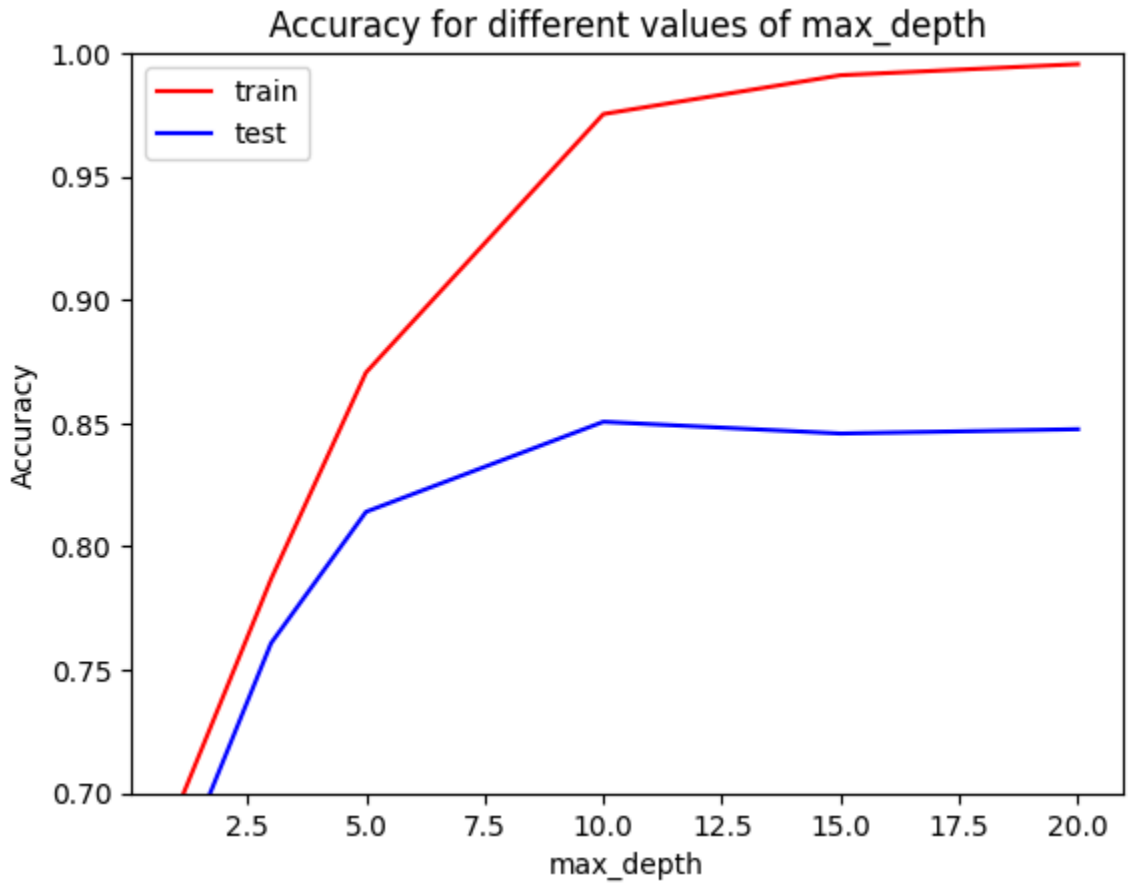
Problem 3

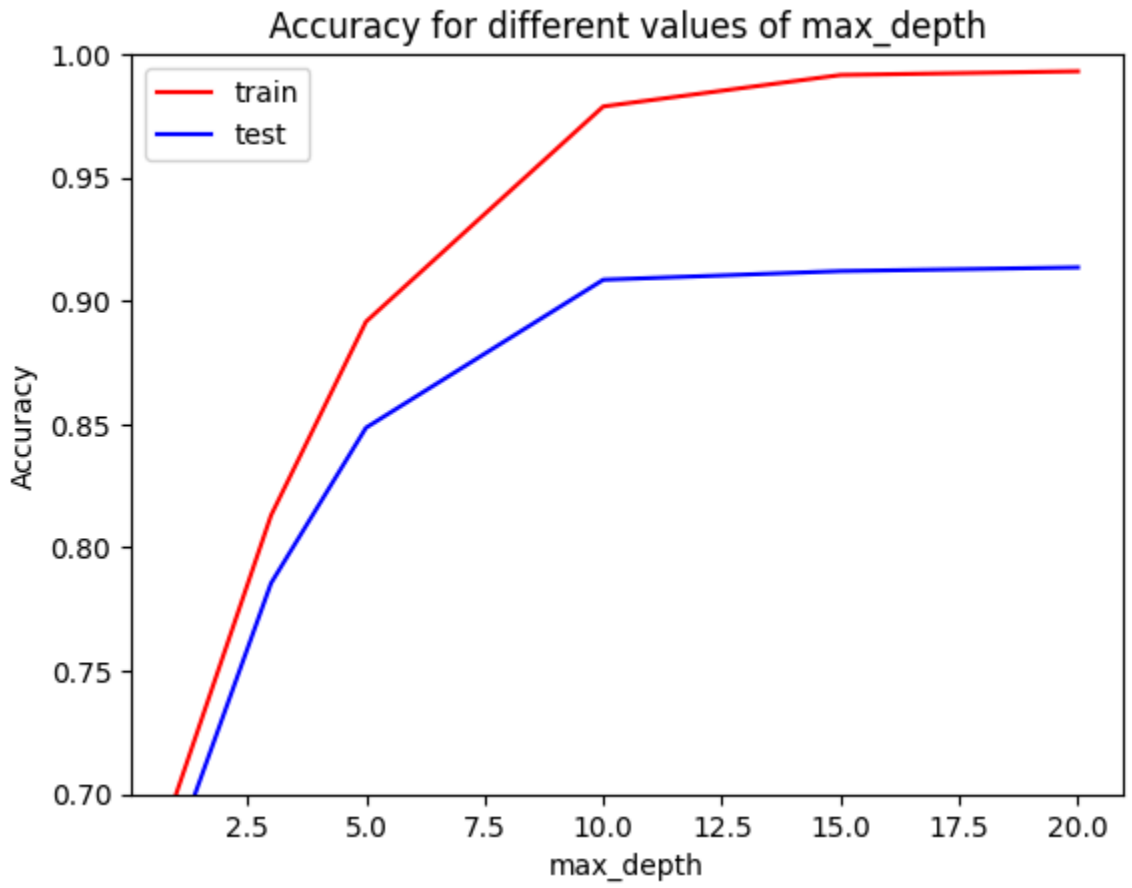
3.1 The first set of plots shows the train and test accuracy for different values of max_depth using a decision tree and a random forest, respectively. How do the accuracies and the generalization gap vary with max_depth in these cases? For a particular value of max_depth, how does the generalization gap for the decision tree compare with that of the random forest? Explain your observations.

