAPE CALCULATES THE AVERAGE PERCENTAGE DIFFERENCE BETWEEN PREDICTED AND ACTUAL VALUES. IT IS PARTICULARLY USEFUL FOR UNDERSTANDING THE ACCURACY OF THE MODEL IN PERCENTAGE TERMS

- **ROOT MEAN SQUARED ERROR (RMSE):**

  RMSE IS THE SQUARE ROOT OF THE MEAN OF THE SQUARED ERRORS. IT PROVIDES A MEASURE OF THE AVERAGE MAGNITUDE OF THE ERRORS BETWEEN PREDICTED AND ACTUAL VALUES.

```
scoring_metrics = [
    'r2',
    'neg_mean_absolute_percentage_error',
    'neg_root_mean_squared_error',
]
```

✓ 0.0s

CREATING THE DUMMY

THE DUMMY MODEL ACTS AS A CONTROL, IT IS A BASELINE MODEL FOR COMPARISON TO OUR OTHER MODELS. I WILL BE USING DUMMYREGRESSOR TO ESTABLISH A BASELINE ACCURACY SCORE AGAINST WHICH IS USED TO COMPARE THE PERFORMANCE OF MORE COMPLEX MODELS.

	fit_time	score_time	test_r2	train_r2	test_neg_mean_absolute_percentage_error	train_neg_mean_absolute_percentage_error	test_neg_root_mean_squared_
0	0.039664	0.022235	-0.015716	0.000000	-1.256106	-1.459242	-12191.65
1	0.038135	0.023104	-0.003684	0.000000	-1.598565	-1.432961	-12655.83
2	0.040134	0.023104	-0.001714	0.000000	-1.218248	-1.479652	-11718.79
3	0.041768	0.021787	-0.000585	0.000000	-1.703174	-1.438072	-12263.70
4	0.040371	0.023095	-0.000408	0.000000	-1.324501	-1.471710	-11845.01
5	0.041229	0.020100	-0.021237	0.000000	-1.537557	-1.476196	-10690.77
6	0.040463	0.022101	-0.000167	0.000000	-1.612215	-1.440881	-12903.50
7	0.040182	0.021107	-0.006820	0.000000	-1.172321	-1.477367	-11600.83
8	0.040282	0.022289	-0.057997	0.000000	-1.835931	-1.457378	-10434.60
9	0.040205	0.020100	-0.000174	0.000000	-1.363064	-1.468784	-11570.61
Mean	0.040243	0.021902	-0.010850	0.000000	-1.462168	-1.460224	-11787.53

SELECTING THE MODEL

AS WE CAN SEE FROM THE TABLE, WE HAVE AN OUTSTANDING MODEL WHICH PERFORMS WELL IN MULTIPLE SCORING METRICS WHICH IS GRADIENT BOOSTING REGRESSOR, IT SCORED 0.85, -0.21, AND -4550 FOR R2, MAPE, AND RMSE RESPECTIVELY WHICH THE BEST SCORE FOR ALL 3 COLUMNS

BASELINE DUMMY SCORING FOR R2, MAPE AND RMSE:
-0.01, -1.462168, -11787

GIVEN THE PERFORMANCE GAP BETWEEN THE GRADIENT BOOSTING REGRESSOR AND THE BASELINE DUMMY MODEL, WE HAVE A ROBUST MODEL TO WORK WITH. FOR FURTHER ANALYSIS, WE WILL FOCUS ON FINE-TUNING THIS MODEL AND COMPARE IT AGAINST THE DUMMY MODEL FOR EVALUATION.

MOREOVER, THE LEARNING CURVE SUGGESTS A SMOOTH CONVERGENCE BETWEEN CROSS-VALIDATION SCORES AND TRAINING SCORES. THIS CONVERGENCE INDICATES THAT THE MODEL IS EFFECTIVELY LEARNING FROM THE TRAINING DATA WITHOUT OVERFITTING, IMPROVING ITS RELIABILITY.

