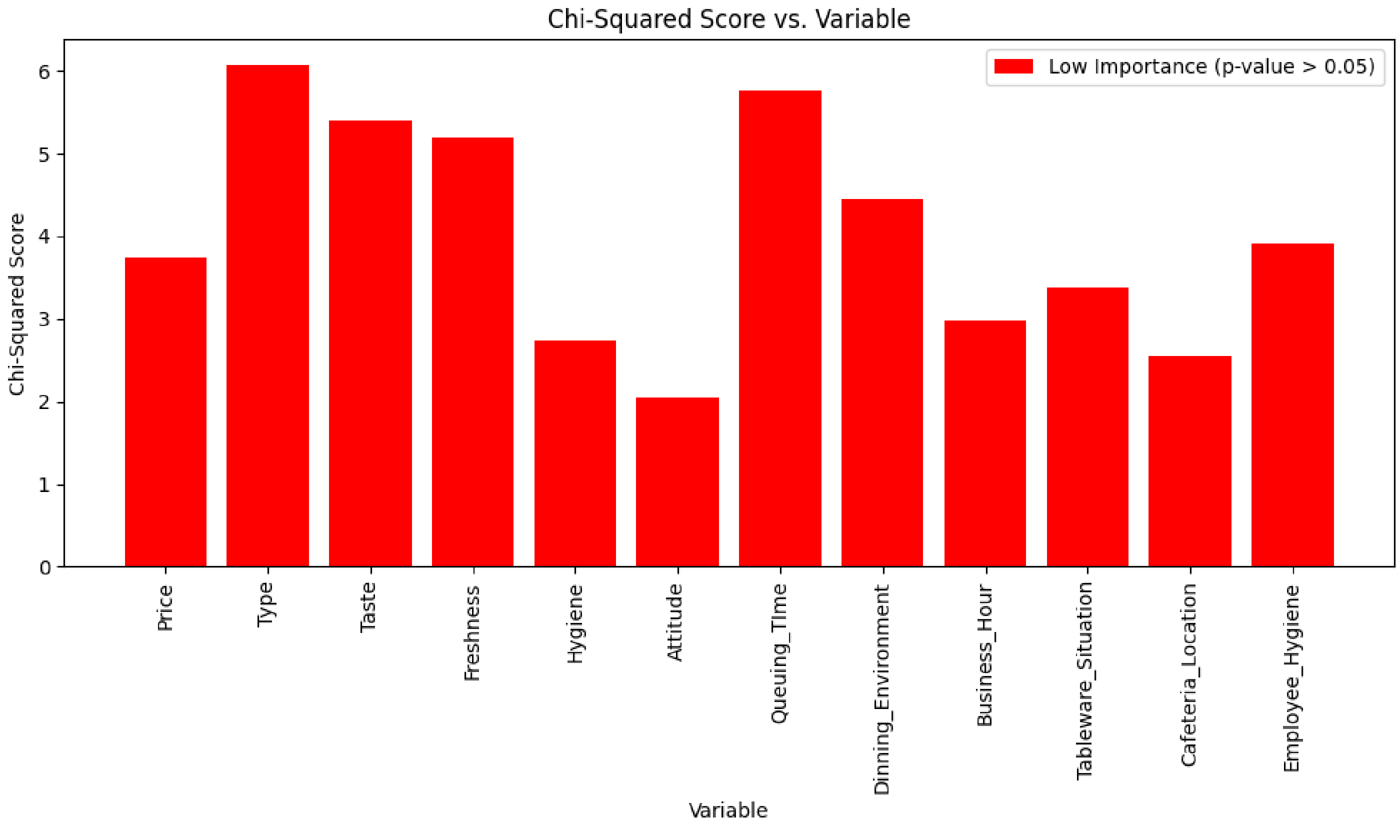
Chi-Squared Analysis of Variables (Number of Features Selection)