The accuracies and the generalization gap increased as the max depth increased for both the decision tree and random forest. For a specific value of max_depth, the generalization gap of the decision tree is larger than that of the random forest, showing that the decision tree was overfitting more than the random forest did.

Model Type: Decision Tree



Model Type: Random Forest

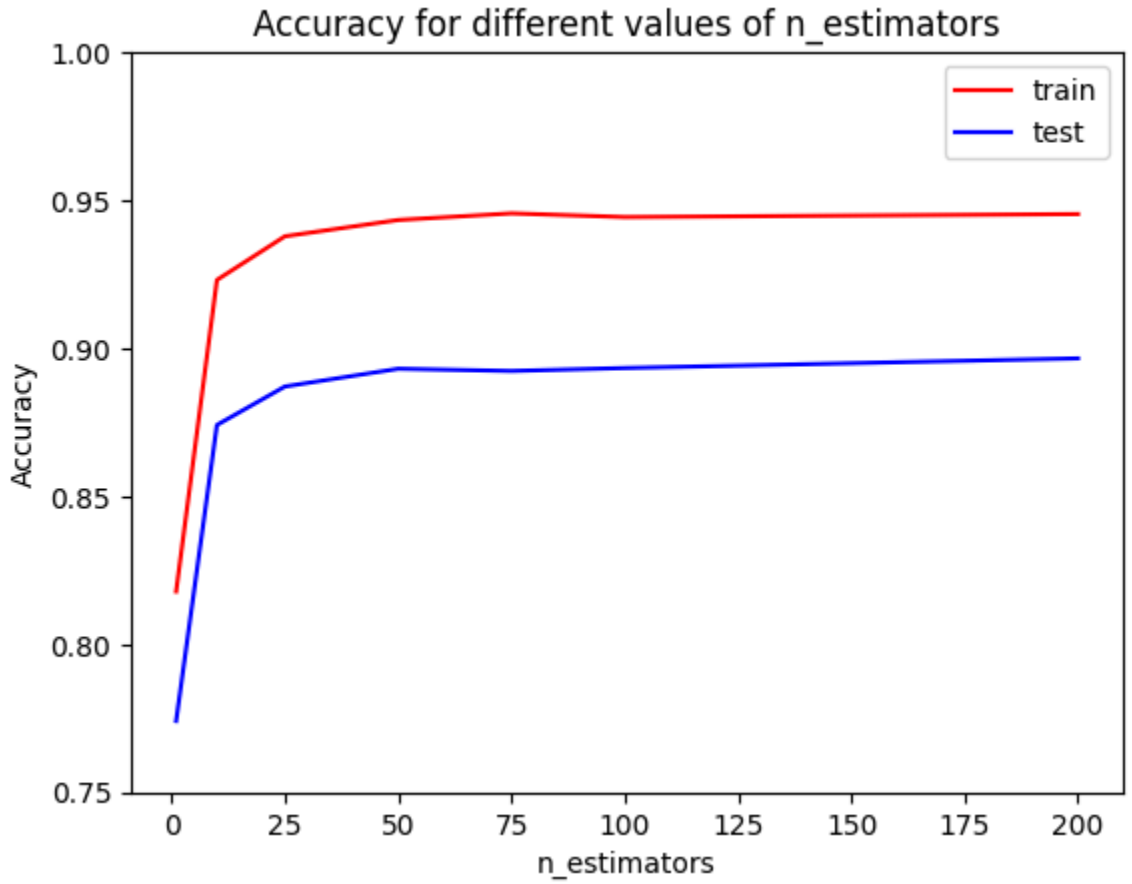


3.2 The next plot shows the train and test accuracy for different values of n_estimators for a random forest. Comment on your observations regarding the accuracies and the generalization gap. What value(s) (range or approximate values are enough) would you prefer to use for this parameter? Give reasons why. Hint: Are there any drawbacks to using very high values of n_estimators? For the next three parts of this question, we will also look at how the size of the training set influences the accuracy trends for a given parameter. In each case, for the first plot, the training set consists of 4000 samples whereas for the second plot, it contains 1000 samples.

Although the accuracies increased rapidly along with n_estimators, they flattened after reaching a certain point. Also, the generalization gap remained nearly the same, meaning that fewer trees do not necessarily lead to overfitting.

To pick a suitable number of estimators, we need to take the computational cost into account and balance the random forest between the number of estimators and computational cost. In our observation, the accuracies increased with the number of estimators. However, more

estimators use more computational resources. Therefore, a value within the range where the accuracies started to flatten (25 to 50) would be a suitable choice.