	fit_time	score_time	test_r2	train_r2	test_neg_mean_absolute_percentage_error	train_neg_mean_absolute_percentage_error
GradientBoostingRegressor	0.191815	0.012488	0.849625	0.893149	-0.210173	-0.161957
RandomForestRegressor	0.553134	0.035821	0.848898	0.962323	-0.237488	-0.088337
HistGradientBoostingRegressor	0.687106	0.023604	0.843637	0.941193	-0.244804	-0.121569
AdaBoostRegressor	0.076013	0.019393	0.841943	0.857821	-0.426342	-0.411595
ExtraTreesRegressor	0.301220	0.024842	0.829787	1.000000	-0.290061	-0.000000
DecisionTreeRegressor	0.020704	0.010326	0.708044	1.000000	-0.468773	-0.000000
Ridge	0.032437	0.016345	0.554868	0.566943	-0.293058	-0.288171
LinearRegression	0.032055	0.016467	0.546170	0.558655	-0.293093	-0.288273
Lasso	0.017817	0.010742	-0.125899	-0.122371	-0.946075	-0.942971
ElasticNet	0.019262	0.009949	-0.126002	-0.122353	-0.946187	-0.942954
KNeighborsRegressor	0.018784	0.012730	-0.134919	0.092825	-1.039241	-0.773692
KernelRidge	0.065621	0.019962	-142.225487	-149.648242	-2.628012	-2.498697

HYPERTUNING OUR MODELS

```
list(GradientBoostingRegressor().get_params().keys())
```

✓ 0.0s

```
# GradientBoostingRegressor Hypertune
param_grid = {
    'regressor__n_estimators': [100, 200, 300, 400, 500],
    'regressor__learning_rate': [0.01, 0.1, 0.2, 0.3, 0.4, 0.5],
    'regressor__max_depth': [3, 4, 5, 6, 7]
}

# Setting up the Pipeline
steps[STEPS_LEN] = (
    "hyper", GridSearchCV(
        TransformedTargetRegressor(
            regressor=GradientBoostingRegressor(random_state=42),
            func=np.log1p,
            inverse_func=np.expm1,
        ),
        param_grid,
        cv=5,
        n_jobs=-1,
        verbose=1,
        scoring="r2"
    )
)

# fit model
gbr_search = Pipeline(steps=steps)
gbr_search.fit(X_train, y_train)

print(gbr_search["hyper"].best_estimator_)
print(gbr_search["hyper"].best_params_)
print(gbr_search["hyper"].best_score_)
```

```
gbr_tuned = gbr_search["hyper"].best_estimator_
gbr_untuned = TransformedTargetRegressor(
                regressor=GradientBoostingRegressor(random_state=42),
                func=np.log1p,
                inverse_func=np.expm1,
            )

dummy = TransformedTargetRegressor(
                regressor=DummyRegressor(),
                func=np.log1p,
                inverse_func=np.expm1,
            )
```

Fitting the train data into the model

```
# GBR TUNED
steps[STEPS_LEN] = ("model", gbr_tuned)
gbr_tuned_model = Pipeline(steps=steps)
gbr_tuned_model.fit(X_train, y_train)

# GBR UNTUNED
steps[STEPS_LEN] = ("model", gbr_untuned)
gbr_untuned_model = Pipeline(steps=steps)
gbr_untuned_model.fit(X_train, y_train)

# DUMMY MODEL
steps[STEPS_LEN] = ("model", dummy)
dummy_model = Pipeline(steps=steps)
dummy_model.fit(X_train, y_train)
```

PREDICTING WITH OUR MODEL

REGRESSION MODEL PLOTS

THE SCATTER PLOTS DEPICT THE RELATIONSHIP BETWEEN THE PREDICTED VALUES AND THE ACTUAL VALUES FOR THREE DIFFERENT REGRESSION MODELS.

GRADIENT BOOSTING REGRESSOR (TUNED)

POINTS ARE SCATTERED AROUND A LINE, INDICATING THE PREDICTED VALUES ARE CLOSE TO THE ACTUAL VALUES. THE RED LINE REPRESENTS THE REGRESSION LINE, AND THE PREDICTED VALUES ALIGN CLOSELY WITH THIS LINE. THIS SUGGESTS THAT THE TUNED GRADIENT BOOSTING REGRESSOR IS PERFORMING WELL IN CAPTURING THE UNDERLYING PATTERNS IN THE DATA.