In [26]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


data = pd.read_csv("C:\\Users\\ASHIQ\\Desktop\\acafeteria.csv")
X = data.drop(columns=['Overall_Satisfaction'], axis=1)
y = data['Overall_Satisfaction']



from sklearn.feature_selection import chi2
chi_scores = chi2(X, y)
score_value = pd.DataFrame({'Feature': X.columns, \
                            'Chi-Squared Score': chi_scores[0], \
                            'p-value': chi_scores[1]})
score_table = score_value.sort_values(by='Chi-Squared Score', \
                                      ascending=False).reset_index(drop=True)
colors = np.where(score_table['p-value'] > 0.05, 'red', 'blue')
score_table.to_csv("C:\\Users\\ASHIQ\\Desktop\\chi_squared_results.csv", index=False)

print(score_table)

plt.figure(figsize=(10, 6))
plt.bar(score_value ['Feature'], score_value ['Chi-Squared Score'], \
        color=colors)
plt.xlabel('Variable')
plt.ylabel('Chi-Squared Score')
plt.title('Chi-Squared Score vs. Variable')
plt.xticks(rotation=90)
plt.legend(['Low Importance (p-value > 0.05)', 'High Importance (p-value <= 0.05)'])
plt.tight_layout()
plt.savefig('C:\\Users\\ASHIQ\\Desktop\\chi_squared_plot.jpg', format='jpg')
```

```
               Feature  Chi-Squared Score   p-value
0                 Type           6.075000  0.193617
1         Queuing_TIme           5.757473  0.218007
2                Taste           5.395960  0.249027
3            Freshness           5.192332  0.268126
4    Dinning_Environment           4.449314  0.348600
5      Employee_Hygiene           3.905540  0.418941
6                Price           3.745668  0.441519
7    Tableware_Situation           3.380992  0.496203
8        Business_Hour           2.987546  0.559912
9              Hygiene           2.737012  0.602754
10   Cafeteria_Location           2.544226  0.636734
11            Attitude           2.042488  0.727944
```



Recursive Feature Elimination Process

In [24]:

```
import numpy as np
import pandas as pd
from docx import Document
from docx.shared import Inches
from sklearn.feature_selection import RFE
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score, cross_val_predict
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from tabulate import tabulate
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

data = pd.read_csv("C:\\Users\\ASHIQ\\Desktop\\acafeteria.csv")
chisquaredscores = pd.read_csv("C:\\Users\\ASHIQ\\Desktop\\chi_squared_results.csv") \
                                ["Chi-Squared Score"].values

X = data.drop(columns=['Overall_Satisfaction'], axis=1)
y = data['Overall_Satisfaction']

estimators = [
    ('Decision Tree', DecisionTreeClassifier()),
    ('Random Forest', RandomForestClassifier()),
    ('Gradient Boosting', GradientBoostingClassifier()),
    ('Logistic Regression', LogisticRegression())
]

results = []

for name, estimator in estimators:
    rfe = RFE(estimator=estimator, n_features_to_select=4)
    rfe.fit(X, y)

    selected_features = X.columns[rfe.support_]
    X_selected = X[selected_features]

    scores = cross_val_score(estimator, X_selected, y, cv=5)
    mean_score = np.mean(scores)

    y_pred = cross_val_predict(estimator, X_selected, y, cv=5)
    accuracy = accuracy_score(y, y_pred)
    precision = precision_score(y, y_pred, average='weighted')
    recall = recall_score(y, y_pred, average='weighted')
    f1 = f1_score(y, y_pred, average='weighted')

    results.append([name, selected_features, mean_score, \
                    chisquaredscores, accuracy, precision, recall, f1])

## Without Table
#for result in results:
#    print(f"Estimator: {result[0]}")
#    print(f"Selected Features: {', '.join(result[1])}")
#    print(f"Mean Cross-Validation Score: {result[2]}")
#    print(f"Chi-Squared Scores: {result[3]}")
#    print(f"Accuracy: {result[4]}")
#    print(f"Precision: {result[5]}")
#    print(f"Recall: {result[6]}")
#    print(f"F1 Score: {result[7]}\n")

table_headers = ['Estimator', 'Selected Features', 'Mean Cross-Validation Score',\
                 'Chi-Squared Scores', 'Accuracy', 'Precision', 'Recall', 'F1 Score']
table_data = []

for name, selected_features, mean_score, _, accuracy, precision, recall, f1 in results:
    table_data.append([name, ', '.join(selected_features), mean_score, '', \
                       accuracy, precision, recall, f1])

table = tabulate(table_data, headers=table_headers)

doc = Document()
doc.add_heading('Feature Selection Results', level=1)
table_paragraph = doc.add_paragraph()
table_paragraph.add_run(table)

results_final= pd.DataFrame(results, columns=['Estimator', 'Selected Features',\
                                            'Mean CV Score', 'Chi-Squared Scores', \
                                            'Accuracy', 'Precision', 'Recall', \
                                            'F1 Score'])

results_final.to_excel('C:\\Users\\ASHIQ\\Desktop\\feature_selection_results.xlsx', index=False)
```