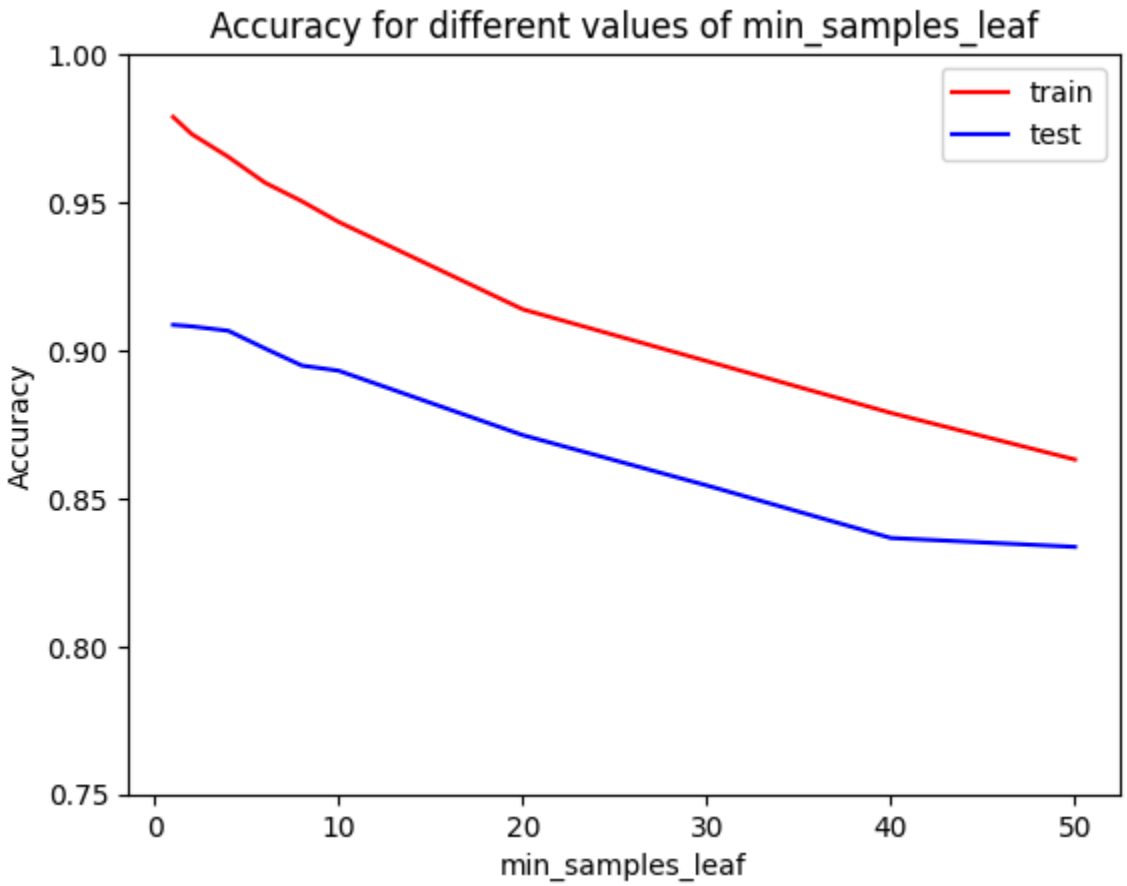


3.3 The next set of plots shows the train and test accuracy for different values of min_samples_leaf for a random forest. Taking into account the behaviors for different training set sizes, explain your observations for very low and very high values of min_samples_leaf. What do you conclude from this trend? What could be the reasons for such a behavior?

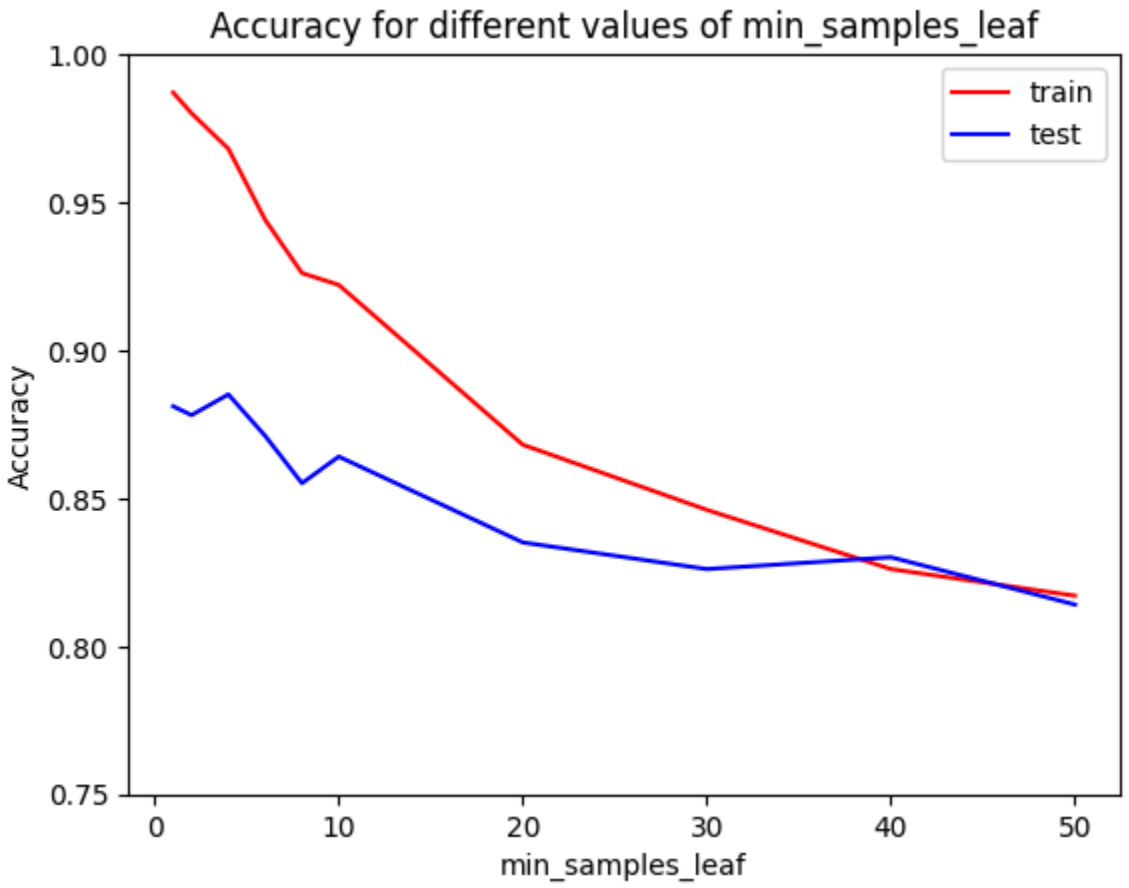
The random forest accuracies steadily decreased as the minimum number of samples required to be a leaf (min_samples_leaf) increased. On the other hand, the training accuracy for the decision tree dropped faster than the testing accuracy, and started the training accuracy started to merge with the testing accuracy.

The reason why training accuracy dropped faster than the testing accuracy in a decision tree might result from the regularization. When min_samples_leaf is small, the model tends to overfit. In contrast, if min_samples_leaf is large, the decision tree will generalize better, thus having a smaller generalization gap.