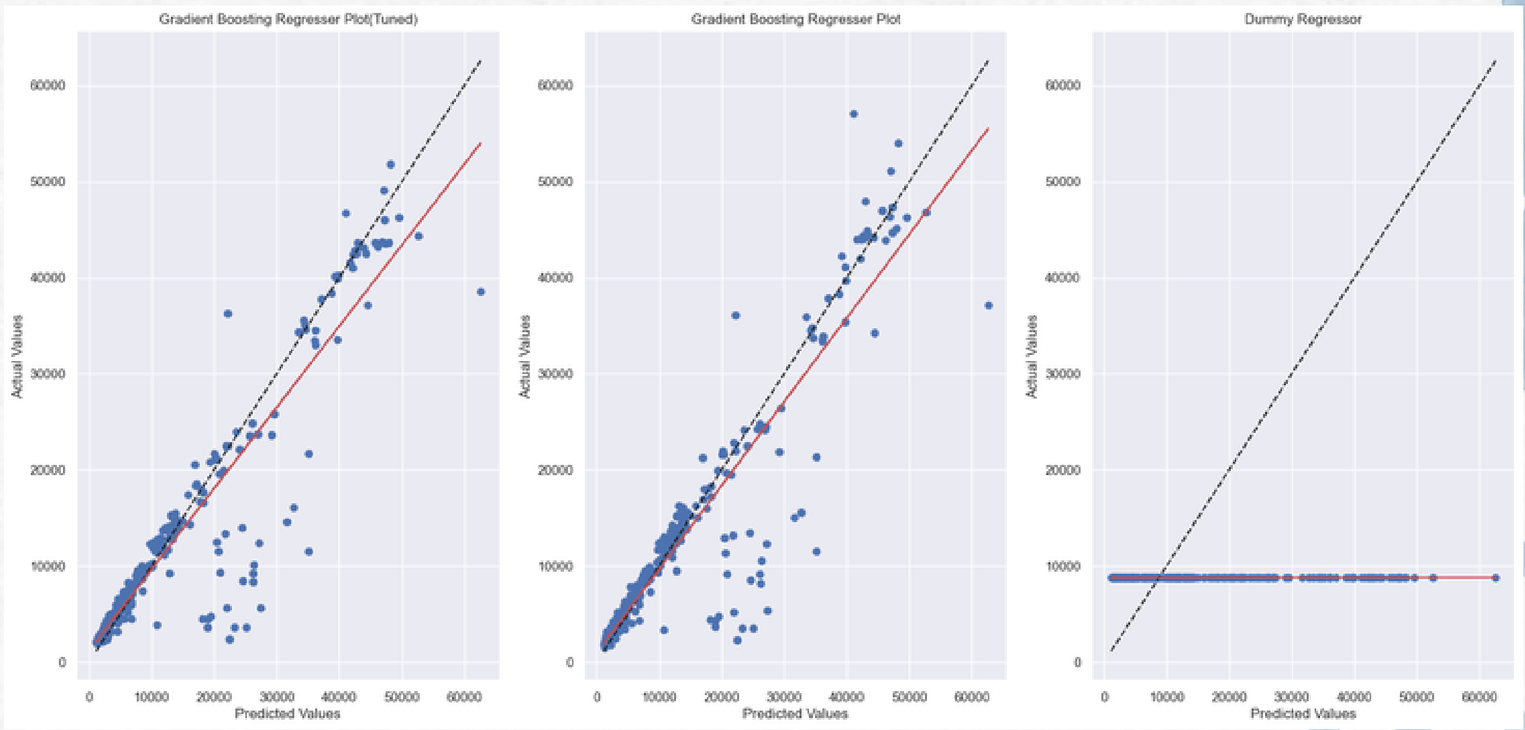
GRADIENT BOOSTING REGRESSOR (UNTUNED)

SIMILAR TO THE TUNED MODEL, POINTS ARE SCATTERED AROUND A LINE, BUT THE ALIGNMENT WITH THE REGRESSION LINE MAY NOT BE AS TIGHT. WHILE STILL CAPTURING THE OVERALL TREND, THE UNTUNED MODEL MIGHT EXHIBIT MORE VARIABILITY IN ITS PREDICTIONS COMPARED TO THE TUNED VERSION.