Random Forest



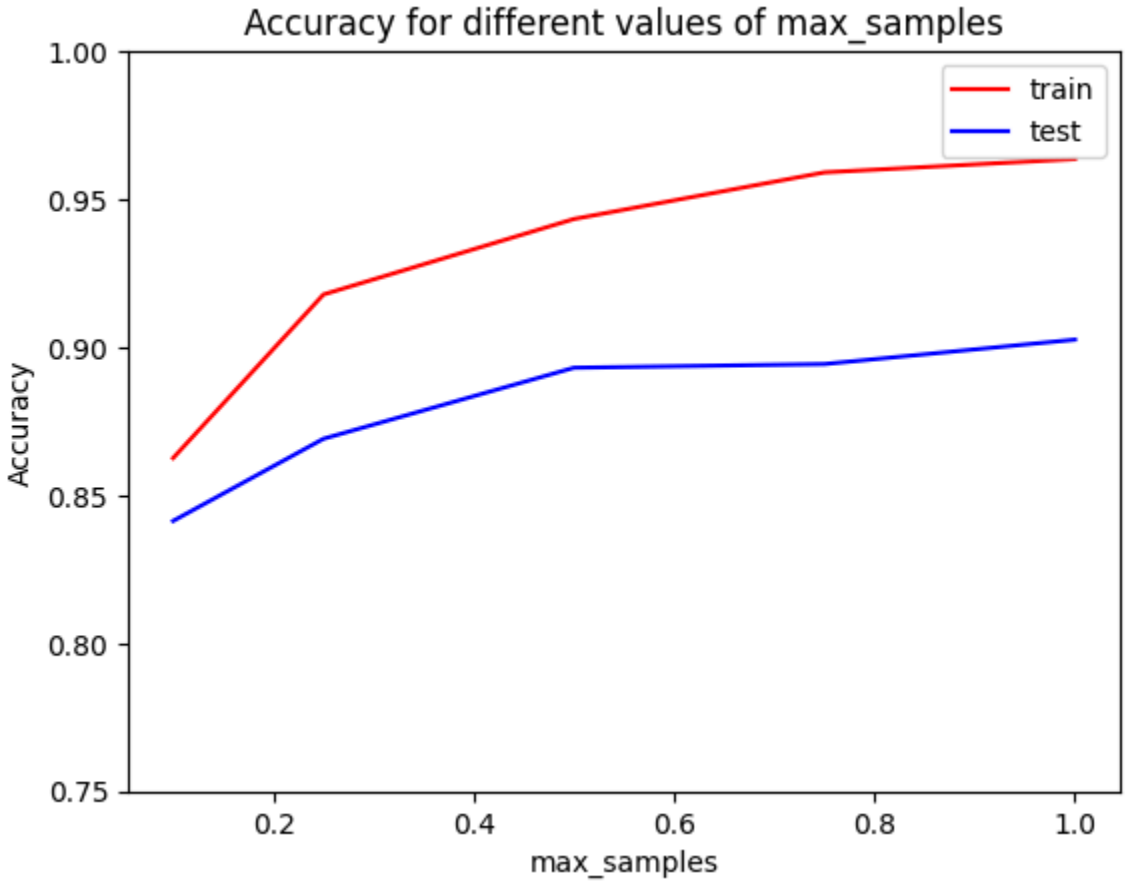
Decision Tree



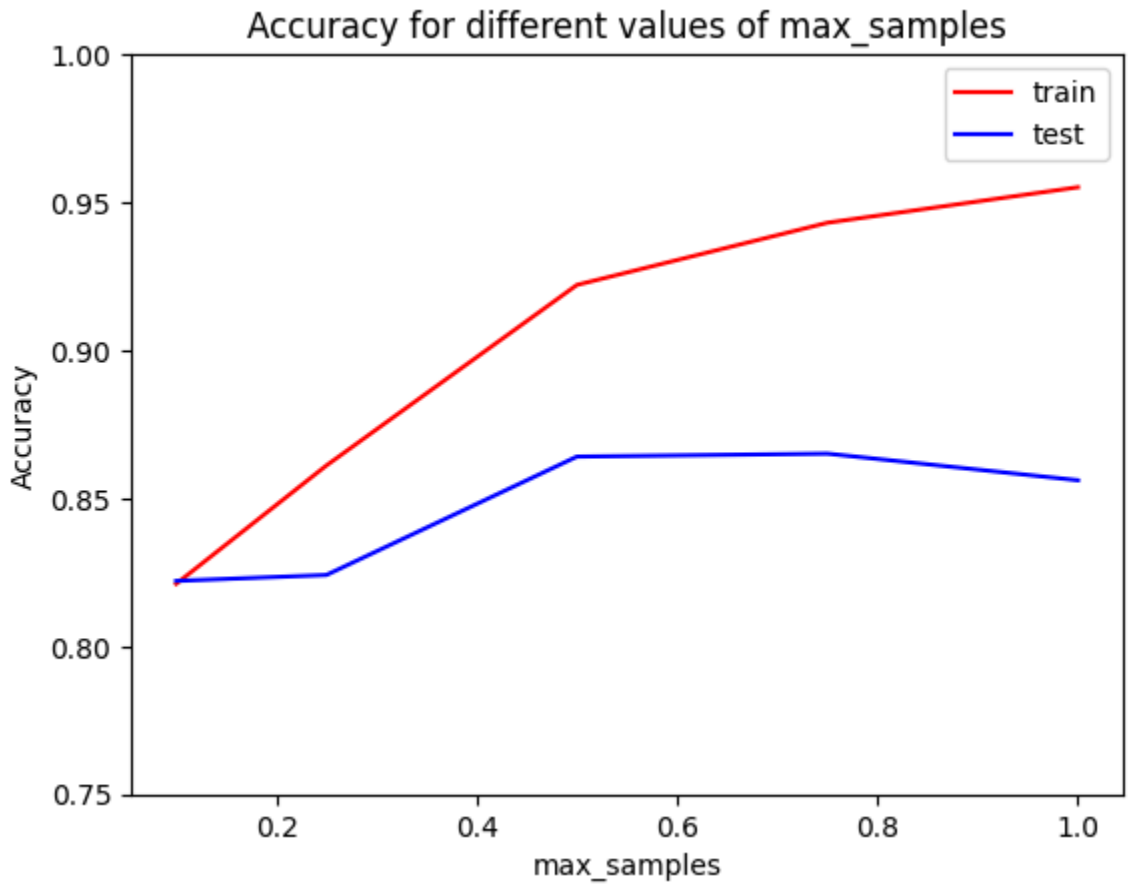
3.4 The next set of plots shows the train and test accuracy for different values of max_samples for a random forest. What do you observe for very low and very high values of max_samples? Would you prefer to use 5 low, intermediate, or high values for this parameter in both cases? Give reasons why. Hint: How does the size of the training set influence the choice of this parameter?

For the random forest and decision tree models, most accuracies and generalization gaps increased with max_samples. However, the testing accuracy of the decision tree reached its peak when max_samples = 0.5 and started to decline at max_samples = 0.75. Therefore, we would prefer to use intermediate values for max_samples. These intermediate values would allow the model to learn from an adequate portion of the dataset while avoiding excessive overfitting.

Random Forest



Decision Tree

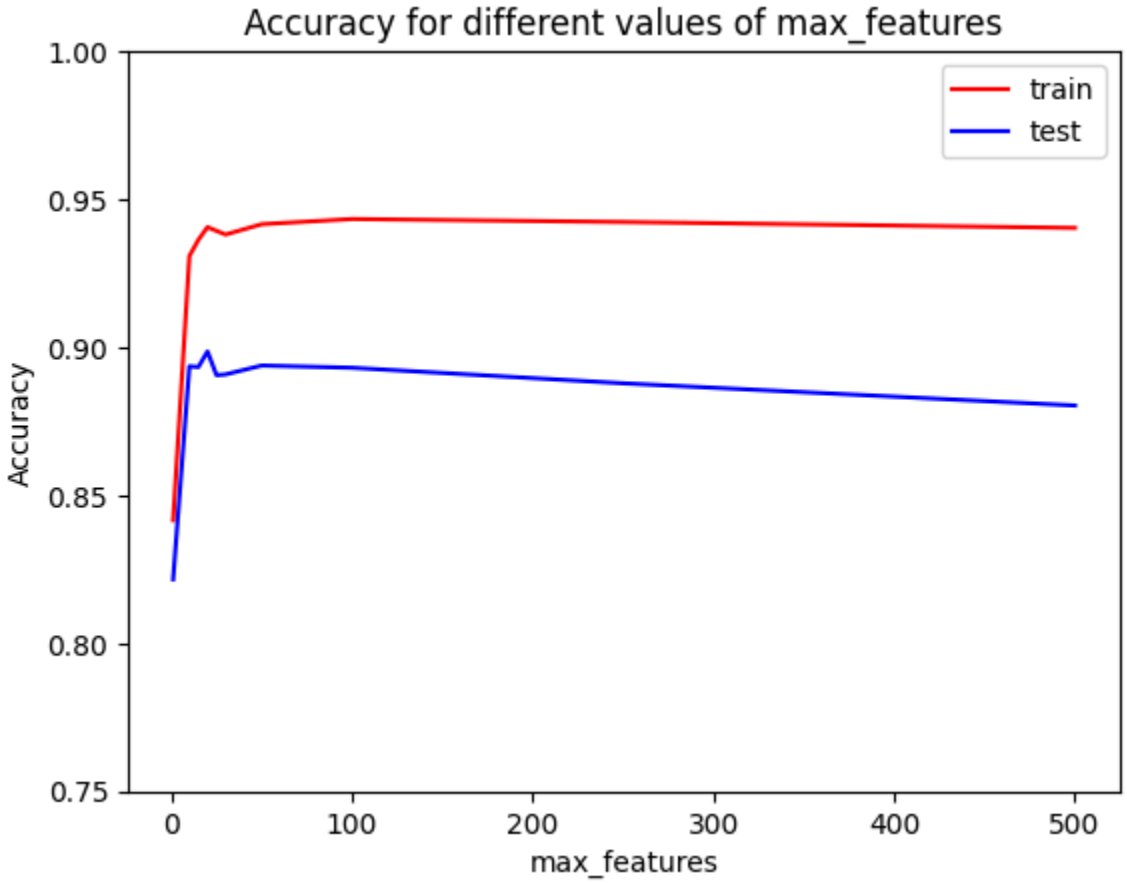


3.5 The next set of plots shows the train and test accuracy for different values of max_features for a random forest. Comment on your observations regarding the accuracies and the generalization gap for the two training set sizes. What is the best range of values for this parameter in both cases? Is it similar/different? Explain your observations.

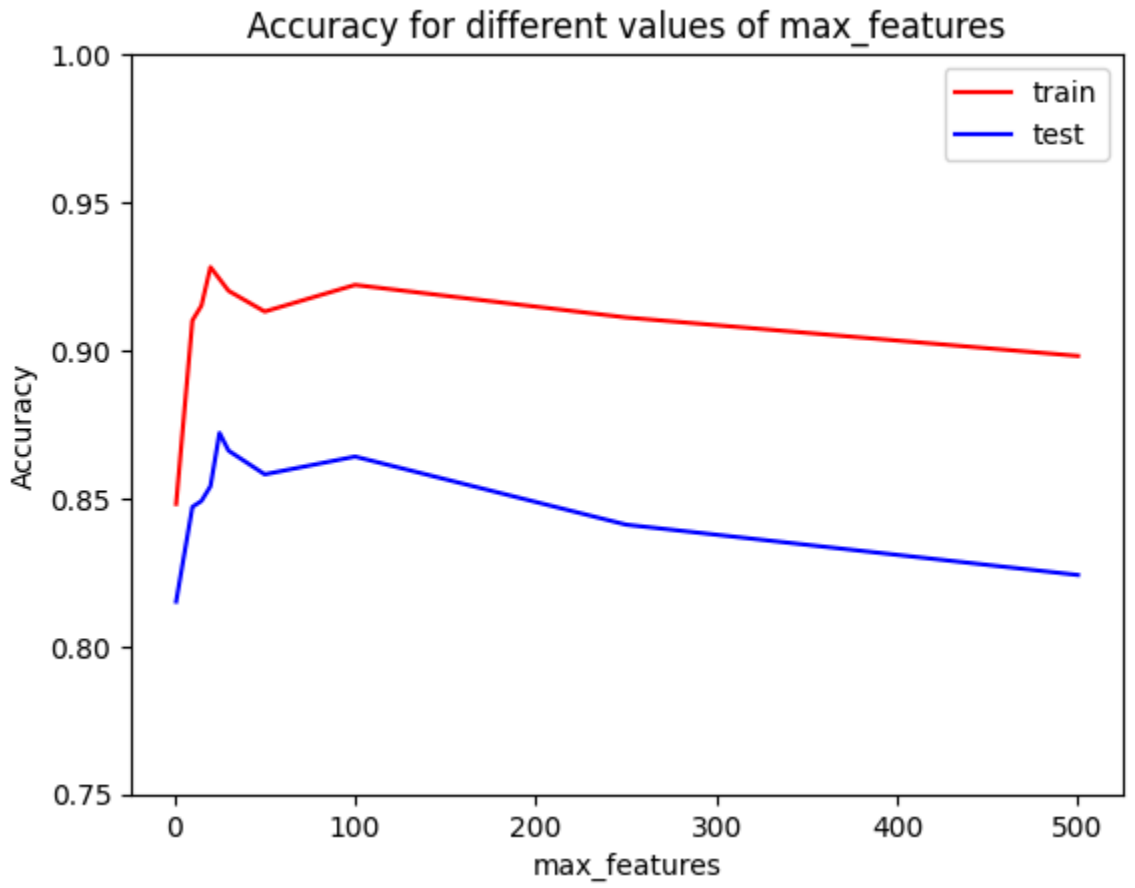
The generalization gaps for both models increased as max_features increased. As for training and testing accuracies, the accuracies for both random forest and decision tree rose rapidly when max_features increased from 1 to 10. The accuracies for the random forest then slightly decreased afterward, whereas the decision tree accuracies decreased more significantly.