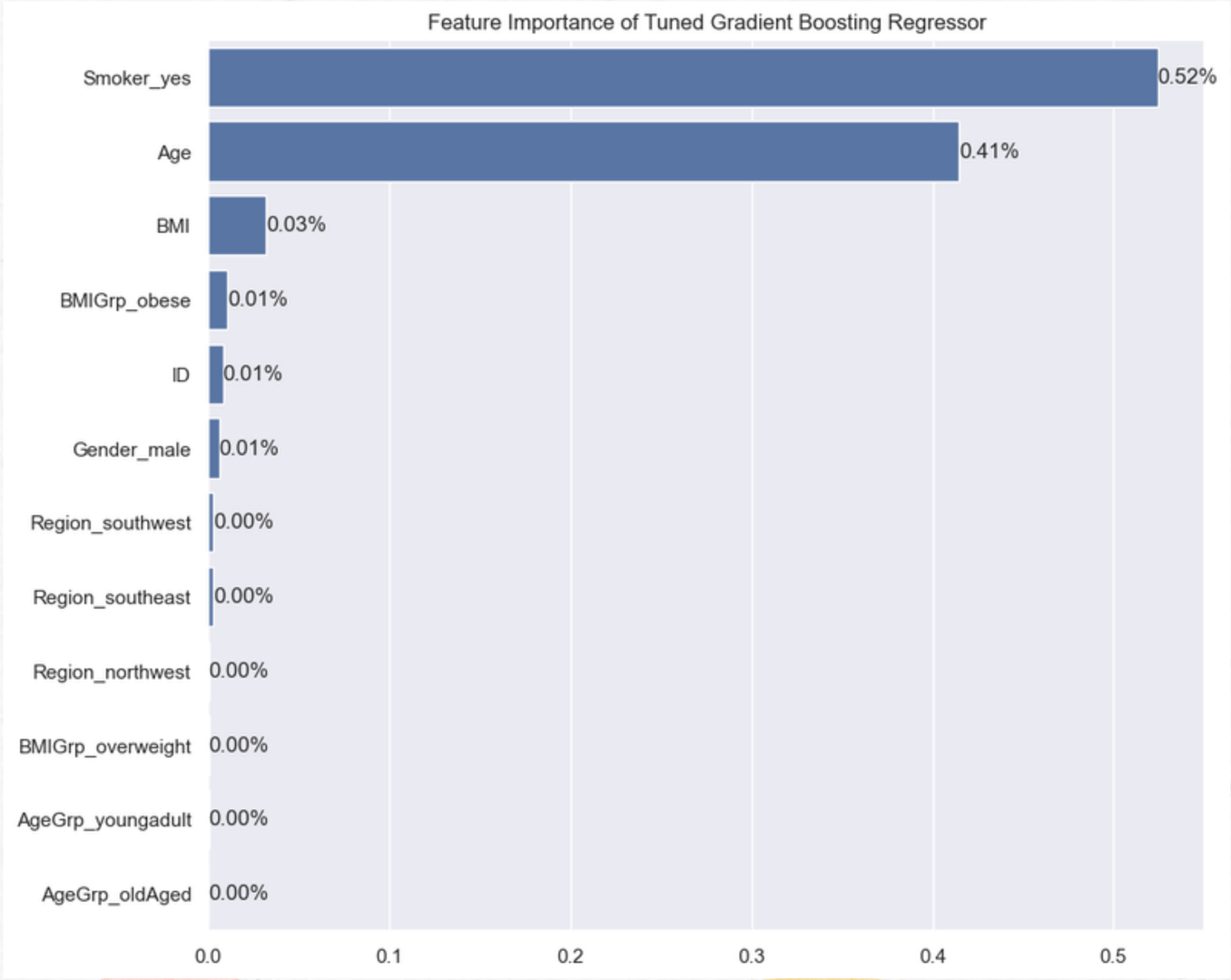
DUMMY REGRESSOR

THE SCATTER PLOT FOR THE DUMMY REGRESSOR SHOWS A LESS ORGANIZED PATTERN, WITH PREDICTED VALUES CONSISTENTLY AROUND 10, 000 ALONG A HORIZONTAL LINE. THIS IS EXPECTED, AS THE DUMMY REGRESSOR PROVIDES A BASELINE FOR COMPARISON AND DOESN'T CAPTURE THE UNDERLYING PATTERNS IN THE DATA AS EFFECTIVELY.

Model	R2 Score	RMSE Score	MAE Score
Gradient Boosting Regressor	0.839962	5261.467838	2555.092285
Dummy Regressor	-0.204991	14437.339914	9690.369294
Gradient Boosting Regressor Untuned	0.830466	5415.321840	2643.459097



GETTING FEATURE IMPORTANCE



CONCLUSIONS

We have successfully created a model to predict cost of healthcare for hospitals in the US
As seen from feature importance, we can see that Smoker has the most importance along with Age which makes sense as smoking would cause bad health issues and some insurance companies do not give insurance to smokers .

From this I learnt how to one hot encode my data and do research on my own on the data and create a model to help predict the actual cost.