The best range of max_features for random forests is 10 to 50. As for decision trees, the best range is 30 to 100.

Random Forest

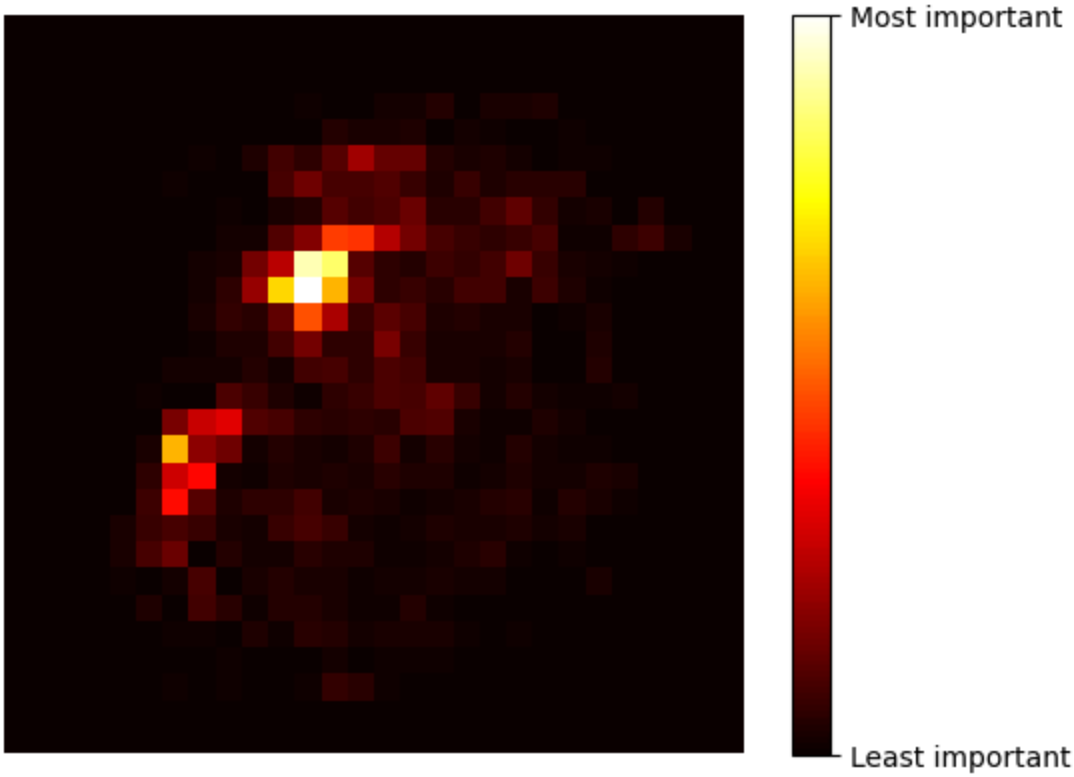


Decision Tree



3.6 The last plot shows the importance of different pixels/portions of the image for a trained random forest model to make its predictions. What portions of the image does the model seem to be focusing on? In other words, can you think of reasons why the pixels with higher importance are indeed important for the prediction task (classifying whether the digit is smaller than 5 or not)? As is usually true for such open-ended questions, there can be multiple correct answers here and we're looking more for your reasoning than a specific answer.

The model seems to be focusing on the slight upper-left and bottom-left to the center of the image, showing that these areas contribute more on making classification. We speculate that this results from the structure difference for each digit. Take the bottom-left for example, the number 2 overlaps with the important part, whereas 7 does not. Therefore, they will be seen as different categories.

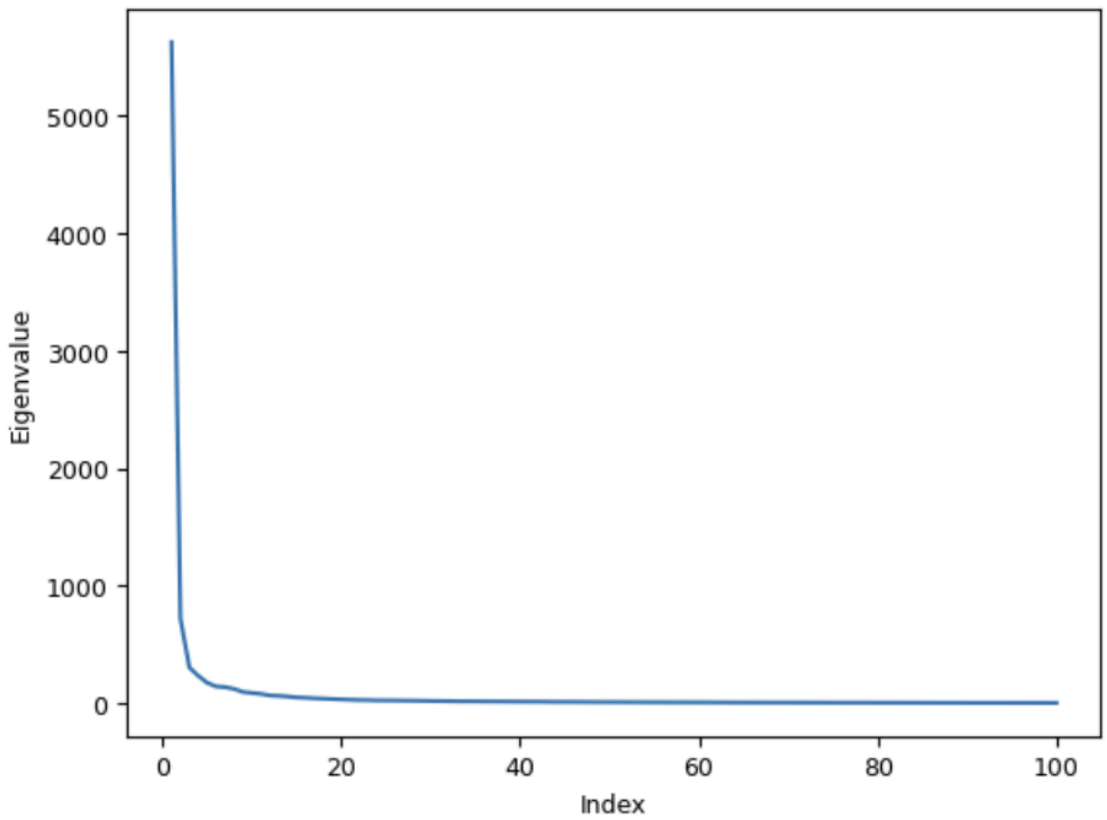

Most important
Least important

Problem 4

4.1 We will use PCA to find the first 100 principal components of the data. Let ˜Mc be the centered version of ˜M. Use the PCA function from the sklearn library (refer https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html) to get V, i.e. the set of first 100 principal components or eigenvectors of ˜Mc. Note that you can directly use the fit method with ˜Mc as the input. Also, get the eigenvalues of the covariance matrix (check the documentation of the function) and plot all the 100 eigenvalues. Do the eigenvalues seem to decay? What percent of the variance in the data is explained by the first 100 eigenvalues we calculated (note that there are 10,000 eigenvalues in total)?

Yes. We observed the eigenvalues decayed.
Percentage of variance explained by the first 100 eigenvalues: 75 %

```
(10000, 10000)
10000
5585
```



4.2 We regard the ith row of ˜E as the embedding of the ith word. Next, we will define a similarity metric for the word embeddings. We will use the cosine-similarity as the similarity metric. As all the vectors have unit ℓ2 norm, the cosine similarity between two words i and j with embeddings wi and wj is equal to the inner product ⟨wi, wj ⟩. Now that we have a similarity metric defined, we can have some fun with these embeddings by querying for the closest word to any word we like! Try finding the closest words to some common words, such as “learning”, “university”, “california”, and comment on your observations.

Most similar word to 'learning': teaching
Most similar word to 'university': college
Most similar word to 'california': florida

We noticed that words with similar meanings or those that appear in similar contexts (even for opposite-meaning words) tend to be grouped closely together in the vector space. This indicates that word embeddings effectively capture semantic relationships and contextual similarities between words.

4.3 We’ll now interpret the principal components/eigenvectors (columns of V). For any i, denote vi as the eigenvector corresponding to the ith largest eigenvalue. Note that the entries of this vector correspond to the 10000 words in our dictionary, we’ll call these our 10000 variables. By sorting the entries of vi by absolute value, and observing what the top 10 variables and their (signed) entries are, we can infer what information the ith eigenvector roughly captures. Can you find 5 interesting eigenvectors, and point out what semantic or syntactic structures they capture? Can you do this for all 100 eigenvectors? Hint: What do you observe about PCs or eigenvectors with small Eigenvalues?

Our results are sorted by eigenvalue in decreasing order

#1
['league', 'school', 'team', 'games', 'college', 'game', 'station', 'season', 'county', 'music']
5618.770631329187
Semantic/Syntactic Structure: Terms related to sports events and organizations.

#2
['born', 'john', 'james', 'david', 'robert', 'william', 'george', 'jr', 'thomas', 'michael']
Explained_variance: 719.8110255847046
Semantic/Syntactic Structure: Personal names often found in English-speaking populations.

#27
['irish', 'scottish', 'german', 'polish', 'italian', 'australian', 'swedish', 'american', 'canadian', 'dutch']
24.50948701960919
Semantic/Syntactic Structure: Nationalities of different countries

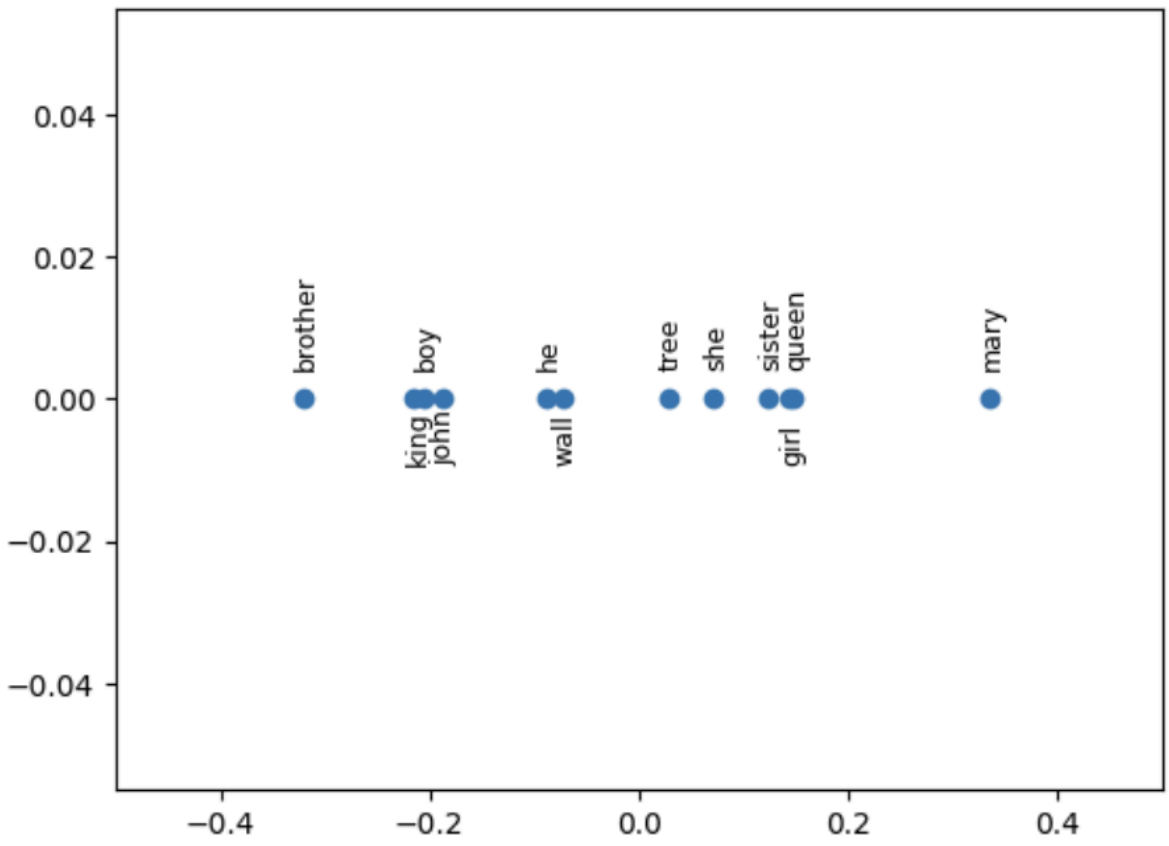
#47
['drama', 'television', 'actress', 'tv', 'comedy', 'starring', 'party', 'directed', 'film', 'actor']
13.45069092377796
Semantic/Syntactic Structure: movies or media context

#96
['were', 'where', 'which', 'was', 'becomes', 'been', 'began', 'species', 'she', 'that']
5.339106700324933
Semantic/Syntactic Structure: common words that can be used in any context

eigenvectors with small eigenvalues, they represent directions in the data space with minimal variance. These eigenvectors capture less important or noisy patterns in the data and may not correspond to clear semantic or syntactic structures. Therefore, they are usually less informative compared to eigenvectors with larger eigenvalues.

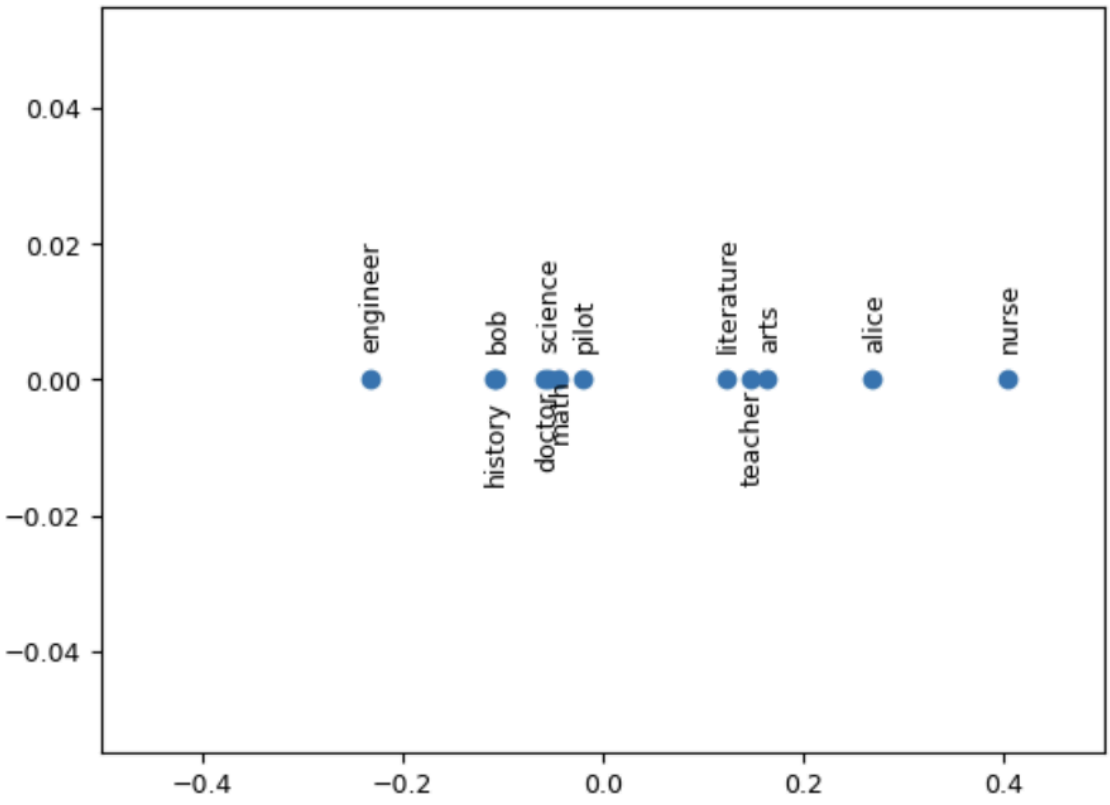
4.4.1 Project the embeddings of the following words onto ˆw: boy, girl, brother, sister, king, queen, he, she, john, mary, wall, tree. Present a plot of projections of the embeddings of these words marked on a line. For example, if the projection of the embedding for “girl” onto ˆw is 0.1, then you should label 0.1 on the line with “girl”. What do you observe?

We notice that terms associated with feminine connotations, like 'Mary', 'queen', 'girl', and 'sister', exhibit positive values in the projections. Conversely, words carrying masculine implications, such as 'brother', 'John', 'boy', 'king', and 'he', display negative values. Gender-neutral terms, such as 'tree', tend to cluster around zero. The proximity of 'she' to zero may stem from its frequent association with objects of personal affection. Interestingly, 'wall' leans toward the masculine side, perhaps due to its symbolism of solidity often associated with masculinity. These projections effectively encapsulate the gender connotations embedded within words.



4.4.2 Present a similar plot of the projections of the embeddings of the following words onto ˆw: math, history, nurse, doctor, pilot, teacher, engineer, science, arts, literature, bob, alice. What do you observe? Why do you think this is the case? Do you see a potential problem with this? Remember that word embeddings are extensively used across NLP. Suppose LinkedIn used such word embeddings to find suitable candidates for a job or to find candidates who best match a search term or job description. What might be the result of this? If you want to learn more about this, you might find it interesting to read the original paper1 which pointed out this issue in word embeddings.

We notice that terms such as nurse, teacher, and arts are biased towards females, whereas terms such as engineer, doctor, and pilot are biased towards males. If this embedding is used on LinkedIn, it might reinforce gender stereotypes by promoting engineering roles to males and nursing roles to females.



4.5 We have provided a dataset, analogy_task.txt, which tests the ability of word embeddings to answer analogy questions, such as those represented in Fig. 3. Using the cosine similarity metric, find and report the accuracy of the word embeddings you have constructed on the word analogy task. Look at the incorrect/correct answers of the approach and comment on the results. For example, what types of analogy questions seem to be harder to answer correctly for this approach?

The accuracy for analogy_task.txt was 55%.
We found queries like (#0) capital-country, (#1) male-female, (#3) activity words are performing well.

In contrast, (#7) city-state, (#6) country-currency, and (#11) singular-plural result in incorrect predictions.

'man' is to 'woman' as 'king' is to 'queen'

	word1	word2	word3	word4	prediction
0	beijing	china	berlin	germany	germany
1	boy	girl	husband	wife	wife
2	good	better	fast	faster	faster
3	go	going	play	playing	playing
4	china	chinese	russia	russian	russian
5	dog	dogs	dollar	dollars	dollars
6	usa	dollar	europe	euro	dollars
7	austin	texas	seattle	washington	minnesota
8	slow	slowly	sudden	suddenly	gradually
9	aware	unaware	certain	uncertain	various
10	saying	said	seeing	saw	realized
11	car	cars	cat	cats	rabbit

Total accuracy: 0.5538048343777977